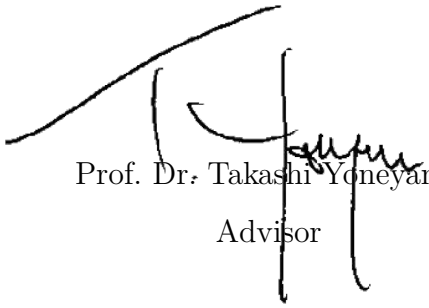
Thesis presented to the Instituto Tecnológico de Aeronáutica, in partial fulfillment of the requirements for the degree of Doctor of Science in the Graduate Program of Electronics and Computer Engineering, Field of Computing.

João Paulo de Andrade Dantas

SIMULATION AND MACHINE LEARNING FOR DECISION SUPPORT AND AUTONOMY IN AIR COMBAT OPERATIONS

Thesis approved in its final version by signatories below:



Prof. Dr. Takashi Yoneyama

Advisor

Prof. Dr. Marcos R. O. A. Maximo

Co-advisor

Campo Montenegro
São José dos Campos, SP - Brazil
2025

Cataloging-in Publication Data
Documentation and Information Division

Dantas, João Paulo de Andrade
Simulation and Machine Learning for Decision Support and Autonomy in Air Combat Operations / João Paulo de Andrade Dantas.
São José dos Campos, 2025.
345f.

Thesis of Doctor of Science – Course of Electronics and Computer Engineering. Area of Computing – Instituto Tecnológico de Aeronáutica, 2025. Advisor: Prof. Dr. Takashi Yoneyama. Co-advisor: Prof. Dr. Marcos R. O. A. Maximo.

1. Combate aéreo. 2. Simulação. 3. Aprendizado de máquina. 4. Veículos autônomos. 5. Sistemas de apoio à decisão. 6. Computação. I. Instituto Tecnológico de Aeronáutica. II. Simulation and Machine Learning for Decision Support and Autonomy in Air Combat Operations.

BIBLIOGRAPHIC REFERENCE

DANTAS, João Paulo de Andrade. **Simulation and Machine Learning for Decision Support and Autonomy in Air Combat Operations**. 2025. 345f. Thesis of Doctor of Science – Instituto Tecnológico de Aeronáutica, São José dos Campos.

CESSION OF RIGHTS

João Paulo de Andrade Dantas
Rua H20B, 102, Campus do CTA
12.228-470 – São José dos Campos–SP

SIMULATION AND MACHINE LEARNING FOR DECISION SUPPORT AND AUTONOMY IN AIR COMBAT OPERATIONS

João Paulo de Andrade Dantas

Thesis Committee Composition:

Prof. Dr.	Carlos H. C. Ribeiro	President	-	ITA
Prof. Dr.	Takashi Yoneyama	Advisor	-	ITA
Prof. Dr.	Marcos R. O. A. Maximo	Co-advisor	-	ITA
Prof. Dr.	Ana Carolina Lorena	Member	-	ITA
Prof. Dr.	Jaime S. Ide	Member	-	Lockheed Martin
Prof. Dr.	Paulo C. G. Da Costa	Member	-	George Mason University

ITA

To my family, for their endless support and encouragement; to my friends, for their constant inspiration; and to my advisors, for their invaluable guidance.

Acknowledgments

First, I thank God for giving me strength and guidance throughout this journey.

Also, I would like to thank my beloved wife, Carolina, for her unconditional love, support, and patience. Without her encouragement, I wouldn't have started this journey.

A special thanks to my entire family, near and far, especially my parents, João and Gleides, for their persistent support, wisdom, sacrifices, values, and examples. Their unconditional love and support have been vital in my life.

I am deeply grateful to my advisors, Dr. Takashi Yoneyama and Dr. Marcos R. O. A. Maximo, for their invaluable guidance, motivation, and support during this journey.

I also wish to express my sincere thanks to the dedicated team of the *Ambiente de Simulação Aeroespacial – ASA* project, whose hard work and commitment were essential in the development of this study.

This research was partially funded by the Funding Authority for Studies and Projects (FINEP) – Finance Code 01.20.0196.00/2824/20, whose financial support is gratefully acknowledged.

"Artificial intelligence is going to be incorporated in everything we do. It is not a question of if these technologies will change the character of war, it is only a question of when."

— General John Michael Murray

Resumo

Esta tese apresenta um framework de pesquisa para a integração de simulação e aprendizado de máquina no apoio à tomada de decisão em operações de combate aéreo. Com foco no combate aéreo Além do Alcance Visual (Beyond Visual Range – BVR), que representa a modalidade moderna da guerra aérea, o estudo explora cinco áreas principais: ferramentas e serviços de simulação, sistemas de armamento, tomada de decisão tática em voo, análise de dados aeroespaciais e agentes autônomos aéreos. A tese revisa aplicações existentes, identifica lacunas de pesquisa e propõe soluções específicas nesses domínios. As contribuições incluem o desenvolvimento de ferramentas e serviços de simulação voltados para cenários de defesa; a modelagem de sistemas de armamento com previsão em tempo real de zonas de engajamento e estimativa de probabilidade de acerto; o uso de técnicas de aprendizado de máquina para aprimorar a tomada de decisão tática em voo por meio da melhoria do momento de lançamento de mísseis, planejamento de manobras e avaliação da situação tática; a aplicação de abordagens orientadas por dados para apoiar análises aeroespaciais em larga escala; e o treinamento de agentes autônomos utilizando aprendizado por imitação, generativo e por reforço para executar tarefas complexas de combate aéreo. O framework de pesquisa proposto busca apoiar os esforços contínuos de avanço em apoio à decisão e autonomia nas forças aéreas modernas.

Abstract

This thesis presents a research framework for integrating simulation and machine learning to support decision-making in air combat operations. Focused on Beyond Visual Range (BVR) air combat, which represents the modern modality of aerial warfare, the study explores five key areas: simulation tools and services, weapon systems, in-flight tactical decision-making, aerospace data analytics, and aerial autonomous agents. It reviews existing applications, identifies research gaps, and proposes targeted solutions across these domains. The contributions include the development of simulation tools and services tailored to defense scenarios; the modeling of weapon systems with real-time engagement zone prediction and kill probability estimation; the use of machine learning techniques to enhance in-flight tactical decision-making through improved missile launch timing, maneuver planning, and situational assessment; the application of data-driven approaches to support large-scale aerospace analytics; and the training of autonomous agents using imitation, generative, and reinforcement learning to perform complex air combat tasks. The proposed research framework is intended to support ongoing efforts to advance decision support and autonomy in modern air forces.

List of Figures

List of Tables

List of Abbreviations and Acronyms

1DCNN	One-Dimensional Convolutional Neural Network
5DOF	Five Degrees of Freedom
6DOF	Six Degrees of Freedom
A2C	Advantage Actor-Critic
A3C	Asynchronous Advantage Actor-Critic
AAR	After Action Review
ABMS	Agent-Based Modeling and Simulation
ACC	Air Combat Command
ACMI	Air Combat Maneuvering Instrumentation
ACT	Air Combat Training
ADASYN	Adaptive Synthetic Sampling Approach
AESA	Active Electronically Scanned Array
AFSIM	Advanced Framework for Simulation, Integration and Modeling
AHMAPPO	Advantage Highlight Multi-Agent Proximal Policy Optimization
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
AIM	Air Intercept Missile
AIM-7	AIM-7 Sparrow (semi-active radar missile)
AIM-54	AIM-54 Phoenix (long-range missile)
AIM-120	Advanced Medium-Range Air-to-Air Missile
AMQP	Advanced Message Queuing Protocol
AMRAAM	Advanced Medium-Range Air-to-Air Missile
AMS	ASA Management System
ANN	Artificial Neural Network
ANOVA	Analysis of Variance
AO	Area of Operations
AOD	Aim-Off Point
AOR	Area of Responsibility
API	Application Programming Interface

ARH	Active Radar Homing
ASCM	Air-to-Surface Cruise Missile
ASA	Aerospace Simulation Environment
AsaCoins	Resource consumption units in ASA
AsaGym	Python library for ASA reinforcement learning
AsaPy	Python library for ASA data analysis
BC	Behavior Cloning
BDA	Battle Damage Assessment
BID	Brazilian Defense Industrial Base
BRANN	Bayesian Regularization of Artificial Neural Networks
BT	Behavior Tree
BVR	Beyond Visual Range
BVRAAM	Beyond Visual Range Air-to-Air Missile
C2	Command and Control
CAP	Combat Air Patrol
CBR	Case-Based Reasoning
CATS	Combat Air Training System
CCDR	Combatant Commander
CE	Circular Error
CEP	Circular Error Probable
CGF	Computer-Generated Forces
COA	Course of Action
COTS	Commercial Off-the-Shelf
CPU	Central Processing Unit
CRUD	Create, Read, Update, Delete
CSVM	Cascaded Support Vector Machine
CVW	Carrier Air Wing
D3QN	Dueling Double Deep Q-Network
DAT	Data Analysis Tool
DAZ	Dynamic Attack Zone
DBN	Dynamic Bayesian Network
DCA	Defensive Counter Air
DCS	Digital Combat Simulator
DDPG	Deep Deterministic Policy Gradient
DDQN	Double Deep Q-Network
DEZ	Dynamic Escape Zones
DGP	Differential Game Problem
DIS	Distributed Interactive Simulation
DL	Deep Learning

DLZ	Dynamic Launch Zone
DNN	Deep Neural Network
DoD	Department of Defense
DoE	Design of Experiments
DoF	Degrees of Freedom
DQN	Deep Q-Network
DQR	Dynamic Quality Replay
DRGRL	Deep Relationship Graph Reinforcement Learning
DRL	Deep Reinforcement Learning
EAM	Engagement Activity Model
ECEF	Earth-Centered, Earth-Fixed
ECS	Entity-Component-System
EDA	Exploratory Data Analysis
ECM	Electronic Countermeasures
Effs	Effectiveness Indicator
Effy	Efficiency Indicator
EEG	Engagement Envelope Generator
EM	Engagement Model
ENN	Edited Nearest Neighbor
E-SAC	Expert-Soft Actor-Critic
EW	Electronic Warfare
EZ	Engagement Zone
F-14	Grumman F-14 Tomcat Fighter Aircraft
F-16	General Dynamics F-16 Fighting Falcon
F-22	Lockheed Martin F-22 Raptor
F-35	Lockheed Martin F-35 Lightning II
FAB	Brazilian Air Force
FAR	False Alarm Rate
FEZ	Fighter Engagement Zone
FLSC	Fighter Lead-In Simulator Cockpit
FM	Force Multiplier
FSM	Finite State Machine
FTD	Flight Training Device
G	Gravity (acceleration)
GAIL	Generative Adversarial Imitation Learning
GAN	Generative Adversarial Network
GDT-SOS	Gradient Descent–Truncated Symbiotic Organisms Search
GFE	Government Furnished Equipment
GOTS	Government Off-the-Shelf

GRU	Gated Recurrent Unit
GUI	Graphical User Interface
H3E	Hierarchical Framework Embedding Expert Knowledge
HLA	High-Level Architecture
HP	High Priority
HUD	Heads-Up Display
IADS	Integrated Air Defense System
IAGSim	Integrated Air-to-Ground Simulator
ICM	Ingress Control Model
IEAv	Institute for Advanced Studies
IO	Input/Output
IQN	Improved Q-Network
ISR	Intelligence, Surveillance, and Reconnaissance
IT	Information Technology
ITA	Aeronautics Institute of Technology
JSB	Joint Simulation Battlespace
JSON	JavaScript Object Notation
JupyterHub	Multi-user server for Jupyter notebooks
KAERS	Key Air Combat Event Reward Shaping
KDE	Kernel Density Estimation
KDS	Knowledge Discovery in Simulation data
KL-AHP	Kullback–Leibler Analytic Hierarchy Process
KNN	K-Nearest Neighbors
LAR	Launch Acceptability Region
LC	Short Limit
LHS	Latin Hypercube Sampling
LL	Long Limit
LOA	Level of Autonomy
LR	Linear Regression
LSTM	Long Short-Term Memory
LVC	Live, Virtual, and Constructive
MAE	Mean Absolute Error
MAC	Maximum Acceptable Configuration
MADDPG	Multi-Agent Deep Deterministic Policy Gradients
MAP	Minimum Attack Point
MAPE	Mean Absolute Percentage Error
MAPPO	Multi-Agent Proximal Policy Optimization
MAV	Micro Aerial Vehicle
MCDM	Multi-Criteria Decision-Making

MCTS	Monte Carlo Tree Search
MECH	Maneuver and Engagement Control Hierarchy
MEBN	Multi-Entity Bayesian Network
MFD	Multi-Function Display
MIXR	Mixed Reality Simulation Platform
ML	Machine Learning
MLC	Mission Level Controller
MLP	Multi-Layer Perceptron
MLOps	Machine Learning Operations
MOP	Measure of Performance
MOT&E	Mission Oriented Training and Evaluation
MSE	Mean Squared Error
NASA-TLX	NASA Task Load Index
NB	Naive Bayes
NCTR	Non-Cooperative Target Recognition
NDS	National Defense Strategy
NED	North-East-Down
NEZ	No Escape Zone
NFSP	Neural Fictitious Self-Play
NGA	National Geospatial-Intelligence Agency
NM	Nautical Mile
NOLH	Nearly Orthogonal Latin Hypercube
NSD	Number of Scenarios Demanded
NSO	Neuroevolution-based Simulation Optimization
NSS	Number of Scenarios Simulated
NTP	Narrative Team Planning
OA	Operational Analysis
OPFOR	Opposing Force
OR	Operational Research
OST	Operational Support Tool
P2SO	Particle-Pair Swarm Optimization
PBX	Plastic-Bonded Explosive
PCA	Principal Component Analysis
PDA	Probability of Damage Assessment
PESA	Passive Electronically Scanned Array
Pk	Probability of Kill
PIP	Point in Polygon
PMESII	Political, Military, Economic, Social, Information, and Infrastructure
PoKER	Probability of Kill Estimation Rate

POMDP	Partially Observable Markov Decision Process
POC	Point of Contact
PPO	Proximal Policy Optimization
PREC	Precision
PR	Polynomial Regression
PSO	Particle Swarm Optimization
PSP	Parallel Self-Play
PT	Processing Time
PUP	Pop-Up Point
REST	Representational State Transfer
RETP	Rapid Estimation of Terminal Performance
RF	Random Forest
RFR	Random Forest Regressor
RL	Reinforcement Learning
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RTO	Return to Owner
RVP	Retrospective Verbal Probing
RWR	Radar Warning Receiver
SA	Situational Awareness
SAC	Soft Actor-Critic
SAE	Sparse Autoencoder
SAM	Surface-to-Air Missile
SAR	Search and Rescue
SAS	Situational Awareness System
SBA	Simulation-Based Acquisition
SCP	Submarine Command Post
SDA	Situational Data Assessment
SDLPIO	Stochastic Dominant Learning Pigeon-Inspired Optimization
SEAD	Suppression of Enemy Air Defenses
SIAP	Single Integrated Air Picture
SIGINT	Signals Intelligence
SimaaS	Simulation as a Service
SITREP	Situation Report
SL	Supervised Learning
SLAM	Simultaneous Localization and Mapping
SME	Subject Matter Expert
SMOTE	Synthetic Minority Oversampling Technique
SMOTE-ENN	SMOTE with Edited Nearest Neighbor

SMOTE-TL	SMOTE with Tomek Links
SNA	Social Network Analysis
SoS	System of Systems
SOP	Standard Operating Procedure
SVM	Support Vector Machines
TA	Target Assignment
TDA	Target Data Assessment
T&E	Test and Evaluation
TCPA	Time to Closest Point of Approach
TD3	Twin Delayed Deep Deterministic Policy Gradient
TFM	Task Force Mission
TL	Tomek Links
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution
TTP	Tactics, Techniques, and Procedures
TTR	Time to Reach
TVI	Total Value Invested
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UCAV	Unmanned Combat Aerial Vehicles
UML	Unified Modeling Language
VAE	Variational Autoencoder
VBS4	Virtual Battlespace 4
VESPA	Visual Environment for Scenario Preparation and Analysis
V&V	Verification and Validation
WAN	Wide Area Network
WARSIM	War Simulation
WEZ	Weapon Engagement Zone
WNN	Wavelet Neural Network
WVR	Within Visual Range
WTI	Weapons and Tactics Instructor
XGBoost	Extreme Gradient Boosting
ZMQ	ZeroMQ

Contents

Part I

Preliminary Considerations

1 Introduction

Machine learning has emerged as a key technology in modern defense systems, especially within the aerospace domain (RASHID *et al.*, 2023). Its ability to process and interpret large volumes of data, sometimes even in real time, is transforming the capabilities of air forces worldwide (AKHGAR *et al.*, 2015). In aerial warfare, machine learning contributes to enhanced situational awareness, faster and more accurate decision-making, and the development of advanced autonomous systems. These capabilities are essential in complex and rapidly evolving operational environments (SUAREZ; BAEZA, 2023).

Integrating machine learning into aerospace defense systems enables the automation of complex tasks traditionally performed by human operators. These systems can process sensor data to detect potential threats, anticipate adversary maneuvers, and recommend tactical actions. By learning patterns and identifying anomalies, machine learning supports proactive defense strategies and optimized resource allocation (WHITTY, 2022). Beyond threat assessment, machine learning also plays an important role in developing Unmanned Aerial Vehicles (UAVs), autonomous weapon systems, and intelligent agents that can operate independently or in coordination with human pilots (DING *et al.*, 2023). Such capabilities are especially valuable in time-sensitive scenarios, where fast and reliable decision-making can determine mission success (ZHANG *et al.*, 2020).

Machine learning models can be trained using historical combat data, when available, or simulated combat data. However, access to real data is often limited due to its classified nature and national security concerns, which restrict its public availability and use in academic research. In this context, simulation plays a fundamental role by providing a safe, flexible, and cost-effective environment for training, testing, and validating machine learning algorithms. By recreating realistic combat scenarios, simulations enable models to adapt to a wide variety of tactical situations without incurring the risks or expenses associated with live exercises (LITTLEWOOD, 1991). This iterative approach strengthens model robustness and prepares systems for deployment in real-world operations.

Simulation also enables the analysis of "what-if" scenarios, helping military planners and developers evaluate alternative strategies and optimize operational decisions. Virtual environments that reflect the complexity of real-world missions allow for the exploration

of adversarial behavior, refinement of tactical plans, and acceleration of the development of intelligent systems for autonomy and decision support in complex operational environments (REDDEN, 1995; PROCTOR; GUBLER, 1998).

In the context of air force operations, simulation and machine learning contribute to various mission-critical functions, including planning, threat assessment, real-time decision support, and the development of autonomous agents. For example, machine learning algorithms can be integrated into air combat simulators to assist tactical training and strategic decision-making (GORTON *et al.*, 2024). These technologies also support the prediction of missile engagement zones, provide real-time guidance to pilots, and enable the creation of autonomous systems capable of executing complex maneuvers (CHEN *et al.*, 2024; SVENMARCKT; DEKKER, 2003; BARNES *et al.*, 2014; LI *et al.*, 2022c).

Advances in machine learning for defense have been made possible by the increasing availability of quality data and continuous improvements in algorithms (MENTHE *et al.*, 2021). Because real combat data is often limited or classified, high-fidelity simulators have become a key resource. These simulators can recreate realistic combat situations and generate data used to train and test machine learning models. With these tools, modern air forces can build more reliable systems and respond more effectively to the challenges of aerial warfare, where quick decisions are essential (SCHULKER *et al.*, 2021).

Figure 1.1 illustrates the technological pipeline for integrating simulation and machine learning in air combat operations. Due to the limited availability and sensitivity of real combat data, simulated data generated by human pilots or computational agents is often used to train models for decision support and autonomous systems.

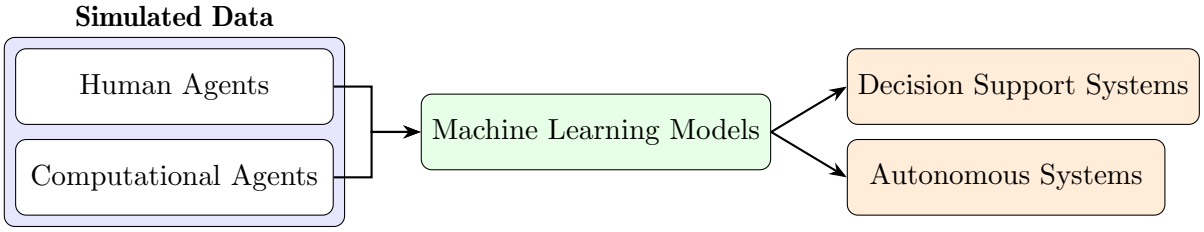


FIGURE 1.1 – Simulated data from human and computational agents is used to train machine learning models that support decision-making and autonomy in air combat operations.

1.1 Proposal

As part of the *Ambiente de Simulação Aeroespacial – ASA* project initiative, developed by the Institute for Advanced Studies (IEAv) of the Brazilian Air Force (FAB), this thesis aims to explore the use of machine learning and simulation in aerial warfare. The research focuses on developing a comprehensive design that integrates simulation and machine learning to enhance decision-making processes for modern air forces.

Research Question

How can a modern air force integrate simulation and machine learning to support decision-making and autonomy in air combat operations, and which areas are most critical to enabling this integration?

The hypothesis is that establishing a unified research framework that integrates simulation and machine learning is essential to effectively address the diverse demands of a modern air force. This structure should include the following five critical areas of development:

Critical Areas of Development

I. Simulation Tools and Services

II. Weapon Systems

III. In-Flight Tactical Systems

IV. Aerospace Data Analytics

V. Aerial Autonomous Agents

To address our scientific question using our hypotheses, this thesis presents a study and review of existing works on autonomous decision-making in modern aerial warfare, focusing specifically on Beyond Visual Range (BVR) air combat. BVR was chosen because it has become the dominant form of modern aerial engagements, largely due to advancements in sensors, missile technology, and electronic warfare (STILLION, 2015). The study begins with a comprehensive review of the literature on BVR air combat, highlighting the evolution of tactics, technologies, and decision-making approaches used in recent aerial operations.

The next phase of this research involves presenting a framework developed by FAB designed to emulate the complexities of modern aerial warfare. This environment focuses on digital transformation through advanced simulation services, incorporating high-fidelity models and real-time data integration to create a realistic and dynamic testing ground for autonomous decision-making systems. A significant portion of the works presented in this thesis was developed using this framework, demonstrating its effectiveness in addressing the diverse demands of modern air combat scenarios.

The thesis then examines the challenges of modeling weapon behavior within simulated combat environments. This includes the development and application of deep learning algorithms to estimate engagement zones, which help determine the effective range and

impact of various systems, particularly in air-to-air scenarios. It also explores the real-time prediction of ground-to-air engagement zones and the use of probabilistic models to estimate kill probability in BVR contexts.

The study also explores machine learning methods to support in-flight decision-making in air combat. This includes using supervised learning to optimize missile launch timing and improve situational awareness, allowing autonomous systems to respond more effectively to changing combat conditions. It also investigates the use of machine learning models to develop adaptive engagement strategies and predict combat outcomes, contributing to higher mission success rates.

The thesis also presents a custom aerospace data analysis library designed to simplify the processing of large simulation datasets and support more informed decision-making. It includes tools for visualization, statistical analysis, and machine learning integration. The library features an automated pipeline that supports multi-source data integration and enables detailed studies of simulation results, helping generate actionable insights.

A key contribution of this research is the development of aerial autonomous agents using reinforcement and imitation learning in BVR air combat simulations. These agents are designed to support tactical decision-making and learn strategic behaviors using either reinforcement learning or imitation learning, depending on the task. To address the limited availability of real flight data, generative learning was used to create realistic synthetic trajectories, expanding the dataset and improving generalization. The research also explores social navigation through scenarios where human pilots and autonomous agents collaborate, combining complementary strengths to accomplish mission objectives.

To summarize the proposed research structure, Figure 1.2 shows the research framework used in this thesis. The diagram explains how the five key areas of development are organized and how they connect to the main goals of using simulation and machine learning in air combat operations. Importantly, simulation, machine learning, and air combat operations are shown at the same level, forming a cycle where each one helps improve the others. Better air combat operations give useful feedback that helps improve simulation tools and machine learning models, while advances in simulation and learning methods also help make combat operations more effective.

This research framework reflects a practical and integrated view adopted throughout the work, serving as the foundation for the developments and experiments presented in this thesis. It also guides the structure of the document: each of the five key areas is addressed in a dedicated chapter following the organization illustrated in the diagram.

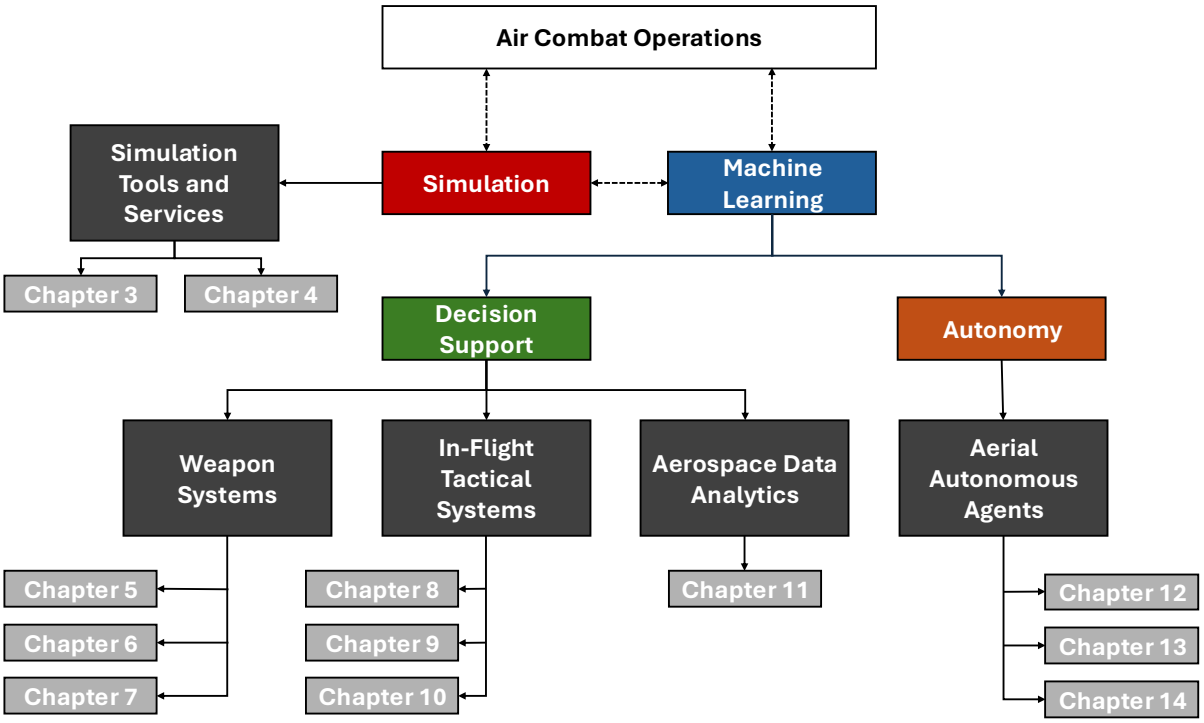


FIGURE 1.2 – Structure of the thesis across the five key areas of development. Simulation, machine learning, and air combat operations form a cycle where improvements in one area support progress in the others.

1.2 Contributions

This thesis aims to enhance the operational effectiveness and decision-making capabilities of a modern air force by investigating how simulation and machine learning can be integrated into aerial combat scenarios. By addressing current gaps and challenges, it proposes a unified research framework and offers practical insights to support future developments in autonomous systems and decision support for aerial operations, with direct applicability to air force institutions and defense organizations involved in planning, training, and operational analysis.

Therefore, the overall contribution of this work is:

Overall Contribution

Propose a unified research framework to guide a modern air force in the integration of simulation and machine learning, identifying the most critical areas for improving decision-making and enabling autonomy in air combat operations.

In addition to this primary objective, the thesis makes several specific contributions, each addressing key aspects of integrating simulation and machine learning in aerial combat. These specific contributions are detailed as follows:

Specific Contributions

① Conduct a review of existing works on simulation and machine learning in modern aerial combat, identifying key methodologies and research gaps.

② Present the development of a simulation tool for aerial combat scenarios to test and evaluate machine learning models in realistic environments.

③ Integrate and analyze simulated weapon systems, assessing their performance and impact when combined with machine learning strategies.

④ Design and implement in-flight decision support systems that use machine learning to enhance situational awareness and decision-making for pilots.

⑤ Conduct aerospace data analytics to process and analyze data from aerial operations, supporting machine learning applications.

⑥ Develop aerial autonomous agents using machine learning to operate independently or in coordination with other agents in air combat operations.

Listed below is a collection of works presented in chronological order, each accompanied by a numerical index at the end of the entry. These indexes correspond directly to the contributions listed in the **Specific Contributions** box above, indicating how each publication relates to the main research efforts developed throughout this thesis.

The specific contributions of this thesis are documented in **fifteen** works, each accompanied, whenever possible, by its respective code repository. The author is the first author of all these works: **fourteen** have been published, and **one** is currently under third-round review. These works are listed below:

DANTAS, J. P. A.; COSTA, A. N.; GERALDO, D.; MAXIMO, M. R. O. A.; YONEYAMA, T. Engagement Decision Support for Beyond Visual Range Air Combat. In: Proceedings of the 2021 Latin American Robotics Symposium, 2021 Brazilian Symposium on Robotics, and 2021 Workshop on Robotics in Education. Proceedings [...]. Natal, RN, Brazil: IEEE, 2021. p. 96–101. ④

DANTAS, J. P. A.; COSTA, A. N.; GERALDO, D.; MAXIMO, M. R. O. A.; YONEYAMA, T. Weapon Engagement Zone Maximum Launch Range Estimation Using a Deep Neural Network. In: BRITTO, A.; DELGADO, K. V. (Ed.). Intelligent Systems. Proceedings [...]. Cham: Springer, 2021. p. 193–207. ISBN 978 3-030-91699-2. ③

DANTAS, J. P. A.; MAXIMO, M. R. O. A.; COSTA, A. N.; GERALDO, D.; YONEYAMA, T. Machine Learning to Improve Situational Awareness in Beyond

Visual Range Air Combat. IEEE Latin America Transactions, v. 20, n. 8, 2022. Available at: https://latamt.ieeer9.org/index.php/transactions/article/view/6530. ④

DANTAS, J. P. A.; COSTA, A. N.; GOMES, V. C. F.; KUROSWISKI, A. R.; MEDEIROS, F. L. L.; GERALDO, D. ASA: A Simulation Environment for Evaluating Military Operational Scenarios. In: THE 2022 WORLD CONGRESS IN COMPUTER SCIENCE, COMPUTER ENGINEERING & APPLIED COMPUTING (CSCE'22). The 20th International Conference on Scientific Computing (CSC'22). Proceedings [...]. Las Vegas, NV, USA, 2022. ②

DANTAS, J. P. A.; COSTA, A. N.; MEDEIROS, F. L. L.; GERALDO, D.; MAXIMO, M. R. O. A.; YONEYAMA, T. Supervised Machine Learning for Effective Missile Launch Based on Beyond Visual Range Air Combat Simulations. In: Proceedings of the Winter Simulation Conference. Proceedings [...]. Singapore: IEEE, 2022. (WSC '22). ④

DANTAS, J. P. A.; MAXIMO, M. R. O. A.; YONEYAMA, T. Autonomous Agent for Beyond Visual Range Air Combat: A Deep Reinforcement Learning Approach. In: Proceedings of the 2023 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation. Proceedings [...]. Orlando, FL, USA: Association for Computing Machinery, 2023. (SIGSIM-PADS'23). ISBN 979-8-4007-0030. Available at: https://doi.org/10.1145/3573900.3593631. ⑥

DANTAS, J. P. A.; GERALDO, D.; COSTA, A. N.; MAXIMO, M. R. O. A.; YONEYAMA, T. ASA-SimaaS: Advancing Digital Transformation through Simulation Services in the Brazilian Air Force. In: Simpósio de Aplicações Operacionais em Áreas de Defesa (SIGE2023). Proceedings [...]. São José dos Campos, Brazil: Instituto Tecnológico de Aeronáutica (ITA), 2023. p. 6. ISSN 1983 7402. Available at: https://www.sige.ita.br/edicoes-anteriores/2023/st/235455_1.pdf. ②

DANTAS, J. P. A.; GERALDO, D.; MEDEIROS, F. L. L.; MAXIMO, M. R. O. A.; YONEYAMA, T. Real-Time Surface-to Air Missile Engagement Zone Prediction Using Simulation and Machine Learning. In: Interservice/Industry Training, Simulation and Education Conference (I/ITSEC). Proceedings [...]. Orlando, FL, USA: National Training and Simulation Association (NTSA), 2023. ③

DANTAS, J. P. A.; SILVA, S. R.; GOMES, V. C. F.; COSTA, A. N.; SAMERSLA, A. R.; GERALDO, D.; MAXIMO, M. R. O. A.; YONEYAMA, T. AsaPy: A Python Library for Aerospace Simulation Analysis. In: Proceedings of the 38th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation. Proceedings [...]. New York, NY, USA: Association for Computing Machinery, 2024.

(SIGSIM-PADS '24), p. 15–24. ISBN 9798400703638. Available at: https://doi.org/10.1145/3615979.3656063. ⑤

DANTAS, J. P. A.; MAXIMO, M. R. O. A.; YONEYAMA, T. Loyal Wingman Assessment: Social Navigation for Human-Autonomous Collaboration in Simulated Air Combat. In: Proceedings of the 38th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation. Proceedings [...]. New York, NY, USA: Association for Computing Machinery, 2024. (SIGSIM-PADS '24), p. 61–62. ISBN 9798400703638. Available at: https://doi.org/10.1145/3615979.3662149. ⑥

DANTAS, J. P. A. Autonomous Pop-Up Attack Maneuver Using Imitation Learning. In: Proceedings of the Winter Simulation Conference, PhD Colloquium. Proceedings [...]. Orlando, FL, USA: IEEE, 2024. ⑥

DANTAS, J. P. A.; COSTA, A. N.; GERALDO, D.; MAXIMO, M. R.; YONEYAMA, T. PoKER: a probability of kill estimation rate model for air-to-air missiles using machine learning on stochastic targets. The Journal of Defense Modeling and Simulation, v. 0, n. 0, p. 15485129241309675, 2025. Available at: https://doi.org/10.1177/15485129241309675 ③

DANTAS, J. P. A.*; COSTA, A. N.*; SCUKINS, E.; MEDEIROS, F. L. L.; ÖGREN, P. Simulation and Machine Learning in Beyond Visual Range Air Combat: A Survey. IEEE Access, v. 13, p. 76755–76774, 2025. ①

DANTAS, J. P. A.; MAXIMO, M. R. O. A.; YONEYAMA, T. Autonomous Aircraft Tactical Pop-Up Attack Using Imitation and Generative Learning. IEEE Access, v. 13, p. 81204–81217, 2025. ⑥

DANTAS, J. P. A.*; MEDEIROS, F. L. L.*; SAMERSLA, A. R.*; BOTELHO, P. L. R.; GOMES, V. C. F.; SILVA, S. R.; FERREIRA, Y. D.; ARANTES, A. O.; AQUINO, M. R. C.; MAXIMO, M. R. O. A. Deep Reinforcement Learning Agents with Collective Situational Awareness for Beyond Visual Range Air Combat. IEEE Access, 2025. (The manuscript is under review at the time of writing this thesis — third round.) ⑥

Additionally, during the research for this thesis, **twelve** additional works were produced on related topics: **ten** of these have been published, **one** has been accepted and is pending publication, and **one** is currently under first-round review. These contributions are detailed in the following articles, where the author of this thesis is listed as an author or co-author:

*Equal contribution.

DANTAS, J. P. A.; SILVESTRE, C. A. de M. Modelo de simulação aplicado às missões de transporte na região amazônica. Aplicações Operacionais em Áreas de Defesa, v. 21, p. 10–15, set. 2020. ⑤

COSTA, A. N.; MEDEIROS, F. L.; **DANTAS, J. P. A.**; GERALDO, D.; SOMA, N. Y. Formation control method based on artificial potential fields for aircraft flight simulation. SIMULATION, v. 98, n. 7, p. 575 595, 2022. Available at: https://doi.org/10.1177/00375497211063380. ⑥

PATRIKAR, J.; **DANTAS, J. P. A.**; GHOSH, S.; KAPOOR, P.; HIGGINS, I.; ALOOR, J. J.; NAVARRO, I.; SUN, J.; STOLER, B.; HAMIDI, M. et al. Challenges in Close-Proximity Safe and Seamless Operation of Manned and Unmanned Aircraft in Shared Airspace. In: Aerial Robotics Workshop, International Conference on Robotics and Automation (ICRA) 2022. Proceedings [...]. Philadelphia, PA, USA: International Conference on Robotics and Automation (ICRA), 2022. ⑥

GOBI, G. H.; BOTELHO, P. L. R.; FERREIRA, T. L.; ARAUJO, T. L.; **DANTAS, J. P. A.** Ambiente de Simulação Aeroespacial: Capacidades e Potenciais Benefícios para a Indústria de Defesa Brasileira. In: Congresso Acadêmico sobre Defesa Nacional. Proceedings [...]. Pirassununga, SP: Escola Superior de Defesa (ESD), 2023. Available at: https://doi.org/10.5281/zenodo.10144605. ②

NAVARRO, I.*; PATRIKAR, J.*; **DANTAS, J. P. A.**; BAIJAL, R.; HIGGINS, I.; SCHERER, S.; OH, J. SoRTS: Learned Tree Search for Long Horizon Social Robot Navigation. IEEE Robotics and Automation Letters, v. 9, n. 4, p. 3759–3766, 2024. ⑥

LIMA, L. S.; GIANNICO, R. H.; BRITO, D. D. C.; DANTAS, A. G. S.; **DANTAS, J. P. A.** Aprendizado de Máquina para a Otimização da Obtenção de Resultados em Simulações de Defesa Aeroespacial. In: Congresso Acadêmico sobre Defesa Nacional. Proceedings [...]. Rio de Janeiro, RJ: Escola Superior de Defesa (ESD), 2024. ⑤

VISCARDI, MARIO; **DANTAS, J. P. A.**; GERALDO, D.; PASSARO, A. Manobra Winding: Defesa Contra Mísseis Passivos e Semiativos Superfície-Ar. Spectrum - The Journal of Operational Applications in Defense Areas, v. 25, n. 1, p. 12–17, out. 2024. Available at: https://spectrum.ita.br/index.php/spectrum/article/view/403. ⑤

PATRIKAR, J.; **DANTAS, J.**; MOON, B.; HAMIDI, M.; GHOSH, S.; KEETHA, N.; HIGGINS, I.; CHANDAK, A.; YONEYAMA, T.; SCHERER, S. Image, speech,

*Equal contribution.

and ADS-B trajectory datasets for terminal airspace operations. Scientific Data, Nature Publishing Group UK London, v. 12, n. 1, p. 468, 2025. ⑥

MURÇA, M. C. R.; MAXIMO, M. R. O. A.; **DANTAS, J. P. A.**; SZENCZUK, J. B. T.; LIMA, C. R.; CARVALHO, L. O.; MELO, G. A. Open Machine Learning Models for Actual Takeoff Weight Prediction. Journal of Open Aviation Science, v. 3, n. 2, Apr. 2025. Available at: `https://journals.open.tudelft.nl/joas/article/view/7963`. ⑤

DANTAS, J. P. A.; MAXIMO, M. R. O. A.; YONEYAMA, T. Autonomous Aircraft Tactical Pop-Up Attack Using Imitation and Generative Learning. IEEE Access, v. 13, p. 81204–81217, 2025. ⑥

LIMA, L. S.; ROCHA, R. D.; GIANNICO, R. H.; BRITO, D. D. C.; **DANTAS, J. P. A.** DroneSwarm2D: Um Simulador de Enxame de Drones Autônomos para o Estudo de Táticas Defensivas Distribuídas. In: Congresso Acadêmico sobre Defesa Nacional. Proceedings [...]. Resende, RJ: Escola Superior de Defesa (ESD), 2025. (The work was accepted and is pending publication in the proceedings at the time of writing this thesis.) ②

SILVA, S. R.; GOMES, V. C. F.; ARANTES, A. O.; CAETANO, A. F. M.; COSTA, V. L. D. B.; SAMERSLA, A. R.; MEDEIROS, F. L. L.; FERREIRA, Y. D.; AQUINO, M. R. C.; **DANTAS, J. P. A.** AsaFG: A Human-in-the-Loop Integration Module for Air Combat Simulations. IEEE Access, 2025. (The manuscript is under review at the time of writing this thesis — first round.) ②

Furthermore, **two** software registration certificates have been received for the following frameworks: one from the Center for Technology Transfer at Carnegie Mellon University in the United States for the AI Pilot and another from the National Institute of Industrial Property (INPI) in Brazil for the ASA framework. The softwares are:

ALOOR, J. J.*; BAJIAL, R.*; **DANTAS, J. P. A.***; HIGGINS, I.*; NAVARRO, I.*; OH, J.*; PATRIKAR, J.*; SCHERER, S.* AI Pilot: Close-Proximity Safe and Seamless Operation of Manned and Unmanned Aircraft in Shared Airspace. 2023. Software registration certificate, Carnegie Mellon University, Center for Technology Transfer, Document 2023-102. ②

AQUINO, M. R. C.*; ARANTES, A. O.*; CAETANO, A. F. M.*; COSTA, A. N.*; **DANTAS, J. P. A.***; FERREIRA, Y. D.; GERALDO, D.*; GOMES, V. C. F.*; KUROSWISKI, A. R.*; MEDEIROS, F. L. L.*; SAMERSLA, A. R.*; SANTOS,

*Equal contribution.

D. S.*; SILVA, S. R.* ASA – Aerospace Simulation Environment. 2024. Software registration certificate, Process number BR 51 2024 002174-4. ②

Finally, during this research, **two** complementary works in related areas were completed, and **three** others are currently in progress, including both undergraduate and master's theses, all under the supervision of the author, who served as a co-advisor:

GOBI, G. H.; BOTELHO, P. L. R. AsaGym: a Python Library for Reinforcement Learning in Aerospace Operational Scenarios. Undergraduate Thesis – Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil, 2023. ⑥

COSTA, V. L. D. B. Análise de Dados de Simulação do Ambiente de Simulação Aeroespacial com Enfoque no Planejamento Baseado em Capacidades. Undergraduate Thesis – Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil, 2024. ⑤

LIMA, L. S. Algoritmo Distribuído para Enxame de Drones Defensivos de Pequeno Porte Baseado em Planejamento. Undergraduate Thesis – Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil, 2025. (This work is currently in progress at the time of writing this thesis.) ⑥

LIMA, L. S. Algoritmo Distribuído para Enxame de Drones Defensivos de Pequeno Porte Baseado em Aprendizado de Máquina. Dissertation (Master's) – Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil, 2025. (This work is currently in progress at the time of writing this thesis.) ⑥

NUNES, A. L. Critérios de Parada em Simulações de Defesa: Análise Comparativa e Proposta de Abordagem Estatística. Dissertation (Master's) – Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil, 2025. (This work is currently in progress at the time of writing this thesis.) ⑤

1.3 Limitations

In this thesis, all contributions were developed as part of software-based solutions. Even though some works related to the main subject involve a degree of hardware development, most of the work does not deeply investigate this aspect. This thesis does not explore the creation of all components used for simulations, such as sensors and aircraft, due to the restricted nature of this information. However, the text offers detailed insights into how these components were conceptualized and modeled. It covers the theoretical framework and design principles used in the development process, providing a good understanding of the methodologies. Additionally, the thesis discusses the various software

tools and programming languages employed, as well as the challenges faced during the development phase and how they were addressed.

Experts in the defense field validated the models and components used in the scenarios. Most of these experts are pilots and operational analysts who evaluated the models based on their extensive knowledge and experience, ensuring that the simulations accurately represented real-world conditions and scenarios. The thesis includes the validation techniques and criteria used, as well as feedback and adjustments made based on the experts' evaluations, which will be presented in some parts of the text.

Formal validation with real-world data remains a challenge in defense contexts due to restricted access to information. The classified nature of many defense systems limits the availability of data necessary for validation. Despite these limitations, this thesis discusses alternative strategies, such as using declassified or synthetic data, and emphasizes the importance of collaboration with defense stakeholders, primarily represented by FAB, to enhance simulation reliability. It also explores future directions for addressing validation challenges, including advances in human-in-the-loop systems and integration of autonomous agents with human pilots.

Another important limitation concerns the evaluation metrics adopted. This thesis focuses on metrics specific to decision support systems or the performance of autonomous agents, rather than metrics directly related to end-user outcomes. The formulation of more user-oriented evaluation metrics is recognized as an important research direction and should be explored in future work.

1.4 Thesis Structure

The thesis is organized into **seven** parts, each focusing on different aspects of the research:

Part I, **Preliminary Considerations**, the current part of the thesis, introduces the context and motivation for the research. It outlines the main objectives, contributions, and limitations of the study, setting the stage for the detailed investigations that follow.

Part II, **Simulation Tools and Services**, focuses on presenting the main simulation environment used in this thesis. It discusses the architecture and components of the simulation tool, highlighting the applications of the simulation services and emphasizing their role in advancing digital transformation for FAB.

Part III, **Weapon Systems**, presents the modeling and analysis of weapon systems, covering both air-to-air and surface-to-air missile engagement zones and providing methodologies for simulation, preprocessing, model training, and evaluation. It includes detailed

results and analyses, exploring the effectiveness of machine learning models in predicting engagement zones and estimating the probability of kill for air-to-air missiles.

Part IV, **In-Flight Tactical Systems**, investigates the application of machine learning techniques to enhance decision-making during aerial combat. It explores the development of decision support systems that utilize data collected from simulated scenarios for use in real-time applications, with supervised learning methods applied to missile launch timing, situational awareness improvement, and engagement strategy formulation.

Part V, **Aerospace Data Analytics**, focuses on the extraction, processing, and analysis of aerospace data. It introduces AsaPy, a Python library developed for aerospace simulation analysis, and discusses its structure and applications. This section covers various aspects of data analytics, including the design of experiments, execution control, data analysis, and prediction, demonstrating the utility of AsaPy.

Part VI, **Aerial Autonomous Agents**, explores the development and deployment of autonomous agents for aerial combat using deep reinforcement learning, imitation learning, and generative learning. This section also addresses the modeling and application of these agents in simulated environments, discussing social navigation, proposed metrics, and a user study experiment.

Finally, Part VII, **Final Considerations**, concludes the thesis with a summary of the research findings and suggestions for future research directions. It provides an overview of the advancements in integrating machine learning and simulation technologies for modern aerial warfare.

2 Literature Review

This chapter provides a comprehensive review of the existing literature on the application of simulation and machine learning in air combat, with a special focus on Beyond Visual Range (BVR) air combat. It aims to present an overview of the key methodologies, advancements, and challenges in this field. Examining various studies and their contributions provides the basis for understanding the current state of research and identifying future directions.

The following survey work served as the basis for this chapter:

> **DANTAS, J. P. A.***; COSTA, A. N.*; SCUKINS, E.; MEDEIROS, F. L. L.; ÖGREN, P. Simulation and Machine Learning in Beyond Visual Range Air Combat: A Survey. IEEE Access, v. 13, p. 76755–76774, 2025.

2.1 Summary

Beyond Visual Range (BVR) air combat is an essential part of modern aerial warfare, relying on advanced radar, missile systems, and decision-support technologies. This survey provides a comprehensive overview of methodologies, applications, and challenges in BVR combat, focusing on machine learning and simulation-based techniques. We examine how machine learning enables adaptive tactics to improve behavior recognition and threat assessment to enhance situational awareness. The paper also traces the historical evolution of BVR combat, outlining key engagement phases such as detection, missile launch, and post-engagement assessment. A key focus is on the role of simulation environments in modeling realistic combat scenarios, supporting pilot training, and validating AI-driven decision-making strategies. We analyze state-of-the-art simulation tools, comparing their capabilities and limitations for studying multi-agent coordination and real-time adaptability. This survey's main contributions include descriptions of machine learning applications in BVR air combat, evaluations of simulation tools, identifications of research gaps, and insights into future research directions. This work provides an overview of how traditional

*Equal contribution.

simulation approaches merge with artificial intelligence to enable advanced, effective human and autonomous decision-making systems in dynamic and contested environments.

2.2 Introduction

Beyond Visual Range (BVR) air combat is a key component of modern aerial warfare, characterized by engagements occurring at distances beyond the pilot's visual sight (YANG *et al.*, 2022). It relies heavily on advanced radar systems, long-range missiles, and detection and tracking methods to neutralize adversaries before visual contact (QIAN *et al.*, 2023). As the nature of air combat evolves, BVR engagements have grown in importance, demanding innovative approaches to overcome the challenges posed by long-range confrontations. The strategic significance of BVR air combat lies in its ability to allow forces to strike first while maintaining a tactical advantage (HE *et al.*, 2024). However, the complexity of these engagements requires interdisciplinary technologies — including sensor fusion, target tracking, decision-making algorithms, and missile guidance systems (DANTAS *et al.*, 2023b; DANTAS *et al.*, 2025) — to improve engagement effectiveness, ensure mission success, and enhance pilot Situational Awareness (SA) (SCUKINS *et al.*, 2024).

Within Visual Range (WVR) air combat engagements occur within relatively short ranges, often involving close-range dogfighting maneuvers reliant on agility, speed, and aiming precision (DANTAS, 2018; LIU *et al.*, 2020). In contrast, BVR engagements leverage advanced sensors and long-range missiles to outperform adversaries (DANTAS *et al.*, 2022). Despite this difference, BVR scenarios can transition into WVR combat as aircraft close in, requiring capabilities in both domains (TOUBMAN *et al.*, 2016; HA *et al.*, 2018).

This survey comprehensively overviews state-of-the-art methods and technologies in BVR combat, highlighting recent advancements and strategic approaches. It begins with a historical overview of BVR combat, tracing its evolution from early air-to-air missile systems to modern multi-sensor platforms. This perspective highlights key technological advancements and their impact on engagement strategies. Next, we examine the core phases of BVR engagements, including detection, missile launch, support, and evasive maneuvers, illustrating how methods discussed in this survey contribute to operational effectiveness. We then review key methodological approaches, such as Machine Learning (ML) algorithms for adaptive decision-making in dynamic environments and the role of Artificial Intelligence (AI) in engagements and autonomous tactics. Practical applications range from pilot decision-support systems to Unmanned Aerial vehicle (UAV) operations. Finally, we highlight the importance of simulation tools in tactical development, pilot training, and algorithm validation. Both general-purpose and specialized platforms are discussed in the context of modeling complex combat scenarios.

To our knowledge, this is the first dedicated survey that examines simulation and ML applications in BVR combat. Existing reviews on air combat either provide a general overview or mention BVR only as a secondary topic. Most discussions on ML for long-range engagements are limited to related work sections of individual papers, offering only partial insights without a structured synthesis of methodologies and applications. Unlike previous works, this survey covers research across multiple studies, providing a comprehensive perspective on how ML and simulation enhance decision-making and engagement strategies. In addition, we analyze available simulation tools, highlighting their capabilities, limitations, and suitability for different applications. In this work, we also identify open challenges and research gaps that remain unexplored, offering guidance for future studies.

The key contributions of this survey are:

- A comprehensive review of ML methodologies applied to BVR, detailing their role in decision-making and autonomous tactics
- An analysis of simulation tools, comparing their capabilities and limitations for modeling realistic combat scenarios
- Identification of key challenges in integrating ML and simulation for improved tactical decision-making
- A perspective on research trends, outlining open questions and guiding future advancements in the field

2.2.1 A Brief History of BVR Air Combat

The origins of BVR air combat trace back to the Cold War, an era defined by rapid technological innovation and an escalating arms race. Early radar-guided missiles, such as the AIM-7 Sparrow, provided the first glimpse of long-range engagements, allowing pilots to strike adversaries from unprecedented distances (KARLE; HALL, 1988). However, these early systems suffered from limited accuracy and vulnerability to Electronic Countermeasures (ECM), leading to mixed performance outcomes in real-world scenarios (GUHL, 2013).

In the 1960s and 1970s, the United States and the Soviet Union invested heavily in BVR capabilities, creating more sophisticated missile systems and advanced radar technologies. The AIM-54 Phoenix missile, used by the U.S. Navy's F-14 Tomcat, was one of the first missile systems capable of simultaneously engaging multiple targets at long ranges, revolutionizing air combat doctrine (MOORE, 2011). This missile was designed to

provide air defense against Soviet bombers and their anti-ship missiles, significantly extending the engagement envelope of carrier-based aircraft (GIBBONS; BOTHA, 2015). The introduction of the AIM-54 Phoenix marked a key moment in BVR combat, highlighting the potential of long-range engagements (BRADLEY, 1987). However, the missile's large size and weight and reliance on semi-active radar homing in the initial phase presented operational challenges (LI; LIU, 2016). Despite these drawbacks, the Phoenix's range and speed made it an important weapon in the U.S. Navy's arsenal (KURODA; IMADO, 1990).

The 1980s and 1990s brought transformative progress with the introduction of the AIM-120 AMRAAM (Advanced Medium-Range Air-to-Air Missile). The AMRAAM's active radar homing capability enabled "fire-and-forget" tactics, allowing pilots to launch missiles and maneuver freely while minimizing exposure to enemy fire (LI; LIU, 2016; HUANG *et al.*, 2012). Its compact size, compatibility with various platforms, and enhanced resistance to ECM made it a versatile and reliable weapon (DOERRY, 2014). The AMRAAM's integration into NATO forces underscored the importance of interoperability and standardization in coalition warfare (RAGIRA, 2017).

In the 21st century, advancements in radar, Electronic Warfare (EW), and missile technology have significantly influenced BVR combat. Modern platforms like the F-22 Raptor and F-35 Lightning II utilize Active Electronically Scanned Array (AESA) radars, stealth technology, and advanced sensors to enhance their effectiveness in BVR engagements (GIBBONS; BOTHA, 2015; EVANS, 2000). These aircraft are designed to operate in contested environments, employing stealth to reduce detection and sensor fusion technologies to improve SA (BHARGAVE *et al.*, 2007; SUMMEY; KEKELIS, 1996). Data links and network-centric warfare concepts further support coordination and target-sharing among allied forces, facilitating adaptive engagement strategies (BRAUNLINGER, 2005; WITSKEN, 2002). EW has also played a pivotal role in shaping BVR tactics. Modern aircraft employ advanced countermeasures to disrupt enemy systems and protect themselves from incoming threats (WAN *et al.*, 2019b).

Looking ahead, the future of BVR combat will be driven by technological breakthroughs such as hypersonic missiles, which promise extended engagement ranges and drastically reduced reaction times (ISBY, 2003). Additionally, integrating AI and autonomous systems is expected to revolutionize BVR engagements. Unmanned platforms collaborating with human pilots can enhance decision-making, improve SA, and reduce operational risks (DANTAS *et al.*, 2024; BENNETT *et al.*, 2005).

2.2.2 Phases of BVR Air Combat

The phases of BVR air combat (Figure 2.1) consist of the following steps: detection, threat assessment, tactical maneuvering, missile engagement, and post-engagement

assessment. Advances in sensor technology, AI, and constructive simulations have significantly enhanced these phases, leading to more effective air combat strategies (DANTAS *et al.*, 2022b).

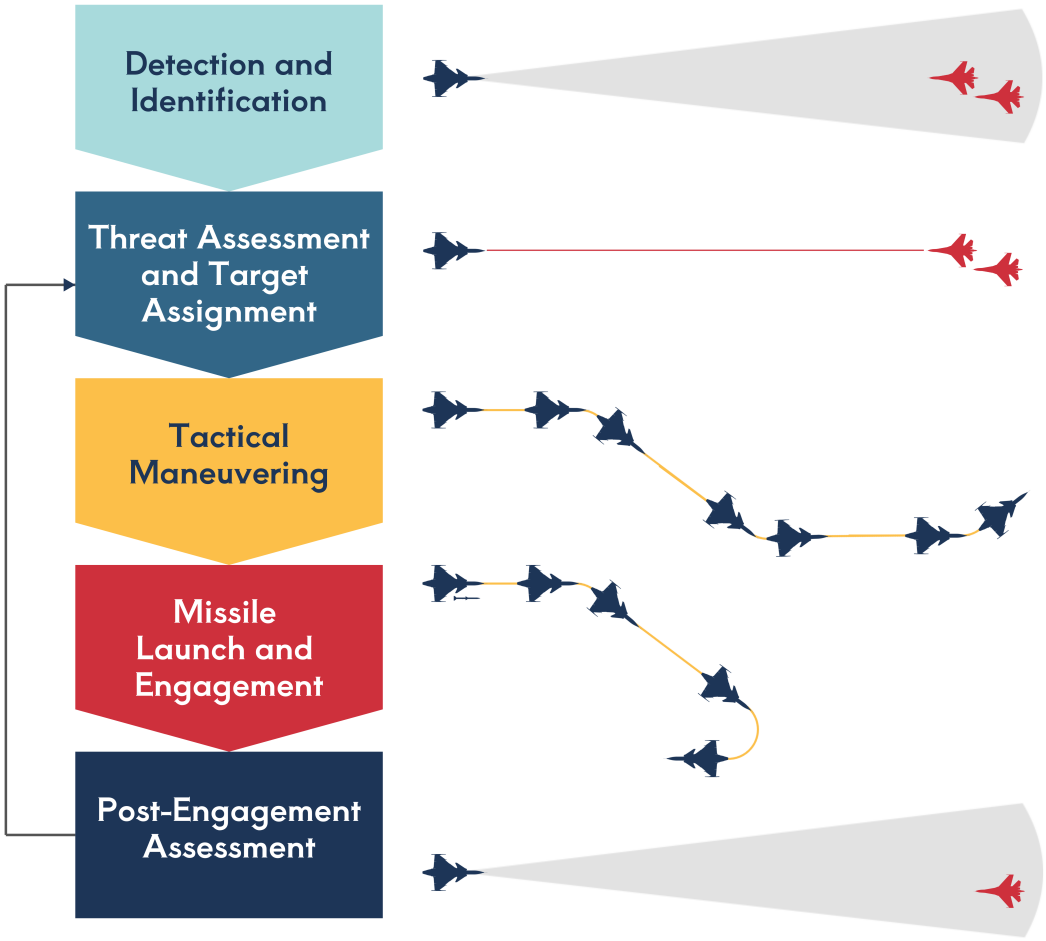


FIGURE 2.1 – Illustration of the main phases of Beyond Visual Range (BVR) air combat.

2.2.2.1 Detection and Identification

The first phase of BVR combat is detecting and identifying enemy aircraft using advanced radar systems. These systems enable long-range target detection, allowing pilots to identify threats well before visual contact (TOUBMAN *et al.*, 2016). Typically, the process begins when an aircraft enters hostile airspace; radar systems are activated to search for opponents, eventually locking onto a detected target (XIANG-JUN *et al.*, 2006). This early detection is critical to achieving a strategic advantage by enabling preemptive actions (HA *et al.*, 2015).

2.2.2.2 Threat Assessment and Target Assignment

Once targets are detected, the next phase involves assessing their threat level and selecting the primary target for engagement. This decision-making process combines human expertise with increasingly sophisticated AI-driven systems. Methods such as Bayesian optimization and Artificial Neural Networks (ANN) have been developed to enhance decision-making (DANTAS *et al.*, 2022b). Effective target selection is essential, as it dictates subsequent engagement strategies and directly influences mission outcomes (HA *et al.*, 2015).

2.2.2.3 Tactical Maneuvering

Following target selection, aircraft perform tactical maneuvers to achieve optimal missile launch positions while evading enemy threats. This phase may include coordinated formations and adaptive flight maneuvers (COSTA *et al.*, 2022). Tactical decisions involve balancing offensive positioning, evasion techniques, and maintaining favorable missile launch parameters (LIMA-FILHO *et al.*, 2022). Aircraft may perform specific maneuvers, such as cranking, to confuse enemy sensors and minimize exposure (RAO *et al.*, 2011b). AI and ML have further refined these tactics, enabling complex, autonomous behaviors in simulations (DANTAS *et al.*, 2021a).

2.2.2.4 Missile Launch and Engagement

The engagement phase is often divided into three stages:

(I) **Launch phase**: The missile is fired once the target enters the Weapon Engagement Zone (WEZ). At this stage, missiles may rely on initial guidance from the aircraft or transition directly to onboard sensors (DAHLBOM, 2013).

(II) **Midcourse support phase**: During this stage, the missile advances toward the target while the launching aircraft reduces its exposure to threats. Depending on the missile type, the aircraft may provide midcourse guidance, or the missile may operate autonomously using its internal systems (DAHLBOM, 2013).

(III) **Terminal phase**: In the final stage, the missile activates its onboard sensors to track and intercept the target autonomously. Modern "fire-and-forget" missiles excel in this phase, as they minimize the need for ongoing guidance, allowing the pilot to reposition or prepare for subsequent engagements (HA *et al.*, 2015; TOUBMAN *et al.*, 2016).

Effective missile engagement strategies weigh the probability of successfully neutralizing the target against the risk to the launching aircraft, making this phase decisive for determining BVR combat outcomes (MANSIKKA *et al.*, 2021a).

2.2.2.5 Post-Engagement Assessment

The final phase involves assessing the outcome of the engagement to determine if additional actions are required. This assessment includes verifying the destruction of the target, re-evaluating the tactical situation, and planning subsequent maneuvers (LIMA-FILHO *et al.*, 2022). Although sometimes underemphasized, post-engagement assessment is essential for ensuring mission success and preparing for future engagements (DANTAS *et al.*, 2022b).

2.2.3 Overview

This survey is based on an extensive literature review conducted using Google Scholar* with the keywords "Beyond Visual Range" or "BVR". The focus was on papers written in English and published within the past decade. Initially, 357 papers containing these terms were identified. After a detailed evaluation, papers that were not in English or only referenced BVR without contributing with substantive research to the field (e.g., citing another work with BVR in the title) were excluded. As of September 27, 2024, the final dataset comprised 120 papers.

All papers were classified according to both their primary methodology and application area, and this structure is consistently used throughout the paper. While many studies incorporate multiple methodologies or cover several application domains, each was assigned to the most prominent category in each classification. Since the terminology used for these categories may vary across the literature, we begin each corresponding section with a clarification of how the term is defined in the context of this survey.

Figure 2.2 illustrates the distribution of papers by year, highlighting the trend in research interest over the past decade. The data reveals an upward trajectory in the number of publications related to BVR air combat, particularly in recent years, underscoring the growing relevance of the topic. The lowest number of publications was recorded in 2017 with just one work, while the highest occurred in 2022 and 2023 with 23 works each. It is worth noting that the lower count for 2024 likely reflects the incomplete availability of papers for this year at the time of this study.

The remainder of this paper is organized as follows. The application areas in the context of BVR air combat are presented in Section 2.3. The methodologies applied to

*https://scholar.google.com

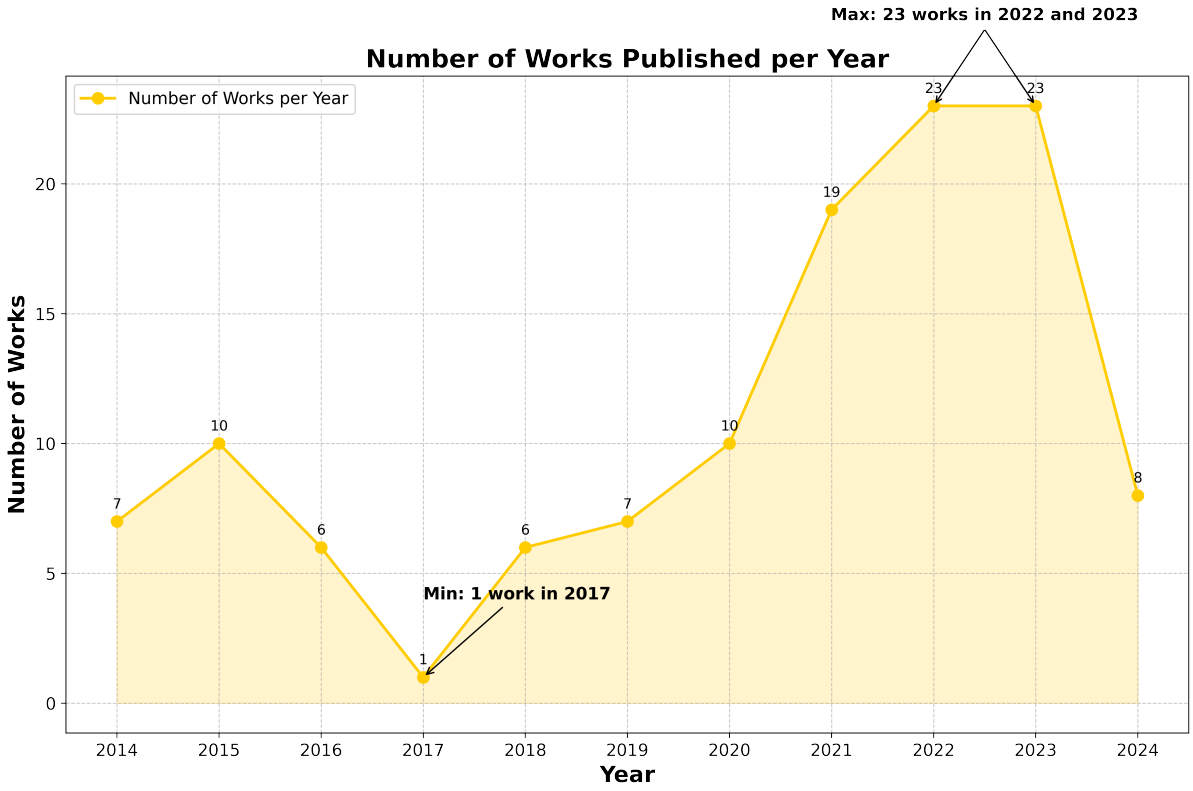


FIGURE 2.2 – Number of articles published per year related to BVR air combat research.

solve problems related to these application areas are described in Section 2.4. Section 2.5 presents simulation environments and tools that have been used to study BVR air combat problems. Section 2.6 describes open challenges regarding BVR air combat simulations. The conclusions of this work are presented in Section 2.7.

2.3 Applications

BVR air combat research spans a diverse set of applications, ranging from autonomous decision-making to multi-agent coordination and pilot training. This section categorizes recent developments across these domains, focusing on how emerging technologies and methods improve tactical performance, adaptability, and mission outcomes.

2.3.1 Autonomous Decision-Making

Autonomous decision-making involves analyzing, selecting, and executing actions that enhance situational control and combat effectiveness. Various approaches were proposed to support this capability, focusing on how agents model tactical behaviors, perform goal reasoning, and assist or replace human pilots in complex scenarios.

A reduction method of tactic features based on granular computing was proposed in (MENG *et al.*, 2014). In (REINISCH *et al.*, 2022) and (AHA, 2018), authors explored behavior modeling in the context of Computer Generated Forces (CGFs) and Goal Reasoning (GR), enabling autonomous systems to make adaptable tactical decisions in rapidly changing scenarios. These capabilities supported the development of autonomous air combat agents that can complement human pilots by taking on specific tasks, such as threat engagement or support maneuvers. Along these lines, (UMMAH *et al.*, 2019) developed a system designed to assist pilots by generating tactical fight strategies.

In (MASEK *et al.*, 2021), a Genetic Programming (GP) framework was presented to discover novel behaviors in air combat scenarios, contributing to more adaptable and unpredictable combat tactics. Furthermore, (YAO *et al.*, 2015a; YAO *et al.*, 2015b) used grammatical evolution to generate adaptive CGFs and Human Behavior Models (HBMs), improving realism and adaptability in training simulations.

The work elaborated in (FU *et al.*, 2014) analyzed the UAV air combat decision process, dividing it into four decision-making phases: situation assessment, attack arrangement, goal assignment, and maneuvering decision. Furthermore, in (QIAN *et al.*, 2023), pilot knowledge was used to create a hierarchical framework that divided air combat into several sub-decision-making systems.

A review of Deep Reinforcement Learning (DRL) methods applied to BVR air combat situations was presented in (WANG *et al.*, 2023b). The autonomous learning of new tactics was addressed in (PIAO *et al.*, 2020), considering a high-fidelity air combat simulation environment. In (DANTAS *et al.*, 2023), an agent based on DRL was developed, being capable of simulating fighter aircraft tactics through self-play, generating novel air combat strategies. This approach enabled human pilots to interact with AI-trained agents, improving their decision-making and adaptability. In (SCUKINS *et al.*, 2024a), a Reinforcement Learning (RL) environment was created aiming at autonomous learning of new air combat tactics and the discovery of new maneuvers.

Many studies also employed RL in one-on-one combat scenarios. For instance, (HE *et al.*, 2023) proposed a self-play training framework to address the action control problem in long-horizon engagements. Research in (JIANG *et al.*, 2022) introduced a DRL-based decision-making algorithm with tailored state and action spaces and an adaptive reward function, demonstrating robustness across diverse confrontation scenarios. In (WEILIN *et al.*, 2018), an improved Q-network enhanced maneuvering decisions by enabling agents to approach opponents from advantageous positions. Similarly, (MAO *et al.*, 2022) presented a DRL-based agent construction method grounded in realistic weapon simulation. Finally, (XIA *et al.*, 2024) developed a hybrid self-play DRL agent capable of maintaining high win rates against a variety of opponents, improving both adaptability and performance.

2.3.2 Behavior Recognition

Behavior recognition is important for understanding and predicting the actions of adversarial agents, informing decision-making, and strategic planning. Several studies explored methods to recognize and predict enemy behaviors under complex and uncertain combat conditions.

An integrated planning and recognition algorithm in (ALFORD *et al.*, 2015) showed that proactive observation gathering accelerates behavior classification. Building on Case-Based Reasoning (CBR), (BORCK *et al.*, 2014; BORCK *et al.*, 2015a; BORCK *et al.*, 2015b) developed a Case-Based Behavior Recognition (CBBR) system that annotated agent behaviors from spatio-temporal features, improving recognition within GR-controlled UAVs. Likewise, (FLOYD *et al.*, 2017a) combined opponent modeling and CBR to identify adversarial team behaviors.

To handle incomplete data, (LIU, 2022) introduced an intention recognition method based on Multi-Granulation Rough Sets (MGRS). The study in (CHEN *et al.*, 2022) fused Dempster-Shafer theory with Deep Temporal Networks for improved classification, while (XIA *et al.*, 2023) used a decision tree and Gated Recurrent Unit (GRU) for state prediction in one-on-one air combat. In (YANG *et al.*, 2022), a hierarchical approach was proposed using Cascaded Support Vector Machines (CSVM) and cumulative features for multi-dimensional target classification.

To recognize tactical intent, (LEI *et al.*, 2023) introduced an attention-enhanced swarm optimization and bidirectional GRU model (A-TSO-PBiGRU) for shift detection. Similarly, (ZHANG *et al.*, 2023) applied Dynamic Bayesian Networks (DBN) to infer causal links between flight states and tactical movements, improving formation recognition and SA.

2.3.3 Guidance and Interception

Guidance and interception mechanisms are essential for increasing the probability of a successful missile engagement, particularly against fast and evasive targets.

Guidance strategies were compared to identify configurations that minimized interception time and maneuver load, offering improved engagement options under varying combat conditions (MEI; WANG, 2021). The interception of hypersonic targets was addressed by enhancing the missile's ability to reach the target at a specific impact angle, improving the conditions for the final engagement phase (WAN *et al.*, 2019a). In unmanned operations, aiming precision in Unmanned Combat Aerial Vehicles (UCAVs) was improved through autonomous guidance techniques, enabling more effective launches against maneuvering

aerial targets (YANG *et al.*, 2020).

Maneuver decisions during missile flight were optimized to support engagement planning and increase success rates in simulated combat scenarios (XIE *et al.*, 2023). Real-time trajectory adjustment was achieved through probabilistic modeling of the Dynamic Attack Zone (DAZ), helping maintain accuracy despite environmental uncertainty (SUN *et al.*, 2020b). Coordination between radars and missiles was improved by cooperative guidance models, which enhanced system-level precision in anti-aircraft defense (LEI *et al.*, 2015).

The influence of data link quality on missile effectiveness was quantified through simulation, showing how update delays and errors impacted seeker activation and overall success (ÖSTROM *et al.*, 2023). For long-range engagements, ignition control and trajectory correction for dual-pulse motor missiles were refined to support effective interception of distant targets (GONG *et al.*, 2023).

2.3.4 Maneuver Planning

Maneuver planning involves calculating a sequence of motion primitives to reach an advantageous tactical situation.

Early work in this area emphasized structured evaluation and decision models. The authors of (LU *et al.*, 2021) introduced a framework comprising a situation evaluation model, a maneuver decision model, and a one-on-one engagement evaluation model. In (YUAN *et al.*, 2016), a tactical decision system was developed, being based on air conditions, threat environment, target weapon performance, and air-combat rules. Incorporating broader situational factors, (YANG *et al.*, 2014) explored Target Assignment (TA) strategies that integrated tactical positioning and weapon capabilities to improve resource allocation.

More recent research focused on learning-based approaches. (HU *et al.*, 2021), (HU *et al.*, 2022) and (HU *et al.*, 2022) applied DRL to maneuver planning, improving threat avoidance and target engagement in dynamic scenarios. These works considered different initial engagement conditions to train more adaptable agent models. In (QIU *et al.*, 2020), an autonomous maneuver strategy was developed using the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, focusing on missile evasion in one-on-one engagements. The study in (WEI *et al.*, 2023a) designed a maneuver decision-making method based on relative azimuths and distances between opponents. Finally, (ZHANG *et al.*, 2022) combined DRL with Monte Carlo Tree Search (MCTS) to investigate maneuver planning without relying on prior pilot knowledge or value-based functions.

2.3.5 Missile Engagement

Missile engagement and evasion require optimizing both launch timing and maneuver strategies to maximize offensive impact and survivability.

On the offensive side, (DANTAS *et al.*, 2022b) employed Supervised Learning (SL) to estimate optimal missile launch moments, enhancing mission effectiveness. For stealthy operations, (LIU *et al.*, 2022b) introduced a radar blind zone maneuver control method, enabling undetected approaches, while (KUNG; CHIANG, 2015) analyzed missile capture areas and minimum evasive ranges. This was done to identify optimal launch distances and defensive strategies in coordinated team air combat scenarios.

On the defensive side, (YANG *et al.*, 2020) proposed an autonomous evasive maneuver strategy for a UCAVs using a hierarchical multi-objective evolutionary algorithm to increase survivability. In (ALKAHER; MOSHAIOV, 2016), the missile evasion problem was modeled as a two-team zero-sum differential game, where one aircraft aimed to increase its distance from an incoming missile, while simultaneously closing in on another non-aggressive target.

In UCAVs operations, (LI *et al.*, 2022c) introduced a cooperative occupation method based on WEZ, improving coordinated positioning. Lastly, (JIA *et al.*, 2023) addressed the challenge of information blindness after the launch of air-to-air missiles.

2.3.6 Multi-Agent Coordination

Multi-agent coordination combat enables cooperative decision-making, joint tactical execution, and improved responsiveness among autonomous platforms. Applications range from coordinated engagement strategies to dynamic team reconfiguration and human–AI teaming.

Tactical strategies for multiple UAVs were applied to decompose air-to-air confrontations into one-on-one cases, improving maneuver efficiency and engagement success (KANG *et al.*, 2019). Cooperative position allocation and target assignment were modeled as a zero-sum game, where a hybrid Double Oracle and neighborhood search algorithm improved solution quality under time constraints (MA *et al.*, 2020).

Extensions to the Tactical Battle Manager incorporated distributed discrepancy detection to enhance mission outcomes in (KARNEEB *et al.*, 2016). GR techniques were advanced through a character-oriented framework that improved coordination among autonomous agents operating with limited communication (SAMSONOVICH; AHA, 2015). To support human-AI teaming, the AlphaMosaic architecture integrated human feedback into Battle Management Systems (BMS), enabling trust-based collaboration in dynamic missions

(ALBARADO *et al.*, 2022).

Swarm intelligence was adapted to fixed-wing UCAVs platforms, enabling behaviors such as formation flight, self-reorganization, and dynamic adaptation after losses (BAKIRCI; OZER, 2023). A centralized AI planning system was used to coordinate multi-agent mission plans with full observability and verifiability (CHAO *et al.*, 2023). War game simulations were employed to test coordinated fleet behavior, with tactical parameters optimized to enhance mission outcomes in matched-force engagements (JIA; KIANG, 2022).

Tactical formations of UAVs were optimized against uncertain enemy behavior, using simulation-based evaluations (LIMA-FILHO *et al.*, 2022). A two-stage cooperative pursuit strategy was introduced, combining luring tactics and Hybrid A* path planning to increase interception success (RU *et al.*, 2024). Adaptive guidance methods were designed to improve UAV occupancy using a multi-objective function and the GDT-SOS metaheuristic (YIN *et al.*, 2022).

Hierarchical RL architectures enabled multi-agent teams to learn both low-level and high-level tactics through self-play and scenario decomposition (HE *et al.*, 2024). Multi-agent Proximal Policy Optimization (PPO) was applied to UCAVs coordination, integrating domain knowledge into the reward structure and achieving improved performance in combat drills (LIU *et al.*, 2022a).

A graph-based reasoning model combined expert knowledge with graph neural networks to model complex collaboration patterns and simplify decision-making in large-scale engagements (PIAO *et al.*, 2023). An algorithm based on adversarial self-play and hierarchical policy gradients was used to learn emergent strategies that outperformed expert baselines (SUN *et al.*, 2021). Deep Deterministic Policy Gradient (DDPG) was applied in swarm maneuvering, where inter-agent cooperation and target engagement were jointly learned (WANG *et al.*, 2021). Finally, neural networks and artificial potential fields were combined to support cooperative path planning against adaptive adversaries (ZHANG *et al.*, 2018).

2.3.7 Operational Analysis

Operational Analysis (OA) involves using simulations, models, and metrics to evaluate combat effectiveness, support tactical planning, and inform operational decisions.

Stochastic game-based models were applied to analyze multi-aircraft engagements under uncertainty, providing insights into coordination strategies and missile allocation in BVR scenarios (HA *et al.*, 2015; HA *et al.*, 2018). Simulations involving human operators were used to assess pilot and team performance under realistic combat conditions, focusing on compliance with operational procedures, cognitive workload, and shared SA

(MANSIKKA *et al.*, 2021b; MANSIKKA *et al.*, 2021c; MANSIKKA *et al.*, 2021a).

Several studies presented simulation platforms for training, tactical testing, and operational planning. These included a tactical-level air combat simulation system developed to support intelligent decision-making (LIU *et al.*, 2020), the ASA framework designed for evaluating military scenarios in the Brazilian Air Force (DANTAS *et al.*, 2022a), and its cloud-based extension, ASA-SimaaS, which enabled scalable and autonomous simulation services (DANTAS *et al.*, 2023a). AsaPy complemented these tools by offering post-simulation analysis capabilities using statistical and ML methods (DANTAS *et al.*, 2024).

To assess fleet effectiveness, system-of-systems (SoS) simulations were used to evaluate aircraft design, platform interoperability, and mission-level success indicators such as survivability and weapon usage (DIETL *et al.*, 2023). Parametric studies investigated how variables like radar cross-section, missile range, flight altitude, and communication delays affected outcome metrics such as probability of kill and overall combat effectiveness (HAOYU *et al.*, 2018; YE *et al.*, 2023; SUSENO; SASONGKO, 2016). The influence of agent behavior on simulation credibility was explored through agent-based models, enhancing validation methods in both symmetric and asymmetric BVR scenarios (KUROSWISKI *et al.*, 2023).

Communication systems were addressed through the design of a dual-mode protocol that adapted to network conditions in cooperative air combat operations (LUO *et al.*, 2014). Simulation architectures emphasized scalability and flexibility, underlining the need for multi-agent systems capable of managing AI-driven entities and distributed decision-making processes (NEWTON *et al.*, 2021). Meanwhile, validation environments for high-dynamic flight conditions were developed to assess electro-optical system performance under large maneuvers (ZHANG *et al.*, 2021).

Network-centric operations were modeled to analyze combat effectiveness under varying levels of synergy between sensors, command structures, and fire control systems (ZOU-JIE; WUWEI, 2016). Finally, decision-support tools were proposed based on Multi-Criteria Decision-Making (MCDM) (LI; LIU, 2015), relevance vector machines (HUANG *et al.*, 2020), and improved Extreme Learning Machine (ELM) models (WANG *et al.*, 2023a), offering quantitative evaluations of fighter aircraft and tactical configurations.

2.3.8 Pilot Training

Pilot training focuses on enhancing readiness and effectiveness through advanced simulation environments, performance measurement, and adaptive learning techniques. These studies aim to improve decision-making and SA in complex combat scenarios.

The approach outlined in (MANSIKKA *et al.*, 2020) provided insights into retrospective

performance evaluation to identify areas for improvement, informing targeted training adjustments. Similarly, (ARONSSON *et al.*, 2019) explored behavioral modeling to enhance pilot decision-making under high-stress conditions, improving the realism of training exercises.

The integration of Live, Virtual, and Constructive (LVC) environments, as discussed in (ARONSSON *et al.*, 2023), offers comprehensive training scenarios that combine real and simulated elements to create more realistic and immersive training. This enables pilots to experience diverse combat situations, improving adaptability under varying conditions. To further refine training outcomes, (VIRTANEN *et al.*, 2022) proposed a performance-weighting system to optimize training outcomes, ensuring that pilots meet competency benchmarks efficiently.

The survey on adaptive training methodologies in (GORTON *et al.*, 2024) highlighted advancements in AI-driven systems that personalize training content based on pilot performance. Building on this, (TOUBMAN *et al.*, 2016) and (TOUBMAN, 2019) discuss methods for rapidly adapting air combat behaviors and validating training simulations. These studies aimed to ensure that simulation systems accurately reflect real-world combat dynamics, providing practical tools to directly impact pilot training effectiveness by improving responsiveness and situational understanding.

2.3.9 Situational Awareness

SA is essential for understanding the tactical environment, including the positions, actions, and intentions of both friendly and enemy aircraft. Effective SA supports informed decision-making in engagement, positioning, and evasion, ultimately enhancing combat effectiveness and survivability.

In (SHI *et al.*, 2020), methodologies for real-time data processing were explored, enabling pilots to interpret complex information efficiently. Expanding SA to a team level, (MANSIKKA *et al.*, 2021d) demonstrated the benefits of collaborative data sharing for mission coherence and performance.

For threat assessment, (YOU *et al.*, 2019) and (WANG *et al.*, 2022) discuss methods to determine enemy WEZ, providing pilots with spatial awareness to avoid or confront threats strategically. Real-time threat analysis tools, such as those developed in (GAO *et al.*, 2022), continuously updated situational data, ensuring that pilots can adapt their tactics accordingly. Furthermore, (SHI *et al.*, 2021), (CAO *et al.*, 2021), and (WANG *et al.*, 2023) integrated the prediction of target intention into the assessment of threats, analyzing battlefield situations and establishing threat index systems.

AI-driven approaches to SA, like those discussed in (SCUKINS *et al.*, 2023; SCUKINS

et al., 2024b), applied ML to threat detection, helping pilots anticipate and respond to potential threats more quickly. Additionally, Monte Carlo-based methods for probabilistic assessment, such as in (SCUKINS *et al.*, 2023), enable pilots to navigate uncertain situations with better-informed risk management. Research in (DANTAS *et al.*, 2021a) proposed an engagement decision support tool based on the Defensive Counter Air (DCA) operational metric. The use of Deep Neural Networks (DNNs) to perform the estimation of the WEZ maximum launch range was analyzed in (DANTAS *et al.*, 2021b).

Additional decision support systems like (LIMA-FILHO *et al.*, 2021) employed onboard sensor data and neural networks to assess shoot-down probabilities in real-time. In (SCUKINS *et al.*, 2024), a method to estimate maneuver flexibility under adversarial conditions was introduced, aiding formation-level decision-making.

2.3.10 Target Assignment

TA involves efficiently allocating resources, such as air-to-air or surface-to-air missiles and aircraft, to neutralize enemy threats. This process requires strategies to optimize engagements while minimizing resource expenditure and maximizing mission success.

Several studies focused on target allocation methodologies to improve combat effectiveness. In (DING *et al.*, 2018; LI *et al.*, 2022a; LI *et al.*, 2022b), the authors discuss Multi-Target Assignment (MTA) strategies that dynamically assign missiles and aircraft to multiple targets. Likewise, (LEI *et al.*, 2022) proposed methods for assigning multiple friendly aircraft to a set of enemy targets, focusing on coordinated attack strategies to improve the efficiency of engagement.

Studies such as (PENGYUAN *et al.*, 2014; ZHAO *et al.*, 2020) examined algorithms that determine the optimal pairing of weapons to threats based on mission objectives and constraints, aiming to maximize kill probability while preserving resources. In (ZHOU *et al.*, 2014), the authors refined this process by introducing an improved assignment model that incorporates target priority and engagement timing.

Lastly, (YAO, 2021) investigated a hybrid approach, combining optimization techniques with real-time tactical adjustments to adapt to evolving combat conditions.

2.4 Methodologies

This section outlines the methodological foundations used to address key challenges in BVR air combat, where dynamic, uncertain, and adversarial environments demand robust and adaptive decision-making. Techniques span a wide spectrum—from data-driven methods like supervised, unsupervised, and reinforcement learning to structured

reasoning approaches such as control theory, graphical models, and game theory. Each subsection explores the specifics of each methodology, illustrating their application and possible advantages.

2.4.1 Control Theory

Control theory employs mathematical models to guide and influence the behavior of dynamic systems, making it especially valuable in BVR air combat for precise aircraft and missile guidance, interception of maneuvering targets, and strategic positioning under uncertainty.

Guidance laws were developed to optimize interception effectiveness by balancing engagement time and maneuver load (MEI; WANG, 2021), while model predictive strategies improved midcourse guidance in hypersonic scenarios with terminal-angle constraints (WAN *et al.*, 2019a).

Stealthy engagement strategies integrated sliding mode control techniques with electronic support measures, enabling adaptive transitions between stealth and aggressive maneuvers to exploit enemy radar vulnerabilities (LIU *et al.*, 2022b). Autonomous UCAVs engagements similarly benefited from adaptive fuzzy control within a Model Predictive Control (MPC) framework, enhancing precision and responsiveness against agile targets (YANG *et al.*, 2020). Fuzzy logic was also applied to recommend combat modes using basic situational inputs (UMMAH *et al.*, 2019).

2.4.2 Evolutionary Algorithms

Evolutionary Algorithms (EA) are a class of optimization techniques inspired by biological evolution, capable of solving complex, high-dimensional, and non-convex problems.

In behavior modeling, genetic programming evolved Behavior Trees (BTs) to discover novel air combat strategies under EW effects (MASEK *et al.*, 2021). Grammatical evolution was used to generate adaptive CGFs and HBMs by encoding subject matter expert (SME) knowledge in modular BTs, enabling dynamic responses in training simulations (YAO *et al.*, 2015a; YAO *et al.*, 2015b).

Differential evolution algorithms introduced adaptive parameter strategies to improve convergence and robustness in tactical planning (XIE *et al.*, 2023). Multi-objective evolutionary approaches applied hierarchical and Pareto-based methods to balance competing goals (YANG *et al.*, 2020).

Hybrid evolutionary methods, including genetic algorithms enhanced by simulated annealing and discrete evolutionary strategies, were frequently applied to resource allocation

and assignment tasks, where mechanisms like adaptive crossover, mutation rates, and disturbance strategies promote efficient convergence and solution diversity (ZHAO *et al.*, 2020; ZHOU *et al.*, 2014).

2.4.3 Game Theory

Game theory offers mathematical tools for analyzing strategic interactions in adversarial and cooperative settings. In BVR air combat, it supports optimal strategies for navigation, engagement, evasion, and team coordination.

In (ALKAHER; MOSHAIOV, 2016), differential game theory was used to model dynamic zero-sum interactions with closed-loop control to ensure safe maneuvering under threat. On the other hand, stochastic games enabled sequential decision-making under uncertainty through subgame decomposition and equilibrium analysis (HA *et al.*, 2015; HA *et al.*, 2018).

Pursuit-evasion games defined missile capture and evasion ranges using utilitarian formulations to guide tactical behavior (KUNG; CHIANG, 2015). For cooperation, consensus-based algorithms combined with auctions and matrix games enabled scalable and efficient resource allocation (LI *et al.*, 2022b; MA *et al.*, 2020).

Additionally, min-max approaches further simplified multi-agent engagements into pairwise confrontations, supporting fast and systematic tactical decisions (KANG *et al.*, 2019).

2.4.4 Goal Reasoning

GR enables autonomous agents to dynamically deliberate, adapt, and reprioritize goals based on real-time context, making it well-suited for adversarial environments like BVR air combat. As discussed in the overview by (AHA, 2018), GR supports deliberative autonomy, dynamic goal management, and contributes to AI safety.

CBBR methods enhance GR by annotating adversarial behaviors from spatio-temporal data, improving adaptation under partial observability (BORCK *et al.*, 2014; BORCK *et al.*, 2015a). (BORCK *et al.*, 2015b) extended this with the Policy and Goal Recognizer (PaGR) system, capable of inferring and adjusting assumptions about opponent goals and strategies. Likewise, (FLOYD *et al.*, 2017a) combined case retrieval with learned classifications to recognize adversarial team behaviors, while (ALFORD *et al.*, 2015) integrated behavior recognition with active planning in a Partially Observable Markov Decision Process (POMDP) framework to improve recognition efficiency.

Distributed GR further enabled decentralized adaptation by detecting discrepancies between expected and observed behaviors (KARNEEB *et al.*, 2016). Narrative Team Plan-

ning (NTP) supported decentralized coordination through hierarchical narrative-based goal structures, improving resilience under conditions of limited communication and uncertainty (SAMSONOVICH; AHA, 2015).

2.4.5 Graphical Models

Graphical models support probabilistic reasoning under uncertainty by capturing structured relationships among variables. In BVR air combat, techniques such as Bayesian Networks (BNs), DBN, and Influence Diagrams address challenges in threat assessment, recognition, and decision-making.

DBN were used to infer maneuver and formation tactics by modeling causal links between aircraft states and actions (ZHANG *et al.*, 2023), while BNs captured missile guidance uncertainty by linking kinematic parameters to the Dynamic Attack Zone (SUN *et al.*, 2020b).

To improve scalability, Multi-Entity Bayesian Networks (MEBNs) offered modular representations of battlefield entities and interactions (SHI *et al.*, 2020). Influence diagrams combined situation assessment with maneuver decisions to evaluate tactical alternatives and equipment effectiveness under operational constraints (LU *et al.*, 2021).

2.4.6 Human Performance Evaluation

Human performance evaluation methodologies assess cognitive load, situational awareness, decision-making, procedural adherence, and team effectiveness under the complex and time-critical conditions characteristic of BVR air combat. Rather than relying solely on outcome metrics (e.g., mission success), recent methodological advances emphasize internal cognitive states, team dynamics, and alignment with doctrinal procedures.

Workload was assessed using modified NASA-TLX methods, incorporating enhanced weighting techniques such as Swing and the Analytic Hierarchy Process (AHP) to improve interpretability (VIRTANEN *et al.*, 2022). Additionally, Retrospective Verbal Probing (RVP) was employed to elicit structured and non-intrusive insights into pilots' mental models after the mission (MANSIKKA *et al.*, 2020).

Team assessments incorporated the Critical Decision Method to evaluate shared situational awareness and its tactical impact (MANSIKKA *et al.*, 2021d), supported by multidimensional frameworks integrating taskwork, normative behavior, and workload within LVC simulations (MANSIKKA *et al.*, 2021b).

Normative Performance (NP) was assessed via structured observer scoring to ensure doctrinal alignment during debriefings (MANSIKKA *et al.*, 2021c), while human-agent in-

teraction models were validated for reliability and relevance in adaptive training systems (TOUBMAN, 2019).

2.4.7 Modeling and Simulation

Modeling and Simulation (M&S) provide a methodological foundation for representing, analyzing, and validating the complex dynamics of BVR air combat. These techniques enable the abstraction of aircraft, weapons, sensors, and command structures into executable models that support experimentation, decision-making, and system development across both tactical and strategic levels.

Several studies introduced simulation frameworks that support experimentation and decision-making. ASA (DANTAS *et al.*, 2022a; DANTAS *et al.*, 2023a) offers a scalable, object-oriented environment with distributed execution and runtime model loading, while AsaPy (DANTAS *et al.*, 2024) supports statistical and machine learning-based post-processing. Other contributions focused on AI-integrated systems, such as AlphaMosaic (ALBARADO *et al.*, 2022), a battle management architecture that supports human–AI teaming through trust-aware decision loops in dynamic BVR environments. The study in (LIU *et al.*, 2020) introduced a tactical-level simulation system that generates data for intelligent decision-making models across human–human, human–machine, and machine–machine configurations. Similarly, (SUSENO; SASONGKO, 2016) presented a modeling environment for analyzing tactics and maneuver effectiveness using the probability of kill (Pk) via a missile launch envelope model. Complementing these were LVC-based environments that support the iterative development of tactics, techniques, and procedures (TTPs) (MANSIKKA *et al.*, 2021a), including pilot co-designed training tools for asymmetric roles and after-action review visualization (ARONSSON *et al.*, 2019; ARONSSON *et al.*, 2023). Additional frameworks were proposed for electro-optical system testing under high-dynamic flight simulation (ZHANG *et al.*, 2021) and for scaling multi-agent simulation environments with AI-enabled entities (NEWTON *et al.*, 2021).

Agent-based M&S (ABMS) examined how behavioral variation influences engagement outcomes in (KUROSWISKI *et al.*, 2023). Likewise, SoS simulations linked aircraft design variables to mission-level effectiveness by modeling interactions between manned fighters and cooperative unmanned platforms (DIETL *et al.*, 2023). Symbolic and AI-integrated systems extend simulation's role into planning and autonomy. High-level planners decomposed multi-agent objectives into coordinated execution plans (CHAO *et al.*, 2023), while swarm-intelligent UCAVs platforms simulated full-mission coordination, formation management, and mid-flight reconfiguration in response to unit loss or dynamic updates (BAKIRCI; OZER, 2023).

Other studies used simulation to evaluate operational variables. These included the

effect of data link quality on seeker lock and Pk (ÖSTROM *et al.*, 2023), the impact of stealth and missile range on 1-vs-1 effectiveness (HAOYU *et al.*, 2018), and cooperative radar-missile guidance modeled through Monte Carlo estimation (LEI *et al.*, 2015). Threat assessment models incorporated both expert and data-driven weighting schemes alongside intent prediction and event impact modeling (SHI *et al.*, 2021; WANG *et al.*, 2023). The influence of communication delays on fleet-level combat effectiveness was explored through threat matrix modeling and adjudication methods (YE *et al.*, 2023). Network-centric air combat was also addressed through communication protocols (LUO *et al.*, 2014) and synergy models (ZOUJIE; WUWEI, 2016) to assess decentralized cooperation under bandwidth and latency constraints.

Beyond performance evaluation, M&S also supports behavior modeling and interface testing. BTs were used to structure decision logic for CGFs in virtual BVR simulations (REINISCH *et al.*, 2022), while testbeds for electro-optical tracking under high-maneuver dynamics validated hardware behavior under simulated aerial conditions (ZHANG *et al.*, 2021). Broader overviews of UAV air combat decision modeling highlighted simulation's role in structuring multi-stage decision pipelines (FU *et al.*, 2014), and virtual expert systems for tactic generation model high-level reasoning across BVR and WVR conditions (YUAN *et al.*, 2016).

Additional information on the application of ML to model air combat behavior, covering both WVR and BVR engagements, was presented in (GORTON *et al.*, 2024).

2.4.8 Optimization

Optimization aims to make the most effective use of available resources, actions, or strategies to achieve mission objectives under constraints. In BVR air combat, it provides a general framework for decision-making tasks such as engagement timing, resource allocation, and path planning. Methods span from classical optimization to bio-inspired heuristics, often tailored to dynamic, high-dimensional, and time-critical environments.

A wide range of swarm intelligence algorithms were adapted to the air combat domain. Variants of Particle Swarm Optimization (PSO) were used to handle missile-target assignment (DING *et al.*, 2018), cooperative UCAVs occupation modeling through discrete PSO (LI *et al.*, 2022c), and air defense target allocation under real-time constraints (PENGYUAN *et al.*, 2014). Additional extensions included Particle-Pair Swarm Optimization (P2SO) to co-optimize fleet parameters in symmetric war games (JIA; KIANG, 2022), and Stochastic Dominant Learning Pigeon-Inspired Optimization (SDLPIO), which integrated payoff-based decision-making for multi-UAV target allocation (LEI *et al.*, 2022).

Multi-objective optimization and MCDM methodologies explicitly handle trade-offs

among competing objectives or decision criteria. Techniques like Gradient Descent–Truncated Symbiotic Organisms Search (GDT-SOS) systematically balanced multiple tactical parameters (e.g., distance, speed, angles) to derive optimized solutions for trajectory planning and positional guidance (YIN *et al.*, 2022). Similarly, MCDM approaches such as the integration of AHP and Kullback–Leibler divergence (KL-AHP), as well as TOPSIS, systematically evaluated and ranked alternative solutions according to multiple criteria, enabling strategic and tactical decision-making under uncertainty (LI; LIU, 2015). Hybrid A* path planning, guided by multi-objective considerations, enabled coordinated UAV trajectories (RU *et al.*, 2024). Models based on geometric constraints and engagement zones were also used to derive best attack positions in multi-target scenarios (YANG *et al.*, 2014), while methods like the Golden Section Search were applied to real-time missile zone computation (YOU *et al.*, 2019).

In distributed allocation, asynchronous consensus-based auction algorithms coordinated missile-target assignments across UAV teams, improving solution quality under limited communication (LI *et al.*, 2022a).

In more tactical-level formation planning, metaheuristic comparisons were used to optimize UAV swarm configurations under uncertainty. In (LIMA-FILHO *et al.*, 2022), six metaheuristics were evaluated in a war game setup to determine robust formation strategies against an opposing force, incorporating variability in enemy location and engagement potential.

Finally, in (SCUKINS *et al.*, 2023), Monte Carlo Tree Search (MCTS) was integrated with convex optimization to determine safe missile guidance trajectories in adversarial environments, supporting real-time pilot decision-making.

2.4.9 Reinforcement Learning

RL techniques may enhance tactical creativity and efficiency by autonomously learning and adapting strategies through interactions with dynamic environments, leveraging algorithms for exploration, self-play, and expert knowledge integration. It is also one of the most common research methods in the BVR research field.

Although BVR air combat often involves multiple units per team, researchers frequently simplify the setting to one-on-one engagements to isolate tactical decision-making. The study in (TOUBMAN *et al.*, 2016) applied RL to generate adaptive behaviors for CGFs, while (SCUKINS *et al.*, 2023) used PPO to model evasive behavior against incoming missiles, with a reward function based on the smallest distance between the agent and the missile. In (WEILIN *et al.*, 2018), an Improved Q-network (IQN) was used to balance exploration and exploitation. The research in (QIU *et al.*, 2020) introduced a modified

TD3 algorithm for maneuver and missile engagement strategies, and (JIANG *et al.*, 2022) developed a Dueling Double Deep Q-network (D3QN) for decision-making across varied engagement conditions.

Several works explored improvements in learning efficiency and realism, such as (HU *et al.*, 2021), which proposed an enhanced Deep Q-network (DQN) with LSTM-based perception layers for maneuver planning. The work in (HU *et al.*, 2022) introduced Dynamic Quality Replay (DQR) to improve policy learning from confrontation demonstrations, with Soft Actor-Critic (SAC) outperforming other methods. In (GONG *et al.*, 2023), DRL was integrated with singular perturbation theory to generate ignition and acceleration commands for dual-pulse air-to-air missiles, enhancing long-range guidance.

Agent construction and training methods were also explored in (MAO *et al.*, 2022), where a SAC-based agent was trained using curriculum learning across staged tasks—flight control, guided engagement, and defeating expert systems. Similarly, (XIA *et al.*, 2024) proposed a hybrid self-play DRL strategy, allowing agents to train against both expert systems and delayed self-play opponents to prevent local optima. In (ZHANG *et al.*, 2022), authors combined RL with MCTS to learn maneuver strategies without reliance on hand-crafted reward functions or expert features.

While one-on-one scenarios dominate early exploration, multi-agent reinforcement learning enables improved survivability, shared situational awareness, and tactical synergy. Multi-Agent Proximal Policy Optimization (MAPPO) and Hierarchical Framework Embedding Expert Knowledge (H3E) frameworks incorporate hierarchical structures and expert guidance to improve coordination and efficiency across agents (LIU *et al.*, 2022a; QIAN *et al.*, 2023). In (HU *et al.*, 2022), a dual-UAV cooperative air combat strategy was proposed using prioritized sampling and a discretized action space, demonstrating effective maneuver planning and obstacle avoidance. Swarm-based strategies using DDPG allow groups of agents to perform cooperative maneuvers in continuous control spaces (WANG *et al.*, 2021). Multi-agent decision networks, including hierarchical policy gradients (SUN *et al.*, 2021) and improved Neural Fictitious Self-Play (NFSP) (HE *et al.*, 2023; HE *et al.*, 2024), support the emergence of high-level strategies. Meanwhile, (PIAO *et al.*, 2020) introduced Key Air Combat Event Reward Shaping (KAERS) to accelerate learning via sparse but meaningful feedback, and (WEI *et al.*, 2023a) explored curriculum learning to improve convergence in dual-UAV settings. Large-scale air combat scenarios were also addressed in (PIAO *et al.*, 2023), where Graph Neural Networks (GNNs) combined with expert knowledge to reason over abstract combat relationships.

Lastly, comprehensive overviews of DRL applications in air combat simulation environments and military contexts are provided in (DANTAS *et al.*, 2023; SCUKINS *et al.*, 2024a), while (WANG *et al.*, 2023b) reviews broader military applications, outlining key limitations and future directions for DRL-based approaches.

2.4.10 Supervised and Unsupervised Learning

Statistical modeling techniques, particularly those based on supervised and unsupervised learning, have been widely applied to support threat assessment, maneuver intention recognition, missile launch prediction, and combat effectiveness evaluation. These approaches rely on data from simulations or real-world exercises to model combat scenarios, extract patterns, and enhance tactical reasoning under uncertainty.

Most existing work adopts SL, where models are trained on labeled data. For example, (CAO *et al.*, 2021) used Linear Discriminant Analysis (LDA) to preprocess threat indicators and trained an ELM for target threat assessment. For engagement decision support, (DANTAS *et al.*, 2021a) and (DANTAS *et al.*, 2022b) built tree-based models (e.g., XGBoost) to predict engagement outcomes and missile launch timing, while (DANTAS *et al.*, 2021b) applied a DNN to estimate the WEZ from multiple simulated launches. (LIMA-FILHO *et al.*, 2021), in turn, used a missile launch dataset—originated from training exercises—to train an ANN for UCAVs decision support. For air combat effectiveness evaluation, (HUANG *et al.*, 2020) proposed a method based on Relevance Vector Machine (RVM), while (WANG *et al.*, 2023a) introduced an improved ELM with M-estimation to handle gross errors in training data.

Sequence models are often used to address temporal pattern recognition. The study in (CHEN *et al.*, 2022) combined one-dimensional convolutional neural network (1DCNN) and bidirectional Long Short-Term Memory (LSTM) with evidence fusion for intent classification, while (LEI *et al.*, 2023) introduced an attention-enhanced bidirectional GRU architecture tuned via swarm optimization. In (XIA *et al.*, 2023), a GRU was used for enemy state prediction and a decision tree was then applied for intent recognition. In contrast, (YANG *et al.*, 2022) employed a cascaded Support Vector Machines (SVM) framework with hierarchical feature decomposition, which, while not a sequence model per se, operates on temporal trajectory data.

SL techniques have also supported situational awareness and post-launch assessment. For example, (SCUKINS *et al.*, 2024b) and (SCUKINS *et al.*, 2024) trained deep networks to evaluate options under multiple missile threats and assess formation flexibility. In (JIA *et al.*, 2023), random forest regression was used for dynamic post-launch missile effectiveness evaluation. In cooperative contexts, (ZHANG *et al.*, 2018) employed SL to adjust parameters in path planning for multi-agent engagements, and (YAO, 2021) developed a back propagation (BP) neural network to support collaborative target assignment.

In settings with limited or incomplete data, unsupervised learning techniques are often used to extract structure or reduce dimensionality. For instance, (GAO *et al.*, 2022) employed a Sparse Autoencoder (SAE) to approximate the Tactical Control Range (TCR), enabling fast inference without relying on labeled output data.

Other approaches use symbolic or hybrid reasoning frameworks that may not strictly fall under unsupervised learning but operate in low-label or uncertain environments. In (LIU, 2022), Multi-Granulation Rough Set (MGRS) theory was used for adversarial intent recognition, combining logic-based modeling with attribute importance ranking to support classification under uncertainty. Likewise, (MENG *et al.*, 2014) applied granular computing to structure tactical decisions at multiple abstraction levels, proposing a feature reduction method to improve classification accuracy without relying heavily on labeled datasets.

2.5 Existing Simulation Tools

Simulation environments and tools are essential for advancing BVR air combat research, enabling the modeling of complex scenarios, evaluation of decision-making algorithms, and optimization of operational strategies. These tools range from general-purpose platforms to bespoke systems tailored to specific research needs, each offering unique capabilities to address various aspects of BVR combat.

Many platforms support interoperability through standards like HLA (High-Level Architecture) and DIS (Distributed Interactive Simulation), facilitating integration across multiple simulation systems and real-time synchronization. In this section, we describe some of the more common tools that are commonly used in BVR air combat research. At the end of the section, we present an overview table summarizing the key tools, their features, programming languages, and interoperability capabilities.

2.5.1 AFSIM: Advanced Framework for Simulation, Integration, and Modeling

The Advanced Framework for Simulation, Integration, and Modeling (AFSIM) (CLIVE *et al.*, 2015), developed by the United States Air Force Research Laboratory, is a widely used platform in BVR air combat research. AFSIM offers flexibility for modeling combat environments, integrating systems, and supporting mission planning and decision-making processes. It is commonly applied in research on cognitive control, behavior recognition, and AI (AHA, 2018; ALBARADO *et al.*, 2022; ALFORD *et al.*, 2015; BORCK *et al.*, 2014; BORCK *et al.*, 2015a; BORCK *et al.*, 2015b; CHAO *et al.*, 2023; FLOYD *et al.*, 2017a; KARNEEB *et al.*, 2016). AFSIM supports integration with other models, enabling real-time interactions and simulations at both strategic and tactical levels. This interoperability facilitates research in battle management and mission planning. AFSIM is not open-source and is controlled under United States government regulations.

2.5.2 ASA: Aerospace Simulation Environment

The Aerospace Simulation Environment (ASA, from the Portuguese *Ambiente de Simulação Aeroespacial*) (DANTAS *et al.*, 2022a; DANTAS *et al.*, 2023a), developed by the Brazilian Air Force, is a custom-built object-oriented simulation framework in C++. ASA is designed for modeling complex aerospace operations and supports research in SA, mission planning, and operational decision-making (DANTAS *et al.*, 2021a; DANTAS *et al.*, 2021b; DANTAS *et al.*, 2022b; DANTAS *et al.*, 2023; DANTAS *et al.*, 2024; KUROSWISKI *et al.*, 2023; LIMA-FILHO *et al.*, 2022). ASA's flexibility allows integrating ML techniques with traditional simulations, enabling researchers to optimize tactics and predict adversarial behaviors. Its architecture also supports detailed modeling of mission parameters, aircraft systems, and weapons. ASA is not publicly available and is controlled under Brazilian government regulations.

2.5.3 Bespoke Systems

Bespoke systems, which are developed in Python, C++, or MATLAB, are specially designed tools for study where commercially available alternatives are inadequate. Since EW models, missile guidance, and BVR techniques are frequently classified, sensitive data from commercial systems cannot be accessed for open study. Therefore, such tools are often inadequate for the complexity, security, and adaptability requirements of these scenarios. Bespoke systems are the most prevalent as these methodologies promote quick development (LIU *et al.*, 2022b; MEI; WANG, 2021; WAN *et al.*, 2019a; YANG *et al.*, 2020; ZHOU *et al.*, 2014; HA *et al.*, 2015; HA *et al.*, 2018; KANG *et al.*, 2019; KUNG; CHIANG, 2015; MA *et al.*, 2020; SAMSONOVICH; AHA, 2015; SUN *et al.*, 2020b; ZHANG *et al.*, 2023; MANSIKKA *et al.*, 2021b; MANSIKKA *et al.*, 2021c; LIU *et al.*, 2020; HAOYU *et al.*, 2018; LEI *et al.*, 2015; LUO *et al.*, 2014; WANG *et al.*, 2023; YUAN *et al.*, 2016; ZHANG *et al.*, 2021; ZOUJIE; WUWEI, 2016; DING *et al.*, 2018; LEI *et al.*, 2022; LI; LIU, 2015; LI *et al.*, 2022c; LI *et al.*, 2022a; RU *et al.*, 2024; YANG *et al.*, 2014; YIN *et al.*, 2022; YOU *et al.*, 2019; GONG *et al.*, 2023; HU *et al.*, 2021; HU *et al.*, 2022; JIANG *et al.*, 2022; LIU *et al.*, 2022a; MAO *et al.*, 2022; WANG *et al.*, 2021; WEILIN *et al.*, 2018; ZHANG *et al.*, 2022; CAO *et al.*, 2021; CHEN *et al.*, 2022; HUANG *et al.*, 2020; JIA *et al.*, 2023; LIMA-FILHO *et al.*, 2021; LIU, 2022; MENG *et al.*, 2014; WANG *et al.*, 2023a; YAO, 2021).

2.5.4 DCS World: Digital Combat Simulator World

DCS World (EAGLE DYNAMICS, 2024) is a high-fidelity, commercially available combat flight simulator. Known for its realistic flight physics and detailed models, it is widely used in studies on decision-making and RL-based combat engagement (HE *et al.*, 2023; QIU *et al.*,

2020). Its open architecture supports custom module development, enabling researchers to simulate dynamic, high-stakes BVR combat scenarios. This capability makes it an ideal platform for testing AI-driven agents under realistic operational conditions.

2.5.5 FLAMES: Flexible Analysis and Modeling Effectiveness System

FLAMES (TERNION, 2023) is a modular, commercial framework for developing and executing LVC simulations. It supports real-time visualization, scenario management, and OA, making it effective for mission planning and combat simulations (DANTAS *et al.*, 2022b). Despite its adaptability, FLAMES' commercial licensing can limit accessibility, and its complexity can hinder rapid prototyping or use in resource-constrained research contexts.

2.5.6 FLSC: Swedish Air Force Combat Simulation Centre

The Swedish Air Force Combat Simulation Centre (FLSC), developed by the Swedish Defense Research Agency, incorporates LVC simulations to analyze air combat scenarios. FLSC is utilized for pilot training, mission planning, and decision-support research, as well as evaluating Human-AI Collaboration (HAIC) (ARONSSON *et al.*, 2019; ARONSSON *et al.*, 2023). Its features contribute to enhancing SA and decision-making in joint operations. Since FLSC is operated by FOI (Swedish Defence Research Agency), access is restricted and is not publicly available, but researchers working on defense projects may gain access through FOI partnerships.

2.5.7 JSBSim

JSBSim (JSBSIM DEVELOPMENT TEAM, 2024) is an open-source flight dynamics model widely used in RL-based BVR studies requiring precise aircraft simulations. It supports tasks such as decision-making, maneuver optimization, and combat engagement (HE *et al.*, 2024; SCUKINS *et al.*, 2024; SCUKINS *et al.*, 2024b; XIA *et al.*, 2024; SCUKINS *et al.*, 2023; SCUKINS *et al.*, 2024a). JSBSim is often integrated with platforms like Unity (IAGSim) and bespoke environments to create computationally efficient simulations for exploring autonomous decision-making in dynamic scenarios.

2.5.8 MATLAB and Simulink

MATLAB (MATHWORKS, 2024a) and Simulink (MATHWORKS, 2024b) are widely used for simulation, control theory, and optimization research. MATLAB's mathematical ca-

pabilities support studies on decision-making and combat engagements (ALKAHER; MOSHAIOV, 2016; LEI *et al.*, 2023; ÖSTROM *et al.*, 2023; PENGYUAN *et al.*, 2014; SUSENO; SASONGKO, 2016; XIE *et al.*, 2023; YANG *et al.*, 2020; YAO *et al.*, 2015a; YAO *et al.*, 2015b; ZHANG *et al.*, 2018; YANG *et al.*, 2022; YE *et al.*, 2023; GAO *et al.*, 2022; JIA; KIANG, 2022; LU *et al.*, 2021; LI *et al.*, 2022b). Simulink extends MATLAB's functionality with graphical tools for dynamic system modeling, offering a useful platform for control strategies.

2.5.9 Python and R

Python is a key tool for developing simulation environments and ML models. With libraries such as TensorFlow (ABADI *et al.*, 2016) and PyTorch (PASZKE *et al.*, 2024), Python enables mission planning, RL implementation, and optimization (BAKIRCI; OZER, 2023; HU *et al.*, 2022; SCUKINS *et al.*, 2023; XIA *et al.*, 2023). Its flexibility supports rapid prototyping and integration with other platforms for air combat research. R is occasionally used in air combat research for data analysis and simulation-related statistical modeling (DANTAS *et al.*, 2021b). Although Python and R are programming languages rather than simulation tools themselves, they are widely used in many applications in this field. Moreover, since the specific application or simulation framework is not always identified in the literature, these languages were classified as such to ensure completeness and consistency in the review.

2.5.10 Other Tools

Several other tools have been developed or adapted to support BVR air combat research, each contributing in different ways depending on the goals of the study:

- **ACE-2**: A custom simulator used for testing genetic optimization techniques in air combat maneuvering (MASEK *et al.*, 2021).
- **ACEM**: A LVC simulation environment for human performance analysis in air combat (MANSIKKA *et al.*, 2021a).
- **FTD (F/A-18C)**: Flight Training Device for the F/A-18C, used for high-fidelity simulation of pilot behavior, coordination, and training scenarios (MANSIKKA *et al.*, 2020; MANSIKKA *et al.*, 2021d; VIRTANEN *et al.*, 2022).
- **IAGSim (Unity + JSBSim)**: A custom-built simulator that combines JSBSim for flight dynamics and Unity for real-time rendering, designed for autonomous air combat research (QIAN *et al.*, 2023).

- **MACE** (BATTLESPACE SIMULATIONS, 2024): The Modern Air Combat Environment (MACE) is a scalable distributed simulation used for operational analysis and testing of tactical air combat scenarios (DIETL *et al.*, 2023).
- **NLR's Fighter 4-Ship Simulator**: A simulator developed by the Netherlands Aerospace Centre (NLR) for pilot training and human-autonomy interaction in multi-aircraft engagements (TOUBMAN, 2019).
- **STAGE**: A framework for rapidly generating air combat scenarios used in AI and RL training (TOUBMAN *et al.*, 2016).
- **Super Decisions**: A decision-support software implementing the AHP and Analytic Network Process (ANP), used in air combat for threat ranking and mission planning (SHI *et al.*, 2021).
- **UnBBayes-MEBN**: A probabilistic reasoning framework based on MEBN, applied to situation awareness and decision-making under uncertainty (SHI *et al.*, 2020).
- **WESS**: A simulation tool for studying adaptive tactical decision-making. It has been applied in modeling dynamic combat behavior (YAO *et al.*, 2015a; YAO *et al.*, 2015b).
- **Wukong**: A RL-based platform designed for multi-agent tactical decision-making in BVR scenarios (PIAO *et al.*, 2020; PIAO *et al.*, 2023; SUN *et al.*, 2021).
- **X-Plane** (LAMINAR RESEARCH, 2024): A high-fidelity commercial flight simulator used in autonomous behavior validation and operational planning (UMMAH *et al.*, 2019).

2.5.11 Tools Summary

Table 2.1 summarizes the key tools, their primary applications, features, programming languages, and interoperability capabilities. This table includes 116 of the 120 works mapped in this work; the remaining four were survey or overview works that did not employ a specific tool. Each column provides specific information to facilitate comparison among simulation environments: **Simulation Tool** lists the name of the simulator or framework; **Key Features** highlights the main characteristics or functionalities relevant to BVR air combat research; **Programming Language** indicates the primary languages or platforms used for development or customization; **Interoperability** specifies whether the tool supports standard simulation protocols (e.g., HLA, DIS), uses custom interfaces, or lacks interoperability information; and **References Using This Simulation Tool** lists the studies that employed each tool in their experiments or analyses.

TABLE 2.1 – Overview of simulation tools in BVR air combat research.

Simulation Tool	Key Features	Programming Language	Interoperability	References Using This Simulation Tool
ACE-2	Focus on genetic-algorithm-based BVR combat scenarios	Not specified	Custom	(MASEK *et al.*, 2021)
ACEM	Focus on integrated LVC training	Not specified	Custom	(MANSIKKA *et al.*, 2021a)
AFSIM	Supports AI integration, scenario-based analysis, and cognitive models	C++	HLA, DIS	(AHA, 2018; ALFORD *et al.*, 2015; BORCK *et al.*, 2014; BORCK *et al.*, 2015a; BORCK *et al.*, 2015b; FLOYD *et al.*, 2017a; KARNEEB *et al.*, 2016; ALBARADO *et al.*, 2022; CHAO *et al.*, 2023)
ASA	Integrates ML, modular architecture, and object-oriented design	C++	HLA, DIS	(DANTAS *et al.*, 2022a; DANTAS *et al.*, 2023a; DANTAS *et al.*, 2024; KUROSWISKI *et al.*, 2023; LIMA-FILHO *et al.*, 2022; DANTAS *et al.*, 2023; DANTAS *et al.*, 2021a)
Bespoke Systems	Tailored for specific research needs with high adaptability and customizability	Python, C++, MATLAB	Custom	(LIU *et al.*, 2022b; MEI; WANG, 2021; WAN *et al.*, 2019a; YANG *et al.*, 2020; ZHOU *et al.*, 2014; HA *et al.*, 2015; HA *et al.*, 2018; KANG *et al.*, 2019; KUNG; CHIANG, 2015; MA *et al.*, 2020; SAMSONOVICH; AHA, 2015; SUN *et al.*, 2020b; ZHANG *et al.*, 2023; MANSIKKA *et al.*, 2021b; MANSIKKA *et al.*, 2021c; LIU *et al.*, 2020; HAOYU *et al.*, 2018; LEI *et al.*, 2015; LUO *et al.*, 2014; WANG *et al.*, 2023; YUAN *et al.*, 2016; ZHANG *et al.*, 2021; ZOUJIE; WUWEI, 2016; DING *et al.*, 2018; LEI *et al.*, 2022; LI; LIU, 2015; LI *et al.*, 2022c; LI *et al.*, 2022a; RU *et al.*, 2024; YANG *et al.*, 2014; YIN *et al.*, 2022; YOU *et al.*, 2019; GONG *et al.*, 2023; HU *et al.*, 2021; HU *et al.*, 2022; JIANG *et al.*, 2022; LIU *et al.*, 2022a; MAO *et al.*, 2022; WANG *et al.*, 2021; WEI *et al.*, 2023a; WEILIN *et al.*, 2018; ZHANG *et al.*, 2022; CAO *et al.*, 2021; CHEN *et al.*, 2022; HUANG *et al.*, 2020; JIA *et al.*, 2023; LIMA-FILHO *et al.*, 2021; LIU, 2022; MENG *et al.*, 2014; WANG *et al.*, 2023a; YAO, 2021)
DCS World	Realistic flight physics, open architecture for third-party integration	Lua, C++	Custom	(QIU *et al.*, 2020; HE *et al.*, 2023)
F/A-18C FTD	High-fidelity simulation for realistic F/A-18C training scenarios	Not specified	Not specified	(MANSIKKA *et al.*, 2020; MANSIKKA *et al.*, 2021d; VIRTANEN *et al.*, 2022)
FLAMES	Modular platform with LVC support and scalable architecture	C++	HLA, DIS	(DANTAS *et al.*, 2022b)
FLSC	Advanced LVC simulation, designed for real-time collaboration scenarios	Not specified	HLA, DIS	(ARONSSON *et al.*, 2019; ARONSSON *et al.*, 2023)
IAGSim	Custom-built environment combining flight dynamics and real-time rendering	C++, Unity	Custom	(QIAN *et al.*, 2023)
JSBSim	Open-source physics-based flight dynamics model	C++	Custom	(HE *et al.*, 2024; SCUKINS *et al.*, 2023; SCUKINS *et al.*, 2024a; XIA *et al.*, 2024; SCUKINS *et al.*, 2024; SCUKINS *et al.*, 2024b)
MACE	LVC simulation for large-scale air combat scenarios	C++	HLA	(DIETL *et al.*, 2023)
MATLAB and Simulink	Tools for mathematical modeling, control systems, and simulation of algorithms	MATLAB	Custom	(XIE *et al.*, 2023; YANG *et al.*, 2020; ZHAO *et al.*, 2020; ALKAHER; MOSHAIOV, 2016; LI *et al.*, 2022b; LU *et al.*, 2021; SUSENO; SASONGKO, 2016; JIA; KIANG, 2022; REINISCH *et al.*, 2022; PENGYUAN *et al.*, 2014; GAO *et al.*, 2022; ÖSTROM *et al.*, 2023; YE *et al.*, 2023; ZHANG *et al.*, 2018; YANG *et al.*, 2022; LEI *et al.*, 2023)
NLR's Fighter 4-Ship Simulator	Focus on collaborative training with realistic physics and AI integration	C++, Custom	Custom	(TOUBMAN, 2019)
Python	Script-based environment for fast experimentation	Python	Custom	(BAKIRCI; OZER, 2023; SCUKINS *et al.*, 2023; HU *et al.*, 2022; XIA *et al.*, 2023)
R	Statistical environment for rapid prototyping	R	Custom	(DANTAS *et al.*, 2021b)
STAGE	Rapid scenario generation for training and real-time experimentation	Not specified	Custom	(TOUBMAN *et al.*, 2016)
Super Decisions	Implements AHP/ANP-based models for threat assessment	Not specified	Not specified	(SHI *et al.*, 2021)
UnBBayes-MEBN	Implements MEBN-based reasoning for real-time updates	Java	Custom	(SHI *et al.*, 2020)
WESS	Simulation environment structured in sequential phases to model adaptive air combat tactics	Not specified	Not specified	(YAO *et al.*, 2015a; YAO *et al.*, 2015b)
Wukong	RL-driven platform for learning complex combat strategies	Python	Custom	(PIAO *et al.*, 2020; PIAO *et al.*, 2023; SUN *et al.*, 2021)
X-Plane	High-fidelity flight simulation and real-world scenario testing	C++	Custom	(UMMAH *et al.*, 2019)

2.6 Open Challenges

Despite significant advancements in air combat decision-making using modern techniques, several challenges remain, presenting opportunities for future research.

Complexity of Scenarios: Current methods, such as NFSP, RL, and DRL with DQR, are often validated in simplified one-on-one engagements (HE *et al.*, 2023; HU *et al.*, 2022). Extending these approaches to multi-agent environments that reflect the complexity of real-world air combat is crucial. Promising frameworks, including swarm-based strategies leveraging DDPG and hierarchical methods like H3E, highlight potential directions for tackling this challenge (WANG *et al.*, 2021; QIAN *et al.*, 2023). Additionally, TA, detection, and guidance problems predominantly assume homogeneous models of radars, aircraft, and communication nodes (LUO *et al.*, 2014; XU *et al.*, 2017; XIAO *et al.*, 2019; ZHAO *et al.*, 2020; ZHANG *et al.*, 2021; LEI *et al.*, 2022; LI *et al.*, 2022a). Future research can explore heterogeneous models to better capture the complexities of diverse real-world systems.

Full Observability Assumptions: Many methods, such as those based on MCTS, PPO, and CSVM, assume full observability of the environment, omitting critical aspects like radar target searching (ZHANG *et al.*, 2022; ZHANG *et al.*, 2022; YANG *et al.*, 2022). Techniques capable of handling partial observability, such as KAERS in BVR scenarios, offer promising solutions for enhancing model robustness and real-world applicability (PIAO *et al.*, 2020).

Computational Intensity: Approaches like MCTS, while effective, are computationally expensive and time-consuming (ZHANG *et al.*, 2022). Optimizing methods for continuous action spaces and improving computational efficiency is essential for real-time applications. Recent efforts, such as enhancing TD3 algorithms for missile engagement and evasion, demonstrate progress in this area (QIU *et al.*, 2020).

Sensitivity to Initial Conditions: Techniques using curriculum learning and IQN often perform poorly under unfavorable initial configurations (WEI *et al.*, 2023b; WEILIN *et al.*, 2018). Robust curriculum designs and adaptive learning rates, as seen in evolving BTs with GP, offer potential strategies for mitigating sensitivity and improving generalization (MASEK *et al.*, 2021).

Scalability and Real-Time Adaptability: Scalability remains a challenge for multi-agent approaches and hierarchical frameworks, such as MAPPO and H3E, particularly in dynamic and large-scale environments (LIU *et al.*, 2022a; QIAN *et al.*, 2023). Efficient methods are needed to handle cooperative scenarios, as demonstrated in TA research (LI *et al.*, 2022b; MA *et al.*, 2020).

Incorporation of Uncertainties: Many existing methods, such as those based on game theory, BN, and SL, assume deterministic environments (SUN *et al.*, 2020b; MA *et al.*, 2020; YANG *et al.*, 2022). Incorporating stochastic elements and uncertainties into these models will improve their realism and applicability to complex air combat scenarios.

Validation in Diverse Scenarios: Techniques like SAE networks for TCR and DRL-based UAV swarm models have largely been tested in static environments (GAO *et al.*, 2022;

WANG *et al.*, 2021). Expanding validation to dynamic and high-dimensional scenarios, including real-time decision-making and varied combat conditions, is essential. Studies employing ANN and granular computing approaches for cooperative air combat highlight promising steps in this direction (YAO, 2021; MENG *et al.*, 2014).

Interdisciplinary Approaches: Combining RL, Deep Learning (DL), and control theory can significantly enhance decision-making models for BVR combat. Integrating these methods allows for adaptive strategies while adhering to physical constraints. For instance, hierarchical RL and BTs provide scalable frameworks for managing high-level tactics and low-level maneuvers (UMMAH *et al.*, 2019; MENG *et al.*, 2014). Such interdisciplinary methods will foster more robust and interpretable models.

Enhancing Training Efficiency: GP shows promise for optimizing strategies, but challenges remain in handling low-dimensional problems and reducing computational overhead. Curriculum-based RL and techniques for recognizing enemy intent can significantly improve learning efficiency and decision-making capabilities (HE *et al.*, 2023).

Real-World Applicability: Ensuring the real-world viability of advanced methods requires extensive validation in high-fidelity simulations. Collaboration with military and aerospace organizations can bridge the gap between research and operational deployment. Existing tools for swarm strategies and cooperative UCAVs operations demonstrate the value of simulation for practical testing (WANG *et al.*, 2021; LIU *et al.*, 2022a).

Future Trends in Simulation Tools: Simulation tools must evolve to meet research demands as BVR combat scenarios grow in complexity. Key trends include:

- *Higher-Fidelity Multi-Agent Simulations:* Supporting larger-scale swarm coordination and high-fidelity real-time simulations on platforms like AFSIM, ASA, DCS WORLD, and FLSC.
- *Increased Interoperability:* Using standards like HLA and DIS to integrate simulations across heterogeneous systems (e.g., manned aircraft, drones, and missiles).
- *AI and ML Integration:* Embedding adaptive AI agents for real-time mission planning and decision-making (LIU *et al.*, 2022a).
- *Higher Computational Efficiency:* Optimizing simulations to handle growing complexity while enabling real-time adaptability.

By addressing these challenges, future research can develop sophisticated, scalable, and adaptable BVR decision-making models. Addressing these challenges will pave the way for robust autonomous systems capable of adapting and thriving in highly dynamic and contested air combat environments.

2.7 Outcomes

This survey reviewed over 120 research papers on the application of simulation and ML techniques in BVR air combat. We categorized the literature based on key methodologies and application areas, emphasizing advancements in threat assessment, engagement strategies, and autonomous UAV control. Additionally, we underscored the pivotal role of simulation environments in modeling complex BVR scenarios, validating strategic approaches, and developing effective training programs, decision-making algorithms, and autonomous systems.

While significant progress has been made, our review identified several critical challenges that persist. These include ensuring scalability, achieving real-time adaptability, and managing multi-agent coordination in dynamic combat environments. Addressing computational efficiency, simulation fidelity, and the incorporation of uncertainty remains essential for the practical deployment of advanced techniques.

Future research should focus on integrating traditional simulation frameworks with AI-driven methods to overcome these challenges. Combining DL, RL, and control-theoretic approaches holds promise for creating sophisticated, scalable, and interpretable BVR combat systems.

Ultimately, this survey highlights the necessity of advanced tools and ML techniques in shaping the future of BVR air combat. By enhancing both human decision-making and autonomous capabilities, these innovations will enable air forces to operate effectively in increasingly contested and complex operational environments.

Part II

Simulation Tools and Services

3 Simulation Environment

This chapter explores the details of the simulation environment used in this research. This environment plays an important role in evaluating military operational scenarios, providing a robust platform for analysis and testing. As indicated in Figure 1.2, this chapter primarily contributes to the "Simulation Tools and Services" area within the proposed research framework.

The development and insights presented here are based on the following work:

> **DANTAS, J. P. A.**; COSTA, A. N.; GOMES, V. C. F.; KUROSWISKI, A. R.; MEDEIROS, F. L. L.; GERALDO, D. ASA: A Simulation Environment for Evaluating Military Operational Scenarios. In: THE 2022 WORLD CONGRESS IN COMPUTER SCIENCE, COMPUTER ENGINEERING & APPLIED COMPUTING (CSCE'22). The 20th International Conference on Scientific Computing (CSC'22). Proceedings [...]. Las Vegas, NV, USA, 2022.

3.1 Summary

The Aerospace Simulation Environment (*Ambiente de Simulação Aeroespacial – ASA* in Portuguese) is a custom-made object-oriented simulation framework developed mainly in C++ that enables the modeling and simulation of military operational scenarios to support the development of tactics and procedures in the aerospace context for the Brazilian Air Force. This chapter describes the ASA framework and its main features: a distributed architecture for coordinating multiple simulation machines, a modular structure that allows models to be loaded at runtime, a batch execution mode for simulating multiple scenarios with varied initial conditions, and an integrated data analysis platform for post-processing simulation results. In addition, we present a list of recent studies that have used ASA in applications related to decision support and autonomy in air combat scenarios.

3.2 Introduction

The Institute for Advanced Studies (IEAv), a research organization of the Brazilian Air Force (*Força Aérea Brasileira* — FAB, in Portuguese), has developed, since 2018, the Aerospace Simulation Environment (*Ambiente de Simulação Aeroespacial* — ASA, in Portuguese) to provide a computational solution that enables the modeling and simulation of operational scenarios. This solution allows users to define strategies, parameters, and command decisions to support the development of tactics, techniques, and procedures in the aerospace domain for defense purposes.

The characteristics of modern battlefield scenarios present significant challenges to the development of practical combat simulations (COSTA *et al.*, 2025). These challenges call for more integrated and flexible solutions that can address both technical and organizational aspects (HILL *et al.*, 2018).

To meet some of these demands, frameworks such as the Advanced Framework for Simulation, Integration, and Modeling (AFSIM) (CLIVE *et al.*, 2015), Wukong (HAN *et al.*, 2022), FLAMES (TERNION, 2023), and VR-Forces (MAK TECHNOLOGIES, 2023) have been developed. However, many of these tools are commercial products or restricted to use within specific countries. In this context, the ASA environment was conceived as a national solution designed to support FAB's strategic planning, meet operational analysis needs, and promote the development and evaluation of emerging technologies for military research.

ASA is designed as a flexible and modular platform, capable of adapting to diverse user needs. This flexibility is essential given the wide range of requirements from its users. These needs could not be fully addressed by commercial off-the-shelf (COTS) simulation software. Rather than building an entirely new system, the development approach integrated openly available tools into a unified simulation environment that is flexible, accessible, and scalable.

The proposed solution uses the Mixed Reality Simulation Platform (MIXR) (HODSON; GEHL, 2018) as its simulation engine, an open-source software project designed to support the development of robust, scalable, virtual, constructive, stand-alone, and distributed simulation applications. ASA extends MIXR's capabilities by adding components that streamline tasks for both developers and analysts. A manager application was created to serve as an interface between multiple resources, working as a hub to execute, store, and analyze simulations across multiple machines. This application also supports the simultaneous creation of numerous simulations by varying initial conditions according to the analyst's needs. Furthermore, models and tools can be dynamically loaded at runtime to increase flexibility. All simulation data are stored in a dedicated database, expediting

data collection and enabling more robust statistical analysis. Additionally, given the complexity of simulation outcomes and the varied technical background of ASA users, a dedicated data analysis platform was integrated into the system, not only for planning and visualization but also for post-processing the scenario data.

The main contribution of this chapter is the introduction of a new environment for modeling and simulation in the aerospace domain for military applications. It features: (i) a distributed architecture for managing multiple simulation machines; (ii) support for runtime model loading within a modular architecture, allowing new models to be easily integrated; (iii) a batch execution mode that enables the simulation of multiple scenarios with varied initial conditions; and (iv) an enhanced data analysis platform for post-processing military operational scenario data. Additionally, we present a list of recent studies that have used the ASA platform in applications related to decision support and autonomy in military operational scenarios.

The remainder of this paper is organized as follows. Section 3.3 presents the ASA architecture. Section 3.4 discusses studies that have used ASA in air combat analysis as application examples of the simulation framework. Finally, Section 3.5 provides concluding remarks on the current state of ASA and outlines potential directions for future work.

3.3 ASA Architecture

The ASA design consists of three main modules. The first part is the simulation framework, defined as AsaSimulation, which provides the applications and necessary services to create and execute simulations. The second part comprises the interface applications, denominated AsaInterfaces, which provide tools for creating scenarios by listing all available components to be included and a library for interacting with the AsaSimulation module. Lastly, the third part is the analysis module, called AsaDataScience, which allows for post-processing and analysis of scenario executions.

Figure 3.1 displays a summary of the ASA architecture, and the following subsections provide details on all three primary ASA modules. All ASA applications use network communications, allowing processing to be distributed across multiple servers on a network.

3.3.1 AsaSimulation

The AsaSimulation module provides the necessary components for developing and executing a scenario simulation. It consists of applications, services, and libraries that

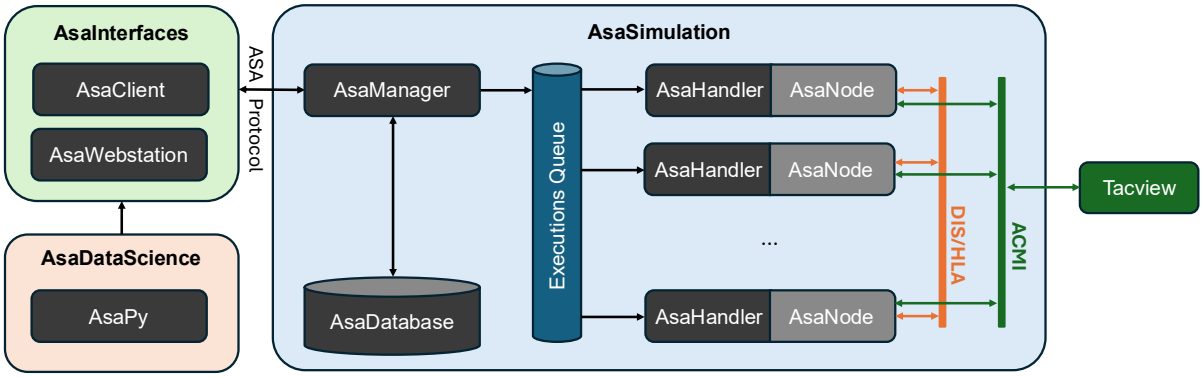


FIGURE 3.1 – ASA: modules and software applications.

help create agent models by developers and in the elaboration and simulation of scenarios by analysts.

The main features of this module are management and execution of simulations, dynamic loading of extension models, distribution of simulation executions, and management of simulation batch processing.

A permanent storage service called AsaDatabase keeps agent metadata, simulation scenarios, execution data, and analysis results. This service meets the storage demands of the AsaSimulation and AsaDataScience modules.

New agent models can be added to ASA by extending the interfaces and classes available in a library called AsaExtension. A new agent model that will be loaded into the AsaSimulation framework must provide its compiled source code (shared object file) and a JSON file that describes the parameters and components accepted by the model. AsaExtension also provides functionalities that allow extensions to store agent data in the AsaDatabase. This is done through the C++ macro `RECORD_ASA_CUSTOM_DATA("tag", agent)`, which receives as its first parameter a tag that identifies the agent's type and the agent object with the attributes to be stored in the AsaDatabase. Storing the state of the agents at each simulation step is optional, but it is essential if there is an interest in performing post-processing of the simulation data.

The AsaSimulation framework enables analysts to specify agent parameters (placeholders) during the simulation execution request, facilitating the execution of batch simulations. Analysts can request a batch execution from these simulation scenario templates by providing a list of initial conditions for the previously selected agent parameters. Each set of initial conditions, combined with the scenario template, generates an execution request that is allocated to run on a distributed processing node.

The distribution of simulation processing is done by dividing the responsibilities of simulating to three applications on a network: AsaManager, AsaHandler, and AsaNode.

The following subsections detail each of these applications.

3.3.1.1 AsaManager

is responsible for coordinating the processes to use ASA in a distributed manner. One of the essential functions of AsaManager is receiving and preprocessing execution requests, and automatically and transparently dispatching the requested simulation to be executed on an available node. The distribution of executions is performed using a queue service based on the AMQP protocol. AsaManager places a validated simulation execution request in the queue (Executions Queue) and, when available, an AsaHandler starts serving the request.

The use of queues to decouple direct management between request handling and execution enables the dynamic addition of new processing nodes, such as AsaHandler/AsaNode, to the system in response to increased demand for parallel execution. Because communication between AsaManager, ExecutionsQueue, and AsaHandler occurs over a network, applications can be distributed across different machines within a networked environment.

AsaManager incorporates a Representational State Transfer API (REST API) to enable interaction between other modules and the simulation functionalities, as well as access to stored data. This API implements the ASA protocol, which exposes AsaSimulation as a simulation service to other applications. It enables the submission, validation, execution, pausing, resumption, cancellation, and monitoring of single- or batch-simulation tasks. Additionally, this protocol defines methods for accessing data from completed simulations and managing user data.

Access is restricted to authenticated and authorized users who are permitted to use the system's functionalities. For each simulation execution request, AsaManager verifies the validity of the scenario to be simulated and checks whether the authenticated user is authorized to use the listed components and has sufficient credit available. Within the ASA framework, these credits are referred to as AsaCoins, which serve as a means to account for user resource consumption. During each simulation execution, both the CPU time consumed and the disk space used to store the simulated agent data were measured. These two variables are then converted into an equivalent value in AsaCoins, which is subsequently debited from the user's account. The reference rate for a single AsaCoin unit was established based on the execution of a standard benchmark scenario.

3.3.1.2 AsaHandler

is responsible for monitoring the execution queue, converting requests, and supervising simulation execution. When an AsaHandler is available and a new execution request

is placed in the queue by AsaManager, the AsaHandler retrieves the request, converts it into a format compatible with the AsaNode, initiates a new AsaNode process, and monitors its execution. Communication between AsaHandler and AsaNode is carried out via interprocess communication, enabling AsaHandler to oversee the simulation run. If the AsaNode terminates unexpectedly or a failure is detected by the AsaHandler, it will terminate the execution and will keep the AsaManager informed of the simulation status.

3.3.1.3 AsaNode

is the simulation engine of the ASA platform, and its primary function is to process the simulation itself. It is an executable file obtained from the compilation of codes developed in MIXR and features developed by the ASA team, such as dynamically loading extensions and controlling the simulation's execution (pause, resume, stop, execution speed, etc.). It estimates how the scenario will evolve, considering the models incorporated in each agent present in the simulation. AsaNode can run on the same machine as the AsaManager application or in a clustered computing environment. This capability is essential when the user wants to simulate a set of scenarios, called batch, and the main difference between them is the initial configuration of each agent.

At predefined time intervals, AsaNode reports the progress of the simulation to AsaHandler, allowing this application, or AsaManager, to manage the execution process. If enabled by the execution requester, AsaNode can transmit data over the network using either the Distributed Interactive Simulation (DIS) protocol or the High-Level Architecture (HLA) standard, both of which are supported by the MIXR framework. DIS is a protocol designed for real-time exchange of information between simulation entities in a distributed environment, while HLA is a general architecture that enables interoperability among different simulation systems through a shared runtime infrastructure (KUHL *et al.*, 1999). To allow simulation data to be streamed for visualization in the Tacview flight analysis tool*, the AsaExtensions library implements the ACMI file format and its associated Real-time Telemetry Protocol (RAIA SOFTWARE, 2024).

3.3.2 AsaInterfaces

The AsaInterface module includes two tools: WebStation and AsaClient. WebStation provides a visual interface for building simulation scenarios, while AsaClient manages requests to the simulation manager and handles user authentication.

*`https://www.tacview.net`

3.3.2.1 WebStation

provides a graphical user interface (GUI) accessible to registered users through a web page developed with VueJS and Django. Analysts use the platform to construct simulation scenarios that include military symbols, geometric drawings, aeronautical charts, and digital terrain models. The interface displays how each scenario progresses during the simulation, offering an interactive environment for analysis and planning. Users can define performance metrics and examine the results after the simulation is completed. The platform also allows users to specify which messages should be stored in the database during execution. Furthermore, WebStation includes a validation feature that checks whether scenario components meet the model's requirements and ensures that all attributes and subcomponents are properly defined.

To support batch execution, WebStation allows users to configure input parameters that vary across multiple runs. After completing the scenario setup, users can download a JSON file containing the scenario data or save it to the database for future access. The interface also supports uploading JSON files, enabling users to edit and reuse existing scenarios.

An overview of the WebStation interface, including scenario construction and visualization features, is shown in Figure 3.2.

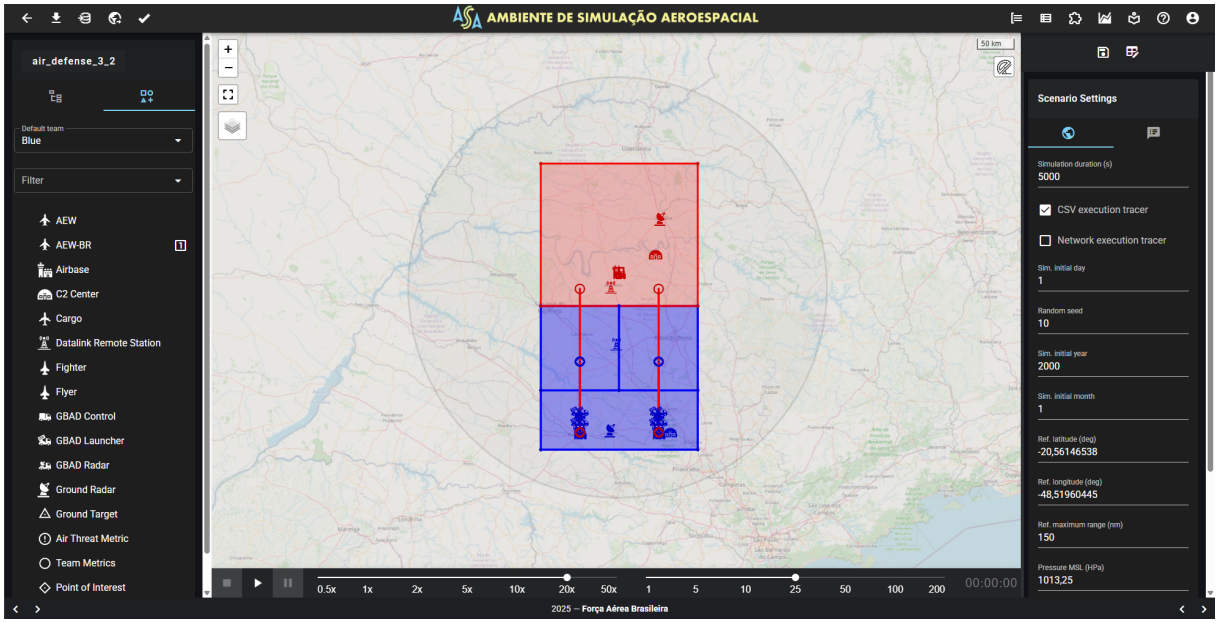


FIGURE 3.2 – WebStation: interface to create and visualize simulations.

3.3.2.2 AsaClient

is a Python library designed to operate with the ASA protocol. It performs two primary functions: managing user authentication and handling communication with the sim-

ulation manager. AsaClient serves as an essential interface between the AsaDataScience environment and the Manager, enabling secure and structured access to simulation resources.

The library provides a set of high-level methods that allow analysts to interact with the core components of the simulation infrastructure. Through AsaClient, users can authenticate in the system, submit execution requests, and monitor simulation progress. AsaClient also supports database access for retrieving results, storing output, and managing simulation metadata.

3.3.3 AsaDataScience

The AsaDataScience module provides tools for understanding the factors that led to the simulation results, supporting analysts in identifying how performance, cost, and operational constraints evolve across military scenarios. It is composed of two main components: (i) the AsaPy Python library, designed to assist in the post-processing of simulation data; and (ii) a cloud-based computational environment with access to high-performance hardware and a JupyterHub interface for large-scale batch execution and interactive data analysis.

At the core of this module is AsaPy, a specialized Python library that provides a structured pipeline for simulation data analysis (DANTAS *et al.*, 2024). It integrates well-established methods for experiment planning, batch execution control, statistical analysis, and machine learning, enabling users to derive actionable insights with minimal programming effort. Its internal structure follows a typical analyst workflow and is composed of four main functional blocks, described in Table 3.1.

TABLE 3.1 – AsaPy functionalities and corresponding descriptions.

Functionality	**Description**
`Design of Experiments`	Generates input configurations to explore the simulation parameter space efficiently.
`Execution Control`	Manages simulation batches with chunking and early stopping based on convergence of key metrics.
`Analysis`	Offers tools for statistical analysis, data exploration, and visualization.
`Prediction`	Applies supervised and unsupervised machine learning models for regression and classification tasks.

Running a simulation invariably results in the collection of massive data at every step. Once these data are adequately stored and structured in the AsaDatabase, analysts can employ techniques such as data visualization, statistical testing, and model fitting

to uncover insights from the simulations. AsaPy streamlines this process by automating routine steps and standardizing the analytical workflow.

A particularly important feature of AsaPy is its support for batch execution. In many military simulations, it is necessary to run hundreds or thousands of scenarios with varying input parameters (GILL *et al.*, 2018). AsaPy allows these executions to be divided into chunks, monitors the results of each chunk, and applies convergence-based early stopping criteria to avoid unnecessary runs. This improves both analytical efficiency and resource usage. Figure 3.3 illustrates this batch execution workflow.

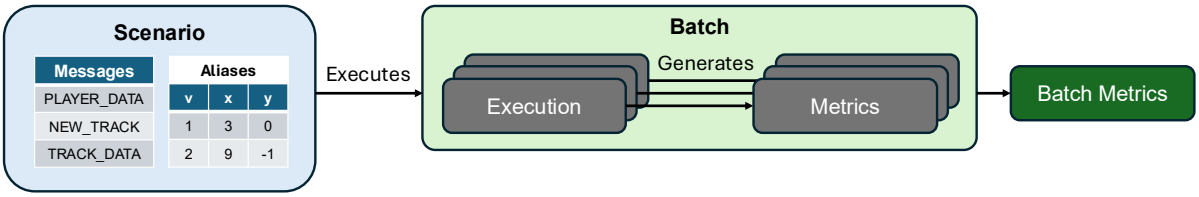


FIGURE 3.3 – Batch execution flow supported by AsaPy. The process begins with the selection of aliases, which define the input parameters to be varied across simulations, and messages, which specify the simulation outputs to be collected. The scenario batch is then divided into sequential chunks. Each chunk is executed, and the selected outputs are used to compute batch-level metrics, which can support monitoring and early stopping of the execution process.

To support large-scale simulations, AsaDataScience includes a cloud-based computational environment equipped with high-performance hardware and a JupyterHub interface[†]. This setup enables analysts to manage simulation studies interactively through notebooks, providing both scripting flexibility and real-time visualization. JupyterHub also facilitates collaborative work by allowing multiple users to run batch executions, monitor progress, and analyze results in a shared and scalable environment, all without requiring direct access to low-level computing resources.

Overall, AsaDataScience serves as a unified environment for analyzing simulation data in defense applications, regardless of the simulation engine employed. In addition to supporting ASA-native data formats, it is also compatible with outputs from other frameworks, as demonstrated in Dantas *et al.* (2022b).

3.4 Applications in Decision Support and Autonomy

ASA has been a flexible and useful platform for research in air combat operations, especially in areas related to decision support and autonomous systems. The studies listed below, in chronological order, demonstrate how ASA has been gradually adopted in increasingly complex and realistic scenarios.

[†]`https://jupyter.org/hub`

The earliest study, Dantas (2018), presented an artificial neural network model to support pilot situational awareness in Beyond Visual Range (BVR) air combat. Using simulation data, the system generated offensive and defensive assessments to aid decision-making during flight. This work was developed using the AEROGRAF Platform (PETERSEN *et al.*, 2008), which preceded ASA and influenced its initial design.

ASA was later used in Kuroswiski (2020) to explore the feasibility of agent-based modeling and simulation for assessing air defense capabilities in BVR engagements. This marked the first application of ASA in a strategic planning context.

In Dantas *et al.* (2021b), over 50,000 missile launches were simulated to train a deep neural network for estimating the maximum launch range of a missile's Weapon Engagement Zone (WEZ), achieving high predictive accuracy while reducing the need for further simulations. This study used the same missile model implemented in ASA.

In Dantas *et al.* (2021a), 3,729 BVR engagements were simulated to train a supervised learning model aimed at supporting the decision of when to engage enemy aircraft during Defensive Counter Air (DCA) missions. The model was trained using simulation-derived operational metrics to capture relevant combat dynamics.

In Dantas *et al.* (2022), ASA/AEROGRAF was used to simulate 10,000 BVR combat scenarios, enabling the development of classifiers that improved pilot situational awareness. The models achieved high accuracy in distinguishing between offensive and defensive conditions.

The work in Costa *et al.* (2022) focused on aircraft formation control using artificial potential fields and optimization techniques. ASA simulations helped identify configurations that improved formation coherence and mission success.

The study Lima-Filho *et al.* (2022) employed ASA to optimize tactical UAV formations under uncertainty in war game scenarios. Metaheuristics were applied to maximize the probability of success against opposing forces.

In Dantas *et al.* (2023), ASA was used to propose an architecture for training a deep reinforcement learning agent capable of autonomously learning BVR tactics.

The work in Dantas *et al.* (2023b) proposed an approach for analyzing surface-to-air missile engagement zones using prediction models to enable real-time estimation while reducing computational costs. The analysis was based on the same surface-to-air missile model implemented in ASA.

The work in Dantas *et al.* (2024) proposed new social navigation metrics to assess collaboration between human pilots and autonomous wingmen in air combat scenarios. It also outlined a validation experiment using ASA to simulate mixed human-autonomous formations.

In Viscardi *et al.* (2024), a defensive "winding maneuver" was proposed to improve aircraft survivability against surface-to-air missiles. Developed using ASA, the maneuver was optimized with a genetic algorithm and validated through simulations, statistical analysis, and operational metrics.

The work in Lima *et al.* (2024) proposed machine learning models to approximate ASA simulation outcomes in stochastic contexts, significantly reducing overall simulation time.

The study in Dantas *et al.* (2025) introduced a probabilistic kill estimation model for air-to-air missiles based on ASA simulations and stochastic target modeling. It employed the same missile structure used in ASA.

Finally, Dantas (2024) and Dantas *et al.* (2025) used pilot data from ASA/AERO-GRAF to train imitation learning models for replicating pop-up attack maneuvers. A variational autoencoder was employed to generate synthetic samples and augment the dataset, improving the robustness and performance of the models.

These studies collectively demonstrate the flexibility and effectiveness of ASA as a simulation tool for operational analysis, autonomy development, and tactical decision support in air combat environments.

3.5 Outcomes

In this chapter, we presented a high-level overview of the ASA simulation framework, developed by IEAv since 2018, with the primary objective of supporting the evaluation of military operational scenarios relevant to FAB. The platform offers several distinguishing features, including the management of multiple simulation machines across one or more computers, as well as the dynamic loading `.so` files at runtime, batch execution of simulations, and the AsaDataScience module, an integrated data analysis platform tailored for military operational studies. Furthermore, we highlighted recent works that have employed ASA as a simulation tool to support air combat applications.

For future work, we plan to release part of the ASA source code, including its general architecture, to a selected group of organizations. This controlled release aims to encourage the development of diverse applications within the same simulation platform while ensuring proper control and security. Additionally, ASA is expected to evolve into a Simulation-as-a-Service (SimaaS) tool, supporting a wide range of simulation demands in the defense and aerospace domains (DANTAS *et al.*, 2023a). This initiative aims to promote greater interoperability among government, academia, and industry.

From a technical perspective, several enhancements are under consideration. First, we

aim to enhance scalability through parallel processing by enabling the flexible addition of simulation nodes as computational demands increase. Another area of focus is observability: ASA will incorporate mechanisms to monitor resource usage (such as CPU, memory, and disk), track feature utilization (identifying which components and libraries are used most frequently), and detect failures. These capabilities will enable better instrumentation, increased failure tolerance, and enhanced platform operational visibility.

We also aim to introduce accountability mechanisms to estimate and control the computational cost of each simulation, allowing resource usage to be better quantified and managed. Regarding the core simulation engine, currently based on an object-oriented paradigm with deep inheritance hierarchies, we are exploring a transition to a more modular and maintainable design. One promising alternative is the adoption of the entity-component-system (ECS) paradigm, which can improve model readability and simplify the integration of new components (MURATET; GARBARINI, 2020).

Additionally, improvements are planned for the internal structure of the AsaPy library, with efforts to reorganize its modules for better maintainability and extensibility. Finally, we will adapt the framework to support more modular and flexible decision-making processes in autonomous agent models by developing an artificial intelligence server in Python that integrates a rich ecosystem of specialized tools and libraries.

4 Simulation as a Service

This chapter explores the application of digital transformation through simulation services, highlighting the innovative use of simulation in enhancing operational capabilities. Simulation as a service system offers a disruptive approach to modeling, analyzing, and optimizing military operations in an era where technological advancements are required to maintain a strategic edge. This service-oriented architecture can seamlessly integrate simulation models, fostering agility and scalability in decision-making. As indicated in Figure 1.2, this chapter primarily contributes to the "Simulation Tools and Services" area within the proposed research framework.

The following work provides the foundation for this discussion:

> **DANTAS, J. P. A.**; GERALDO, D.; COSTA, A. N.; MAXIMO, M. R. O. A.; YONEYAMA, T. ASA-SimaaS: Advancing Digital Transformation through Simulation Services in the Brazilian Air Force. In: Simpósio de Aplicações Operacionais em Áreas de Defesa (SIGE2023). Proceedings [...]. São José dos Campos, Brazil: Instituto Tecnológico de Aeronáutica (ITA), 2023. p. 6. ISSN 1983 7402. Available at: `https://www.sige.ita.br/edicoes-anteriores/2023/st/2354551.pdf`.

4.1 Summary

This work explores the use of military simulations in predicting and evaluating the outcomes of potential scenarios. It highlights the evolution of military simulations and the increased capabilities that have arisen due to the advancement of artificial intelligence. Also, it discusses the various applications of military simulations, such as developing tactics and employment doctrines, training decision-makers, evaluating new acquisitions, and developing new technologies. The chapter then focuses on the Brazilian Air Force's efforts to create its own simulation tool, the Aerospace Simulation Environment (*Ambiente de Simulação Aeroespacial – ASA*), and how this cloud-based service called ASA Simulation as a Service (ASA-SimaaS) can provide greater autonomy and economy for the military force. The main contribution of this work is to present the ASA-SimaaS solution as a

means of empowering digital transformation in defense scenarios, establishing a partnership network, and improving the military's simulation capabilities and competitiveness.

4.2 Introduction

Predicting the outcome of clashes between opposing military forces has been a recurring demand since the formation of the first armies (BIDDLE, 1996). Historically, commanders relied on their experience, intuition, and intelligence gathered by their scouts to make strategic and tactical decisions. However, as warfare became more complex and the range of available weapons and tactics expanded, it became increasingly difficult to predict outcomes accurately (BIDDLE, 2004).

The use of computers in military simulations dates back to the 1950s when the US Army began to use mainframe computers to analyze large datasets and simulate battlefield scenarios (SOKOLOWSKI; BANKS, 2009). Since then, military simulations have become an essential tool for predicting and evaluating the outcomes of potential conflicts.

In recent years, the capabilities of military simulations have expanded significantly thanks to the increased processing power and evolution of artificial intelligence (AI) (DAVIS; BRACKEN, 2022). One of the most effective types of simulation is accelerated time simulation, which compresses the time required to simulate a conflict, allowing decision-makers to evaluate the outcome of a potential conflict more quickly (BAE *et al.*, 2016).

Armed forces use faster-than-real-time simulations for a range of purposes:

- **Select courses of action (SUMILE, 2013):** Accelerated time simulations enable military decision-makers to explore different courses of action and evaluate their potential outcomes. By simulating various scenarios, commanders can assess the significance of different strategies and make knowledgeable judgments about the best course of action.
- **Develop tactics and employment doctrines (CHRISTENSEN; SALMON, 2022):** Simulations provide a platform for developing and refining tactics and employment doctrines, helping military forces identify their strengths and weaknesses and make necessary adjustments to optimize their effectiveness on the battlefield.
- **Perform war games to train decision-makers (SCHWARTZ *et al.*, 2020):** War games conducted through accelerated time simulations serve as valuable training exercises for military decision-makers. These simulations allow commanders and staff to practice their decision-making skills in realistic scenarios, improving their ability to analyze complex situations, anticipate outcomes, and make effective strategic choices.

- **Evaluate new acquisitions (YUAN *et al.*, 2020):** Accelerated time simulations are instrumental in assessing the capabilities and effectiveness of new assets. By simulating the performance of new weapons systems, vehicles, or equipment, military forces can assess their potential impact on the battlefield and make instructed decisions regarding their acquisition and integration into existing forces.

- **Develop new technologies (MITTAL; DAVIDSON, 2021):** Simulation plays a crucial role in developing new technologies, particularly in autonomous systems. For example, AI algorithms for autonomous vehicles must be extensively tested in simulations before being deployed on real platforms. Simulations provide a safe and controlled environment to evaluate the performance and behavior of these technologies, identify potential issues, and refine them before actual deployment.

Although many commercial off-the-shelf solutions are available on the market to meet the demand for accelerated time simulations (LEE *et al.*, 2020), some armed forces have tried to develop their own simulation tools in recent years. The Brazilian Air Force (FAB) is one example of a military force that has taken steps to create its own simulation tool, known as the Aerospace Simulation Environment (*Ambiente de Simulação Aeroespacial – ASA* in Portuguese), a custom-made simulation framework developed by the Institute for Advanced Studies, that enables the modeling and simulation of military operational scenarios (DANTAS *et al.*, 2022a) (Figure 4.1). ASA allows the creation, configuration, and execution of defense scenario simulations through a cloud-based service called Simulation as a Service (SimaaS) (NATO, 2023). This new functionality, which is called ASA-SimaaS, could contribute to establishing a partnership network involving government agencies, academic institutions, and companies from the Brazilian Defense Industrial Base (BID), offering means for interoperability, cooperation, and competitiveness in the simulation field within the country.

Therefore, this chapter's main contribution is to present the ASA-SimaaS solution, which empowers digital transformation in FAB by providing a custom cloud-based simulation service for creating, configuring, and executing Defense scenario simulations.

The chapter is organized as follows. Section 4.3 provides a detailed account of the current process of acquiring and maintaining simulation software within the FAB, outlining its challenges and limitations. Section 4.4 discusses effectiveness and efficiency indicators that can be used to measure the success of the digital transformation process. Section 4.5 describes a web application that helps manage requests for updates, corrections, and new functionalities in ASA-SimaaS. Section 4.6 describes the process after the proposed digital transformation, i.e., the desired end state, focusing on the benefits and potential impact of the ASA-SimaaS on FAB's organizations. Finally, in Section 4.7, we provide some conclusions based on the information presented, highlighting the potential of ASA-SimaaS

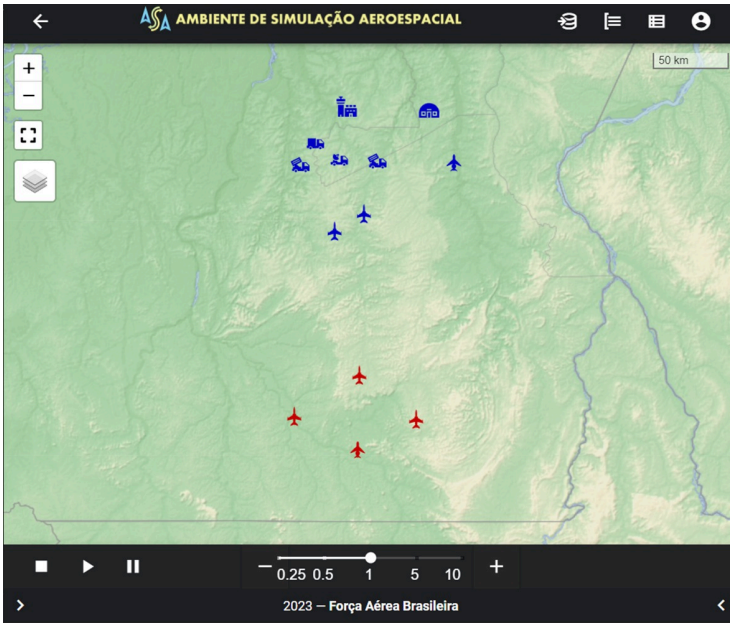


FIGURE 4.1 – The Aerospace Simulation Environment (ASA) user interface for scenario creation allows users to define multiple simulation aspects, including aircraft models, sensors, communication links, and mission objectives.

to positively impact FAB decision-making at all levels and the means of assistance to all elements of the defense triple helix: government, academia, and industry.

4.3 The Process Before the Digital Transformation

The initiatives and needs related to simulation within the FAB are managed in a decentralized way, i.e., different organizations have some sort of autonomy to address their simulation demands through specific and individual solutions that are typically acquired from the market.

These solutions often came with a high price tag, requiring significant investment from the owner, both in terms of purchasing the software and providing the necessary training to use it effectively. In addition, these solutions require constant maintenance and updates, which further increase the cost of ownership.

An organization can often face resource unavailability when attempting to acquire simulation software, and those who manage to acquire it can encounter difficulties in maintaining the software lifecycle. For example, license renewals and version updates are often necessary, which requires additional time, effort, and investment.

The process of acquiring simulation software typically involves several stages, as illustrated in Figure 4.2. First, the organization would identify the demand for simulation software and define the technical and functional requirements. Then, they would search for available options on the market and evaluate them against their requirements. If a

suitable solution was found, negotiations for purchase and licensing would occur. Once the software was acquired, the organization would need to install and configure it and then train personnel to use it effectively. Finally, continued maintenance and updates were required to guarantee that the software remained operational and up to date.

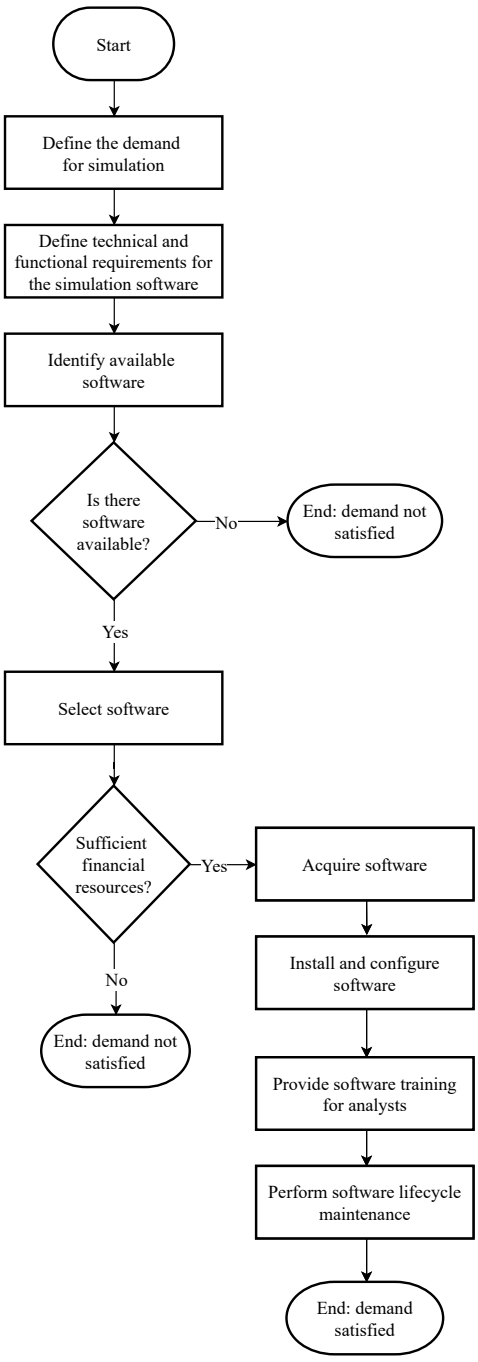


FIGURE 4.2 – Acquiring Simulation Software: A Typical Process Overview.

Overall, acquiring and maintaining simulation software is a time-consuming, resource-intensive, and expensive process that presents substantial challenges for any organization interested in its use. The ASA-SimaaS can be a reasonable alternative over the previous

approach, as it provides a centralized, cloud-based solution that can be accessible to all organizations within the FAB and eliminates many of the burdens associated with traditional software acquisition and maintenance.

4.4 Effectiveness and Efficiency Indicators

Establishing indicators that encompass the effectiveness and efficiency of the simulation systems is essential to facilitating a comprehensive evaluation of the proposed solution in relation to other available alternatives.

To correctly understand the proposed indicators, it's important to clarify the term "Defense Scenario" or simply "Scenario". This term describes a geopolitical and military context in which typical elements of armed conflicts are involved, such as fighter aircraft, tanks, warships, missiles, bombs, radars, artillery, and satellites (JAISWAL, 2012). Some elements, like fighter aircraft, require specialized agents for operation – for example, a pilot capable of handling an escort mission. In the case of accelerated-time simulations, human figures are modeled through AI tools. It is important to note that the models used in simulations must always undergo verification and validation processes to ensure the results are credible (HARTLEY, 1997).

The proposed indicators are meant to evaluate the effectiveness and efficiency of a simulation system that aims to support defense operations. The effectiveness indicator (`Effs`) measures the rate of meeting the demanded scenarios simulation, while the efficiency indicator (`Effy`) estimates the cost of simulating a single scenario.

Equation 4.1 shows how to calculate the effectiveness indicator based on the number of scenarios simulated (`NSS`) and the number of scenarios demanded (`NSD`):

$$\texttt{Effs} = \frac{\texttt{NSS}}{\texttt{NSD}} \tag{4.1}$$

The greater the diversity of scenario elements available in the simulation software, the higher the rate of meeting the demanded scenarios. Therefore, simulation systems that allow third-party extensions will perform better in the effectiveness indicator. Aspects such as the availability of resources for software maintenance and trained personnel to operate it will also impact the effectiveness indicator.

Equation 4.2 shows how to measure the efficiency indicator based on the total value invested in the acquisition and maintenance of the system in a given period (`TVI`) and the number of scenarios simulated using the system in the same period (`NSS`):

$$\texttt{Effy} = \frac{\texttt{TVI}}{\texttt{NSS}} \tag{4.2}$$

The goal of the efficiency indicator is to estimate the cost of simulating one scenario by the system. Ideally, the unit cost value should become lower over time, increasing the positive perception of the return on investment made in the system's acquisition.

It is essential to remark that the effectiveness and efficiency indicators are not standalone metrics and should be analyzed jointly with other parameters, such as the accuracy of the simulation outcomes, the realism of the scenario components, and the usability of the software. Moreover, these indicators should be used as guidelines for decision-making rather than the sole criterion for choosing a simulation system (JUNIOR *et al.*, 2021). The context in which the system will be used, the specific defense objectives, and the available resources should also be taken into consideration when selecting a simulation system.

4.5 ASA Management System

The ASA-SimaaS not only offers simulation tools but also incorporates a simple yet effective management system that enables standardized communication between users and service managers. This system is designed to ensure that all simulation needs of FAB are adequately cataloged and prioritized, thereby avoiding unnecessary acquisitions and maximizing the number of customers served. Therefore, the ASA Management System (AMS) is a useful web application that facilitates evaluating, prioritizing, and monitoring requests for updates, corrections, and new functionalities within ASA-SimaaS. It empowers service managers to efficiently respond to service access requests and effectively manage the entire lifecycle of the ASA Service. Overall, AMS is designed to coordinate simulation-related initiatives and needs throughout FAB. It promotes a more efficient and collaborative approach to organizational simulation needs management. The example in Figure 4.3 demonstrates the usage of AMS for system improvement.

The ASA community consists of two main user groups: user managers and ordinary users. User-managers have the authority to generate and fulfill demands, while ordinary users can only submit demands for consideration. User-managers are further categorized based on thematic areas, including:

- **Overall Manager:** coordinating the service, overseeing resource allocation, and acting as the primary Point of Contact (POC) for interactions with military and civil organizations;
- **Partnership Manager:** addressing requests from organizations seeking accreditation as partners/users of the service;

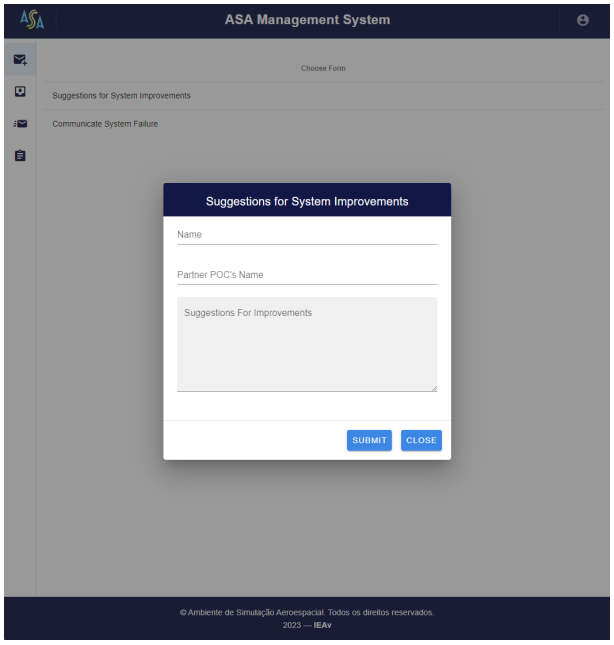


FIGURE 4.3 – AMS's form used for providing suggestions to generate system improvements.

- **Technical Manager:** evaluating demands that involve changes to the simulation framework or the inclusion/update of models compatible with the ASA;
- **Support Manager:** ensuring the correct operation and availability of ASA Service applications, excluding the simulation framework and models;
- **Modeling and Simulation Manager:** receiving requests related to scenario simulations, encompassing operational, strategic, and tactical needs;
- **Intel Manager:** receiving requests related to intelligence information for populating the scenarios; and
- **Research and Development Manager:** receiving requests related to research and development areas to support scenario generation and analysis.

Within the ordinary users group, several subgroups exist:

- **Partner POC:** representing the partner organization and handling institutional demands that benefit all members of the affiliated organization;
- **Basic User:** creating scenarios, simulating them, and visualizing their results. This is done through the Web Station, the main ASA user interface;
- **Analyst User:** analyzing the data generated by simulations, using the Data Analysis Platform, which facilitates the design of experiments, batch executions, and prediction model generation; and

- **Developer User:** adding new features by coding within the service, by accessing the source code repository, which enables the development of the simulation kernel as well as the behavioral, logical, and physical models.

Each user type (Figure 4.4) possesses distinct permissions within the simulation service, determining their scope of actions. Additionally, users have customized "views" within the AMS interface, allowing non-manager users to create specific demands and managers to evaluate and fulfill them.

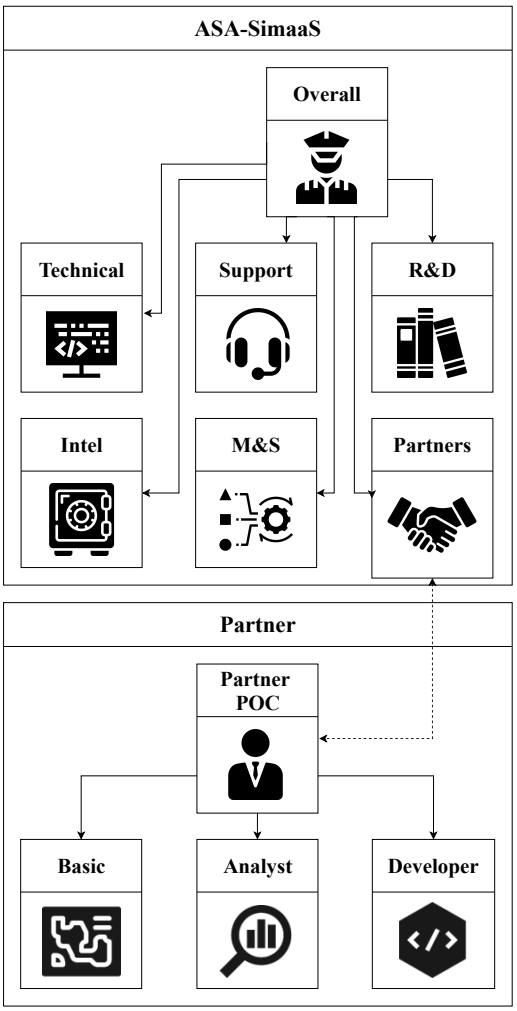


FIGURE 4.4 – ASA-SimaaS Users: Relationship between distinct user types, each carrying unique permissions that shape their roles.

The Partner POC represents a special user role since they are responsible for reporting simulation needs that are not yet met by the ASA-SimaaS. For instance, if a certain FAB organization that makes important operational decisions needs to evaluate a scenario that includes an aircraft model that is not yet available in the list of models provided to users, the POC can report their demand through the service management system. The service manager responsible for this type of request will then seek a solution by turning

to the network of partners. The flowchart in Figure 4.5 details the process for reporting a simulation demand.

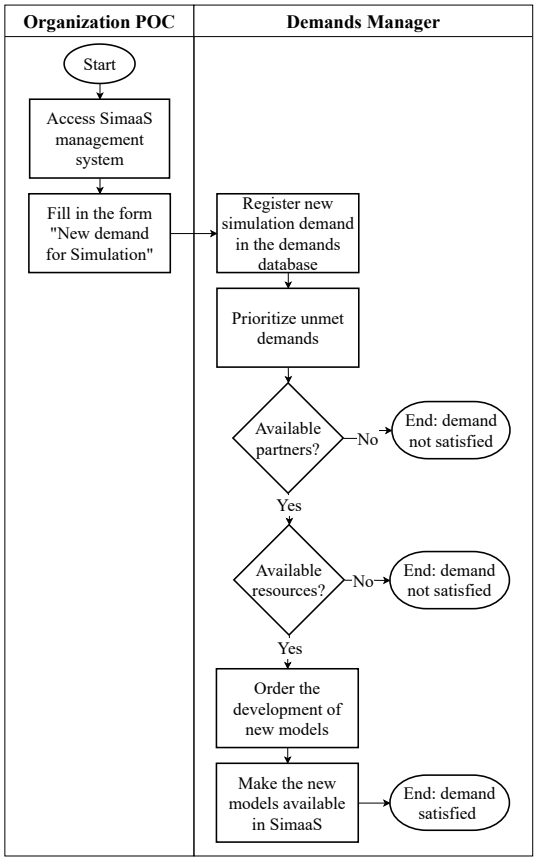


FIGURE 4.5 – Diagram illustrating the steps involved in generating a report for simulation demand.

The tools and services are further explained and exemplified in the service Documentation and Tutorials modules, which, along with the previously mentioned ones, are depicted in Figure 4.6.

4.6 Advancing Digital Transformation

The ASA-SimaaS is expected to positively impact several key areas within FAB, enhancing their capabilities and supporting critical analyses and activities. The following areas are anticipated to benefit from the utilization of this service:

- **Chief of Staff:** ASA-SimaaS could provide valuable support for analyses related to Simulation-Based Acquisition (SBA). With the ability to simulate a wide range of scenarios and evaluate the performance of various systems and technologies, a Chief of Staff can make informed decisions regarding future capabilities and acquisitions.
- **Operational Commands:** ASA-SimaaS could play a vital role in supporting the

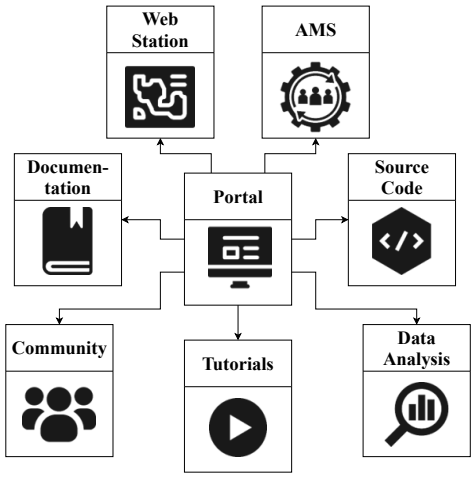


FIGURE 4.6 – ASA-SimaaS services: aimed at providing comprehensive solutions to assist the simulation service.

development of doctrines and tactics. By providing a platform for conducting detailed simulations, an Operational Command could analyze different tactical approaches, evaluate their effectiveness, and refine operational doctrines accordingly. ASA-SimaaS also could serve as a valuable tool in the selection of Courses of Action (COA). By simulating various COAs and evaluating their outcomes, one could identify the most effective strategies for specific operational scenarios.

- **Command and Staff College:** War Games are essential for training and evaluating military personnel in realistic scenarios. ASA-SimaaS could be used to support War Game activities, improve the efficacy of training exercises, and facilitate the evolution of military personnel's strategic thinking and decision-making skills.

- **Research and Development:** ASA-SimaaS could play a special role in supporting the development of Artificial Intelligence (AI) algorithms for a wide range of purposes. By utilizing the simulation environment provided by ASA-SimaaS, researchers can assess and optimize AI algorithms for combat systems and air traffic control applications.

The utilization of ASA-SimaaS across multiple organizations can promote collaboration, efficiency, and effectiveness, ultimately contributing to the Brazilian Air Force's overall readiness and success.

The ASA-SimaaS relies heavily on cloud computing technology to be offered as a web service. Complex simulations, such as defense scenarios, typically require large computational power for processing. In this case, the simulations demanded via the web are not processed locally on users' computers but in a corporate data center. The data center has

a complex Information Technology (IT) infrastructure (hardware and software), available 24/7, that performs demand scaling and parallel execution of simulations. In summary, ASA-SimaaS can enable the organization to perform simulations of complex scenarios at any time of the day without relying on high-powered personal computers.

4.7 Outcomes

In conclusion, the introduction of ASA-SimaaS can streamline the digital transformation process of FAB, providing users with a powerful tool for simulating complex defense scenarios and enabling them to optimize the use of high-performance computing resources available in a corporate data center while simultaneously reducing costs associated with software licensing and maintenance.

One of the main features of ASA-SimaaS is its management system, which facilitates seamless communication between users and service managers. This ensures that users can systematically report simulation needs that are not currently addressed. Such information is vital as it helps inform acquisition strategies, enables the prioritization of resources, and ensures that the service evolves in concert with user needs. ASA-SimaaS has the potential to become an integral part of decision-making processes at all levels within FAB, whether they be tactical, operational, or strategic. The robust infrastructure supporting ASA-SimaaS has been meticulously crafted to provide to the defense ecosystem's diverse elements, including government entities, academic institutions, and the defense industry.

Looking toward the future, ASA-SimaaS holds significant potential for enhancement. Improvements in integrating AI could add a new layer of intelligence to the simulations, making them more capable of predicting and adapting to real-world complexities. Another potential development could include real-time collaboration tools, fostering a cohesive environment where users can share insights and make collective, data-driven decisions. Furthermore, integrating ASA-SimaaS with augmented and virtual reality technologies could result in immersive simulation experiences that closely resemble the intricacies of real-life defense scenarios, which could be made available throughout FAB. A critical factor for sustained success will be ensuring that ASA-SimaaS remains scalable and agile, capable of adapting to the ever-changing landscape of defense needs.

In summary, ASA-SimaaS represents a substantial stride in digitizing defense activities, as it empowers FAB to optimize resources, streamline decision-making processes, and enhance support for critical missions. There are high expectations that ASA-SimaaS will positively impact FAB organizations in the years to come, contributing to Brazil's maintenance of a robust and efficient defense posture.

Part III

Weapon Systems

5 Engagement Zone for Air-to-Air Missiles

Accurate prediction of the Weapon Engagement Zone (WEZ) and maximum launch range of missiles is essential for modern air combat. Traditional methods, relying on precomputed data and heuristics, can be inefficient and inflexible. This chapter explores a novel approach using machine learning to estimate the maximum launch range within a WEZ. As indicated in Figure 1.2, this chapter primarily contributes to the "Weapon Systems" area within the proposed research framework.

This chapter is derived from the following work:

> **DANTAS, J. P. A.**; COSTA, A. N.; GERALDO, D.; MAXIMO, M. R. O. A.; YONEYAMA, T. Weapon Engagement Zone Maximum Launch Range Estimation Using a Deep Neural Network. In: BRITTO, A.; DELGADO, K. V. (Ed.). Intelligent Systems. Proceedings [...]. Cham: Springer, 2021. p. 193–207. ISBN 978 3-030-91699-2.

5.1 Summary

This work investigates the use of a Deep Neural Network (DNN) to estimate the maximum launch range of the Weapon Engagement Zone (WEZ). The WEZ allows the pilot to identify airspace where the available missile is more likely to successfully engage a particular target, i.e., a hypothetical area surrounding an aircraft where an adversary is vulnerable to a shot. We propose an approach to determine the WEZ of a given missile using 50,000 simulated launches in various conditions. These simulations are used to train a DNN that can predict the WEZ when the aircraft finds itself in different firing conditions, with a coefficient of determination of 0.99. It provides another procedure concerning preceding research since it employs a non-discretized model, i.e., it considers all directions of the WEZ at once, which has not been done previously. Additionally, the proposed method uses an experimental design that allows for fewer simulation runs,

providing faster model training.

5.2 Introduction

Within simulated computational environments, military systems must resemble reality in a level of fidelity that leads to useful conclusions (HANCOCK *et al.*, 2008). This is done through the use of reliable computational models that are deemed to encompass the main characteristics of the systems they represent (HILL *et al.*, 2001).

When dealing with air combat, the missile is one of the most critical parts to be modeled. This is true concerning both the missile system itself and the decision of when to employ it, i.e., to fire. That is even more critical when considering Beyond Visual Range (BVR) air combat since this decision must be taken based only on what the situational awareness systems display to the pilot (DANTAS, 2018).

In the context of constructive simulations, in which the aircraft behave autonomously, their controlling algorithms need to provide data similar to what real pilots would receive so that the behaviors perform in accordance (COSTA, 2019). One of the most important aspects that a pilot can use to decide whether to launch a missile on an opposing aircraft is the Weapon Engagement Zone (WEZ), which, in simple terms, represents the range of the weapon (DANTAS *et al.*, 2021a). This definition is discussed with more depth further in Section 5.3.1. The determination of this range is not a simple task, however, since it is influenced by a series of variables from both the shooter and the target. Moreover, it is naturally dependent on the missile itself. In this work, we propose an approach to determine the WEZ of a given missile using a series of simulated launches in various conditions. These simulations are used to train a machine learning algorithm that can predict the WEZ when the aircraft finds itself in different firing conditions. Previous works have employed some types of Artificial Neural Networks (ANN), such as Wavelet Neural Networks (WNN) (YOON *et al.*, 2010) and a Multi Layer Perceptron (MLP) with Bayesian Regularization of Artificial Neural Networks (BRANN) (BIRKMIRE; GALLAGHER, 2012), to make predictions of the WEZ, also from previously simulated data. Purely mathematical approaches are also available within the literature, such as Farlik *et al.* (2017), Li *et al.* (2020), but they provide an intermediate step between unrealistic missile models that consider fixed missile ranges and more complex models based on simulations.

Much more research may have been developed within companies and governments concerning WEZ determination (BIRKMIRE, 2011), but this is still seldom publicly available. The contribution of this work is employing a Deep Neural Network (DNN) with a novel non-discretized model, i.e. the model considers all directions of the WEZ at once, not discretizing the off-boresight angle (Figure 5.5) as done previously to the best of our

knowledge. Additionally, it uses an experimental design that allows for a lower number of simulation runs, which provides faster training of the model.

The remainder of this chapter is organized as follows. Section 5.3 provides the background, explaining in more depth the concept of WEZ and presenting the particular missile model employed and the experimental design utilized. In Section 5.4, the proposed methodology is detailed, whereas the results coming from it are presented and analyzed in Section 5.5. Finally, Section 5.6 states the main conclusions of the work and suggests some future developments.

5.3 Background

In this section, we detail the concept of WEZ, present the missile model, and specify the simulation experimental design used within this work.

5.3.1 Weapon Engagement Zone

The term WEZ may present different definitions throughout the military domain. According to the United States Department of Defense (DEPARTMENT OF DEFENSE, 2021), WEZ can be described as an "airspace of defined dimensions within which the responsibility for engagement of air threats normally rests with a particular weapon system." Although a rather broad definition, its focus resides on the responsibility for the engagement of a target inside the zone by a specific system.

In our work, on the other hand, we are more focused on the airspace defined by the range of a weapon system (missile), which is not necessarily responsible for engaging all threats within this zone. This is rather a possibility, that is, the WEZ in our case allows the pilot to identify an airspace in which the missile available has a larger probability of being successful in engaging a particular target. In other words, the definition of WEZ adopted by us is similar to what Portrey *et al.* (2005) present: a hypothetical area surrounding an aircraft in which an adversary is vulnerable to a shot. This concept can be found in the literature under different terminologies, which may present subtle variations on meaning, such as Launch Acceptability Region (LAR) (YOON *et al.*, 2010) and Dynamic Launch Zone (DLZ) (ALKAHER; MOSHAIOV, 2015).

Figure 5.1 presents a simplified depiction of a WEZ, which stretches from the minimum range R_{min} to the maximum range R_{max}. The R_{max} is defined by us as the maximum distance in which the missile will hit a non-maneuvering target, which means if the target performs any maneuver, the missile will most likely miss if fired at this distance. On the other limit of this zone, the R_{min} is the minimum distance required by the missile

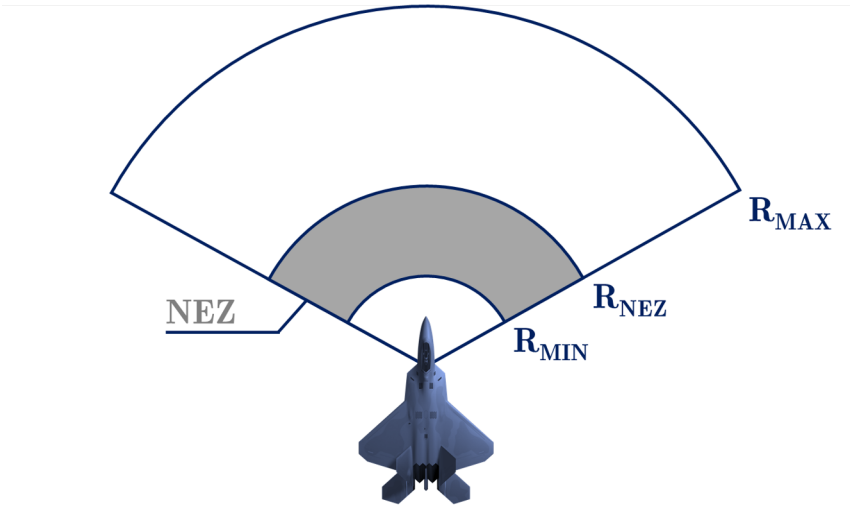


FIGURE 5.1 – Simplified WEZ representation.

to be able to properly activate its systems and, therefore, trigger its warhead. Between these two ranges, there is the no-escape zone (NEZ) range (R_{NEZ}), which represents a distance within which the target is very unlikely to be able to evade the missile, even when employing a high-performance defensive maneuver.

It is important to point out that the WEZ is also a function of the threat since it considers the target's parameters in its calculation. As Portrey *et al.* (2005) states, the WEZ is determined by many factors regarding the shooter and the target, such as "type of weapon, aircraft speed, relative altitudes, and geometry". These factors are used by the authors of Portrey *et al.* (2005) to define a metric that allows the pilot to know what is the amount of G-force that must be pulled to escape from an incoming missile. Therefore, their focus was less on the definition of the WEZ per se but rather on the determination of this particular metric.

On the other hand, Birkmire (2011) focuses on determining the WEZ for a missile in the context of virtual simulations, i.e., simulations in which real pilots interact with simulated systems. Therefore, his goal was to provide the pilots in virtual environments with a similar estimation of the WEZ as pilots in real aircraft have in their heads-up displays (HUDs) to support their decisions to fire a missile (Figure 5.2).

Our work has a slightly different focus since it aims to provide WEZ information to autonomous agents within a constructive simulation environment, i.e., a simulation in which simulated pilots interact with simulated systems. In addition, we provide a map visualization of the estimated WEZ, which can be valuable within the analysis of this type of simulation.

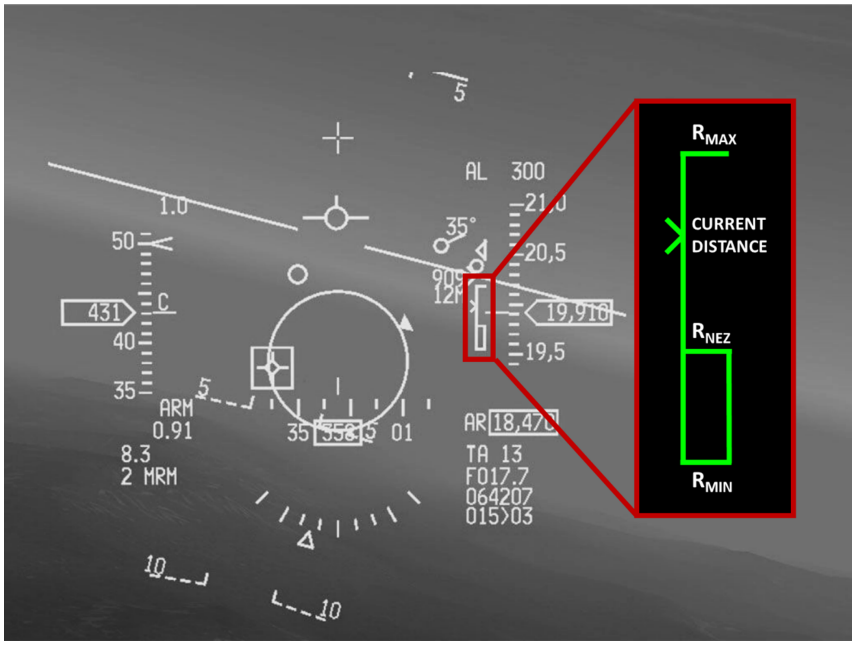


FIGURE 5.2 – HUD representation with a focus on the WEZ indication.
Source: Adapted from Kravchenko (2021).

5.3.2 Missile Model

Since this is not the focus of this work, and the methodology presented may be applied to any simulated missile, we provide a brief overview of the missile model. Our implementation is entirely done in the R programming language (IHAKA; GENTLEMAN, 1996), and it provides a simplified model with 5 degrees of freedom (5DOF) of a Fox 3 missile based on DEPARTMENT OF DEFENSE (1995). According to AIR LAND SEA APPLICATION CENTER (2020), Fox is a brevity code that refers to the guidance of a missile, in which type 3 stands for an active radar-guided missile, i.e., a missile that contains a seeker of its own that can track the target autonomously after reaching its activation distance. Still, regarding its guidance, the missile performs perfect proportional navigation to its target, maneuvering to comply with its guidance law, as well as a loft maneuver (i.e., an aggressive climb right after launch) whenever possible, as Figure 5.3 shows.

The model simulates the missile trajectory, considering a still or maneuvering target. To define the NEZ range, the simulation considers a high-performance maneuver of +5 G, which may be employed with a delay from the moment of launch. Some important metrics for the missile flight are provided in Figure 5.4.

Referring to Figure 5.4, the most straightforward metric is the mass (a). Since the missile operates with a boost-sustain motor (NOAMAN *et al.*, 2020), its mass decays almost linearly during its boost (burn) phase. Due to the loft maneuver, the angle of attack values (b) vary very aggressively at the beginning of the flight, which can also be observed on the pitch angle (theta) chart (c). Concerning heading (psi), there are some maneuvers that

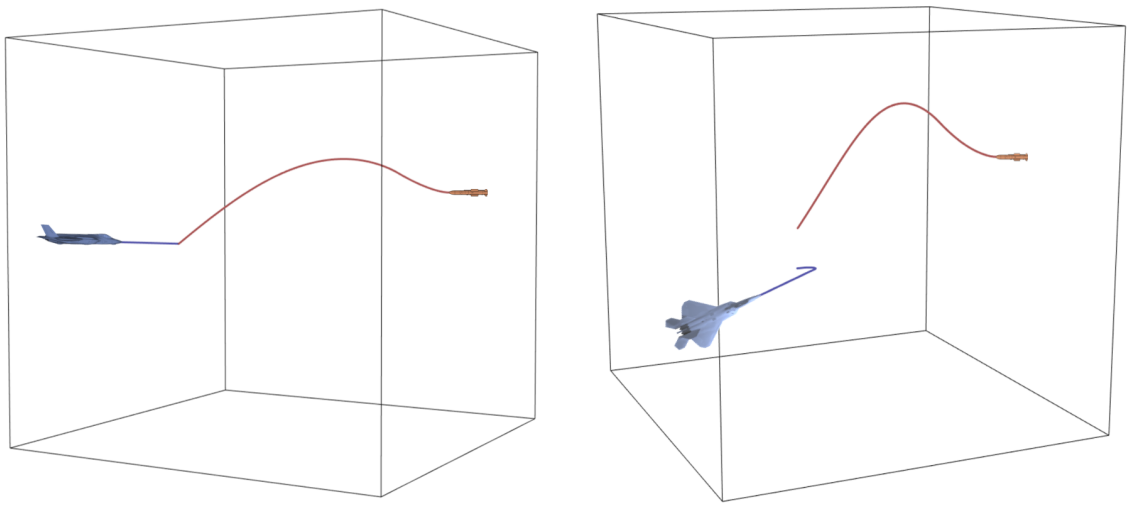


FIGURE 5.3 – Missile simulated trajectory samples.

the target employs to respond to high-performance evasion (d). Accelerations in both axes – East (e) and North (f) in the NED coordinate system (CAI *et al.*, 2011) – are also very abrupt due to the loft maneuver and the target response, respectively. The velocity (g) steadily increases during burn time and decays in the sustain phase. Finally, the seeker angle (h) accounts for the proportional navigation, being defined as the deviation of the shooter's longitudinal axis from the off-boresight angle (Figure 5.5).

5.3.3 Experimental Design

The parameters used as inputs to our missile model (Table 5.1) are very similar to the ones presented in Birkmire (2011), which makes it easier to compare our results with the ones obtained by it. However, instead of using an implementation in MATLAB Simulink (KLEE, 2018), our model was implemented entirely on R language as aforementioned, which has many prepackaged programs that help to solve analytical problems, prioritizing the simplicity of understanding and the parametrization. To provide a common understanding of the angles used, Figure 5.5 provides a depiction of them.

These parameters are selected based on operational experience and the missile model possibilities. The shooter's velocity and altitude are directly related to the energy that will be available to the missile. In particular, the launch altitude also influences the drag to which the missile will be subjected during flight, which is also true concerning the target altitude on the missile's final approach. Depending on its heading, the target's velocity can either help or hinder the missile's effectiveness. However, heading alone cannot provide a full account regarding positioning since this depends on the off-boresight angle

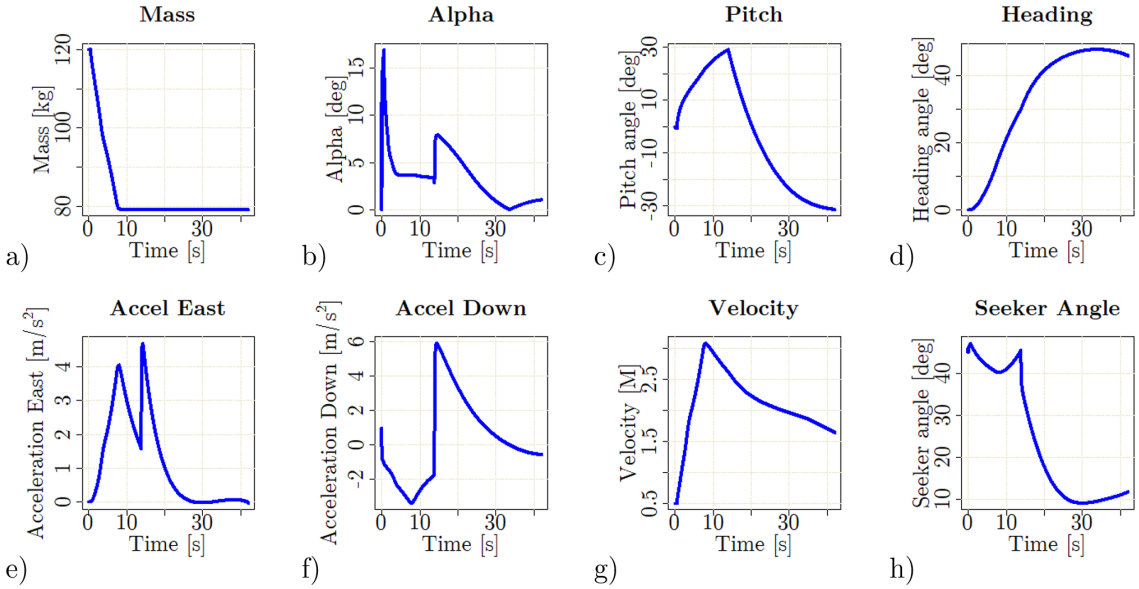


FIGURE 5.4 – Missile trajectory metrics samples.

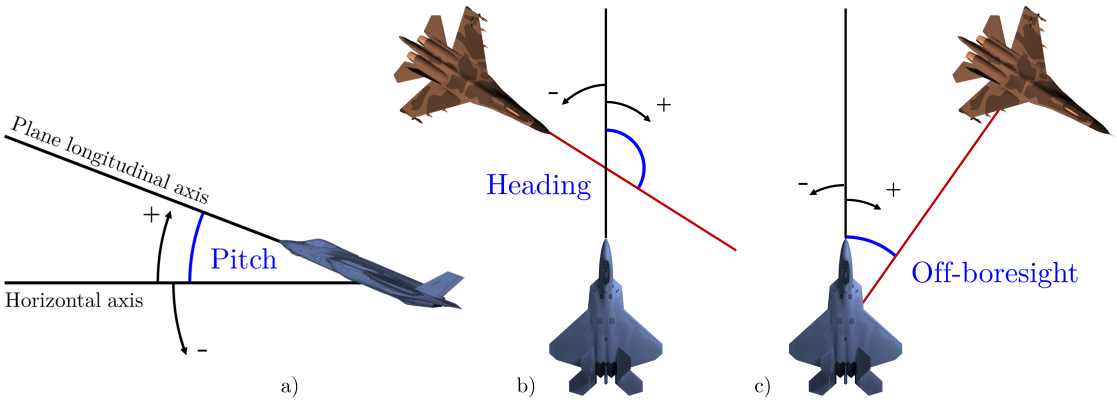


FIGURE 5.5 – Pitch (a), heading (b), and off-boresight (c) angles with respect to the target aircraft.

to determine whether the target aircraft is getting closer to the shooter and, therefore, to the missile itself. At last, the shooter's pitch angle at the moment of launch may help the initial maneuvering of the missile, that is, its loft maneuver.

Instead of a full factorial experiment as Birkmire (2011) presented, we tried to reduce the number of simulation runs by means of a more sophisticated design that takes into account randomness in its formation. Alternatively of a Monte Carlo simulation (MCS) that simply randomly samples the search space (HOMEM-DE-MELLO; BAYRAKSAN, 2014), we used the Latin Hypercube Sampling (LHS), which is deemed to be more efficient (DEUTSCH; DEUTSCH, 2012).

The LHS is a near-random method that aims to provide better search space coverage since a purely random approach may concentrate the samples by chance. Its main idea is to divide the multidimensional space so that the random samples are drawn from these

subdivisions instead of the whole search space (HUSSLAGE *et al.*, 2011). In our particular case, we employed a maximin algorithm, which attempts to optimize the sample through the maximization of the minimum distance between design points, fulfilling the constraints established by the LHS method. Table 5.1 presents the intervals for each variable used in the sampling. These limits were defined by subject matter experts, in this case, pilots, who considered meaningful values concerning their operational context.

TABLE 5.1 – Model parameters with the respective intervals considered.

Parameter	**Variable**	**Min**	**Max**	**Unit**
Shooter altitude	alt_sht	1,000	45,000	feet
Shooter velocity	vel_sht	400	600	knots
Shooter pitch	pit_sht	-45	45	degrees
Target altitude	alt_tgt	1,000	45,000	feet
Target velocity	vel_tgt	400	600	knots
Target heading	hdg_tgt	-180	180	degrees
Target off-boresight	rgt_tgt	-60	60	degrees

5.4 Methodology

This section describes the preprocessing, training, and evaluation of the DNN model applied to the simulation data.

5.4.1 Simulation

After creating the input batch files through LHS and using the limits presented previously, 50,000 simulations were run using 2 Intel Xeon Silver 4210R CPUs with 2.40GHz and 128 GB of RAM. All the simulations took approximately 7 hours to execute, and an output file containing the missile's maximum range for the respective input conditions was generated.

5.4.2 Preprocessing

From that, an Exploratory Data Analysis (EDA) was performed to identify general behaviors of the output data. The methods employed in this analysis were histogram, boxplot, correlation, and descriptive statistics.

Before performing the training of the ANN, some feature engineering techniques were employed. The first one was a form of encoding to better deal with cyclical features. The angles related to aircraft heading and off-boresight were encoded into their sine and cosine counterparts as done in Petneházi (2019), slightly increasing our model performance.

In addition, a form of handling potential outliers was to perform downsampling of the Latin Hypercube design. This was done because the pre-established intervals generated some improbable conditions. For instance, an aircraft at 1,000 ft firing on a target at 45,000 ft is exceedingly rare from the operational standpoint since a pilot would most likely increase its altitude before launching a missile. Therefore, we removed these undesirable samples, like the one presented, from the whole dataset based on subject matter expert operational knowledge, which can vary according to the mission type.

Lastly, data scaling was performed to equally distribute the importance of each input in the ANN learning process (PRIDDY; KELLER, 2005). This was done through a min-max scaler, which individually scales and translates all data features to a range from zero to one (BONACCORSO, 2017).

5.4.3 Model Training

Before training the DNN, a train-validate-test split was performed, allocating 80% for training and validation using a 5-fold cross-validation technique and 20% for testing. This division is done randomly and will allow the evaluation of the machine learning model later. The DNN model was formed by 12 layers of nodes, with the structure represented in Figure 5.6. All nodes have a rectified linear activation function (ReLU) (GOODFELLOW *et al.*, 2016).

In addition, the Adaptive Moment Estimation (Adam) optimizer was employed, an extremely popular training algorithm for ANN (BOCK; WEIß, 2019). Adam is a stochastic gradient descent method based on adaptive estimation of first- and second-order moments function (KINGMA; BA, 2014), which, in our case, aimed to minimize the Mean-Squared Error (MSE) loss. This was monitored by an early stopping method that checked whether the validation set metric had stopped improving (the patience, i.e., the number of epochs to wait before early stop if there was no progress on the validation set, was set to 20).

5.4.4 Model Evaluation

As the model being analyzed deals with a regression problem, the evaluation of the model will be carried out by observing the following metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and coefficient of determination (R^2).

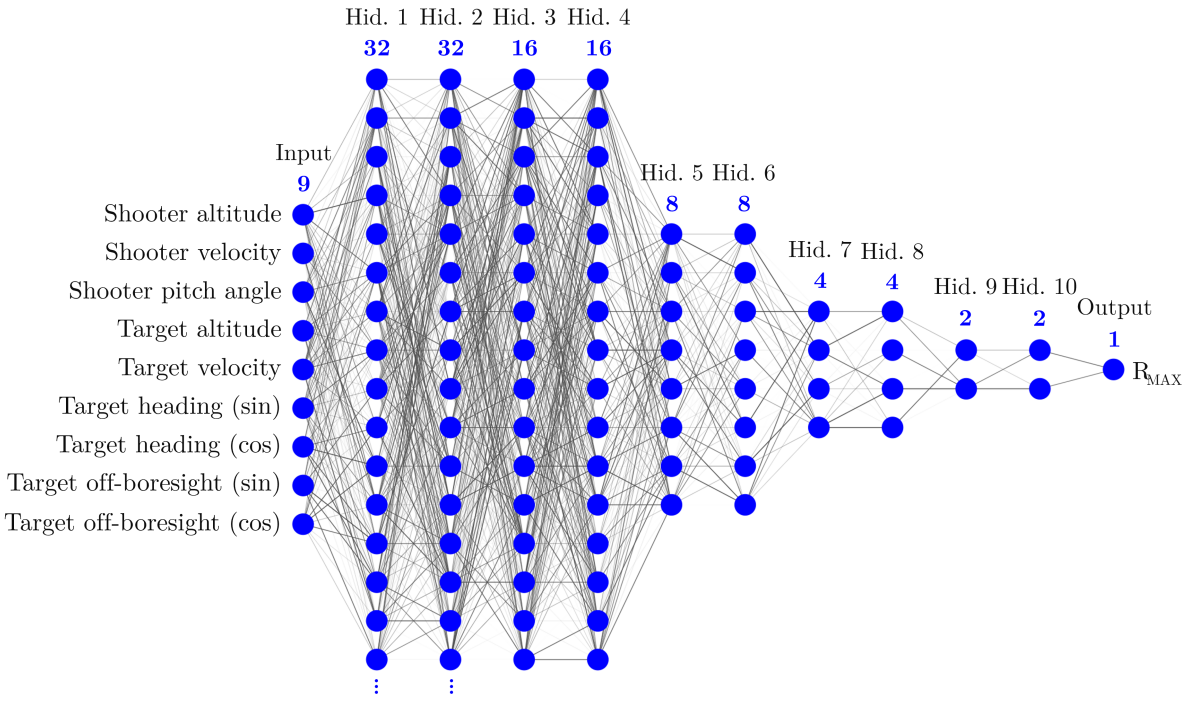


FIGURE 5.6 – Proposed DNN architecture.

5.5 Results and Analysis

This section examines the exploratory data analysis and the test dataset metrics. Additionally, it provides a Multi-Function Display (MFD) representation, focusing on the WEZ indication based on the proposed model.

5.5.1 Exploratory Data Analysis

Initially, an overview of the descriptive statistics of the model's input and output variables was observed, as shown in Table 5.2. The model's input variables follow a uniform distribution since these variables were sampled using the LHS. The model's output variable presents great variability, with an average of 12.38 NM and a standard deviation (std) of 9.37 NM. Notice that the mean and median (50%) vary by 3.24 NM, which indicates a considerable amount of outliers for this variable at the top of the distribution. These outliers will be eliminated from a superior threshold value (33.28 NM), which is not the maximum value (max) but is rather the largest value of the sampling excluding outliers, based on the interquartile range ($75\% - 25\%$). Observing the minimum (min), values of the order of 0.08 NM can be found, which shows that in the dataset, there are values in the target variable (`max_range`) that are smaller than the minimum activation distance of the missile modeled. This distance is considered 2 km (1.079 NM), the inferior threshold for this case. So that the model would not be harmed in its training to try to

predict the maximum missile range distance values, samples in which the model's output variable was smaller than the minimum missile activation distance were removed from the dataset. A histogram and a boxplot were generated together to visualize the target variable's distribution and thresholds, which can be seen in Figure 5.7.

TABLE 5.2 – Descriptive statistics of the model's input and output variables.

	alt_sht (ft)	**vel_sht (kt)**	**pit_sht (deg)**	**alt_tgt (ft)**	**vel_tgt (kt)**	**hdg_tgt (deg)**	**rgt_tgt (deg)**	**max_range (NM)**
mean	23,000.00	500.00	0.00	23,000.00	500.00	0.00	0.00	12.38
std	12,701.83	57.74	25.98	12,701.83	57.74	103.92	34.64	9.37
min	1,000.22	400.00	-45.00	1,000.82	400.00	-180.00	-60.00	0.08
25%	12,000.34	450.00	-22.50	12,000.32	450.00	-90.00	-30.00	5.55
50%	22,999.96	500.00	0.00	22,999.99	500.00	0.00	0.00	9.14
75%	33,999.75	550.00	22.50	33,999.76	550.00	90.00	30.00	16.64
max	44,999.38	600.00	45.00	44,999.42	600.00	179.99	60.00	40.87

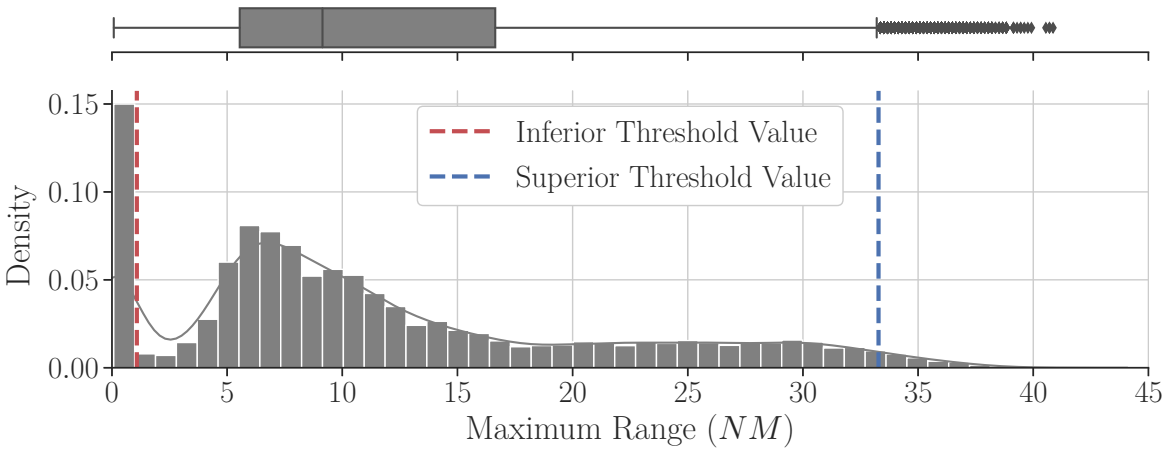


FIGURE 5.7 – Histogram and boxplot of the target variable.

Pearson's correlation analysis of the variables can be seen in the correlation matrix represented in Figure 5.8. Notice that none of the model's features has a strong correlation with each other, with the largest absolute value being only 0.30 between the shooter's altitude (`alt_sht`) and pitch (`pit_sht`). The performance of the algorithm may deteriorate if two or more variables are tightly related, which is called multicollinearity. We may also be interested in the correlation between input variables and the output variable (`max_range`) to provide insight into which variables may or may not be relevant as input for developing a model. Only the variables `alt_sht` and `pit_sht` have a slight correlation with the target variable. Although the correlation of `max_range` with `alt_sht` and `pit_sht` is intuitive, it is worth noting that this relationship highlights the importance of these variables in potentially influencing the range of the shot. This understanding can be very important in refining the model to better predict the maximum range based on these input variables.

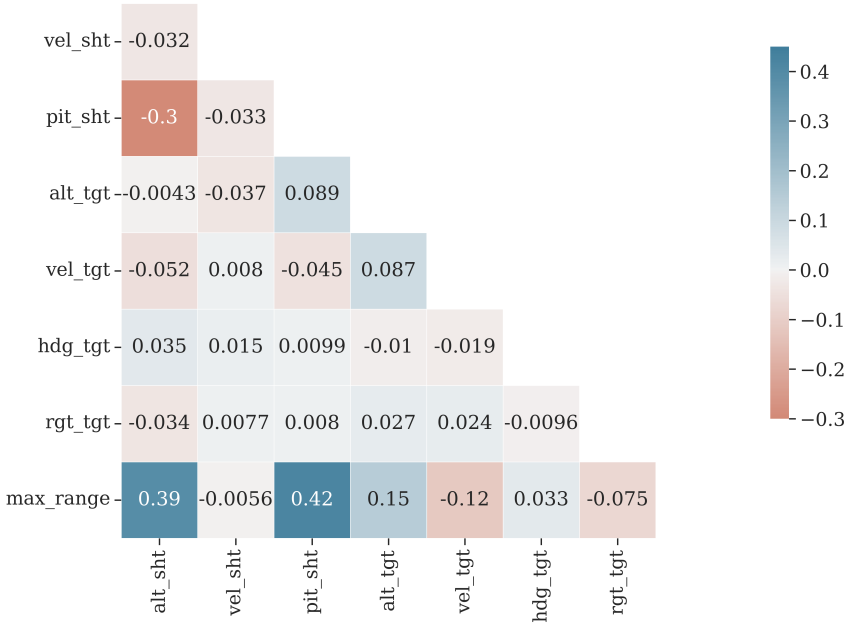


FIGURE 5.8 – Pearson correlation matrix of all model variables.

5.5.2 Model Predictions

Table 5.3 shows all the metrics used to evaluate the model with respect to the test set at the end of the training process. Very satisfactory results were found, with a coefficient of determination above 99%, which shows a very consistent model. In addition, note that the MAE was around 0.58 NM, which can be considered a very low value, considering that the values of the target variable have a mean of 13.13 NM with a standard deviation (std) of 8.58 NM. If we consider the RMSE, which penalizes the outliers' effects, the observed value is around 1.10 NM.

TABLE 5.3 – Metrics used to evaluate the DNN model at the end of the training process.

MAE (NM)	**MSE (NM^2)**	**RMSE (NM)**	R^2
0.58	1.23	1.10	0.99

A 5-fold cross-validation was conducted to estimate the skill of a machine learning model on unseen data and will help to understand our data better, giving much more information about our algorithm performance. The metrics of the five-folds were very similar, as shown in Table 5.4. The low variance found between this sample's folds demonstrates the model's consistency.

5.5.3 Model Representation

We estimated the WEZ Maximum Launch Range from the trained model using one of the samples from the test group. The target's position was varied by changing the

TABLE 5.4 – 5-fold cross-validations metrics.

	MAE (NM)	**MSE (NM^2)**	**RMSE (NM)**	R^2
1º Fold	0.54	1.06	1.03	0.99
2º Fold	0.62	1.22	1.10	0.99
3º Fold	0.71	1.39	1.18	0.98
4º Fold	0.52	1.08	1.04	0.99
5º Fold	0.57	1.34	1.16	0.98
mean	0.59	1.22	1.10	0.99
std	0.08	0.15	0.07	0.01

off-boresight values from $-60°$ to $+60°$ with steps of $0.5°$. An MFD representation with a focus on the WEZ indication can be seen in Figure 5.9. The curve that shows the missile's maximum range proved to be quite consistent, with a continuous aspect throughout the variations of off-boresight angles. Thus, we conclude that employing a different approach, unlike other research, with the incorporation of the off-boresight angle between the reference and the target aircraft as a feature in the model does not affect the performance of the WEZ estimation significantly since the supervised learning model used can be able to generalize well the results obtained in the training dataset to the test dataset.

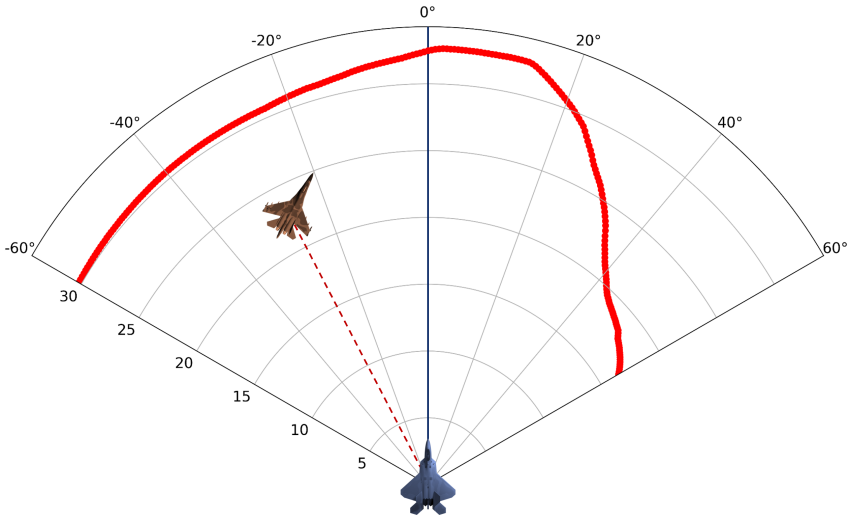


FIGURE 5.9 – MFD representation with a focus on the WEZ maximum range indication, with distances in nautical miles and angles in degrees.

5.6 Outcomes

Therefore, the main improvements advocated by us with respect to Birkmire (2011) is that, instead of discretizing the WEZ calculation concerning the off-boresight angle between shooter and target (Figure 5.4), creating, therefore, several ANNs, we approached the problem considering the whole space defined by the shooter's radar, with only one DNN being able to predict the values of WEZ. In addition, the number of simulation runs

was much lower (50,000 runs, as opposed to 222 million in Birkmire (2011)), which was achieved by a more carefully tailored experimental design.

In addition, in this work, a DNN with an MLP architecture was used, and brought better results than an ANN with only one hidden layer, as done in Birkmire (2011), comparing the coefficient of determination of both approaches applied to their respective datasets. In addition, different configurations of training and test groups were used in the dataset using the k-fold cross-validation. For a value of $k = 5$, the training and test group was sampled 5 times, and the results found were quite similar among all samples, demonstrating the consistency of the DNN model presented in this work.

The use of feature engineering techniques, with the creation of other model input variables, such as the use of sine and cosine for the variables that represent heading and off-boresight angles, also contributed to the greater adequacy of the model to the dataset collected from the simulations. Furthermore, it was observed, with the use of operational knowledge, that some of the samples collected would be unlikely to occur in a real air combat environment. These cases were when, for example, at a given altitude, the speeds of a given agent should meet at certain speed intervals. In the dataset, some samples did not respect these intervals, which could impair the model's performance, as they tried to predict cases that would most likely not occur in a real situation. To avoid these problems, these samples were eliminated from the dataset.

Future work should investigate how possible improvements in the architecture used for the DNN can bring better results and be more efficient, i.e. with a lower computational cost in the training process. In addition, the results found in this work can be compared with the use of other supervised machine learning techniques. These comparisons will help to determine the most appropriate methodology for calculating WEZ. In addition, in future work, it is possible to calculate not only the maximum range but also the distances related to the NEZ or even intermediate distances that could provide pilots with more assertive information about the probability of a missile reaching its target. Also, more advanced simulation models of the missile may be used in the future to provide better reliability to the presented results.

5.7 Source Code

The source code used in this work is publicly available and can be accessed at the following link: `https://github.com/jpadantas/bracis2021`. This repository contains the data preprocessing scripts, model training and evaluation scripts, and supplementary materials that support the results and analyses presented in this chapter, except for the missile simulator launch data, which has restricted access.

6 Engagement Zone for Surface-to-Air Missiles

The analysis of engagement zones for surface-to-air missiles is another very important aspect of modern defense strategies, providing essential insights into missile systems' operational capabilities and limitations. In this chapter, we explore methodologies that leverage advanced simulation techniques and machine learning algorithms to enhance the accuracy and agility of engagement zone predictions. As indicated in Figure 1.2, this chapter primarily contributes to the "Weapon Systems" area within the proposed research framework.

The basis for this discussion is provided by the following work:

> **DANTAS, J. P. A.**; GERALDO, D.; MEDEIROS, F. L. L.; MAXIMO, M. R. O. A.; YONEYAMA, T. Real-Time Surface-to Air Missile Engagement Zone Prediction Using Simulation and Machine Learning. In: Interservice/Industry Training, Simulation and Education Conference (I/ITSEC). Proceedings [...]. Orlando, FL, USA: National Training and Simulation Association (NTSA), 2023.

6.1 Summary

Surface-to-Air Missiles (SAMs) are essencial in modern air defense systems. A critical aspect of their effectiveness is the Engagement Zone (EZ), the spatial region within which a SAM can effectively engage and neutralize a target. Notably, the EZ is naturally related to the missile's maximum range; it defines the furthest distance at which a missile can intercept a target. The accurate computation of this EZ is essential but challenging due to the dynamic and complex factors involved, which often lead to high computational costs and extended processing times when using conventional simulation methods. In light of these challenges, our study investigates the potential of machine learning techniques, proposing an approach that integrates machine learning with a custom-designed simulation tool to train supervised algorithms. We leverage a comprehensive dataset of

pre-computed SAM EZ simulations, enabling our model to predict the SAM EZ for new input parameters accurately. It accelerates SAM EZ simulations, enhances air defense strategic planning, and provides real-time insights, improving SAM system performance. The study also includes a comparative analysis of machine learning algorithms, illuminating their capabilities and performance metrics and suggesting areas for future research, highlighting the transformative potential of machine learning in SAM EZ simulations.

6.2 Introduction

Surface-to-Air Missiles (SAMs) are indispensable in modern air defense systems, forming a robust line of defense against airborne threats (OPEN'KO *et al.*, 2017). The operational effectiveness of these missile systems depends heavily on accurately determining their Engagement Zone (EZ) — the spatial region in which a SAM can successfully engage and potentially eliminate a target (LEONARD, 2011). However, this parameter's complexity is influenced by many factors, including the missile's propulsion and guidance systems, the target's unique characteristics, such as speed, altitude, off-boresight angle, and the evasive maneuver pattern (ALBERTS *et al.*, 1999). The dynamic nature of these factors necessitates robust, efficient simulation tools capable of accurately predicting the EZ, a critical factor in formulating military strategies (DANTAS *et al.*, 2021b).

Nevertheless, conventional simulation methods for estimating missile EZs have been associated with high computational costs and lengthy processing times. Researchers have consistently pointed out that these limitations can significantly interfere with the speed and effectiveness of defense strategies (BIRKMIRE, 2011). Given these challenges, this study explores the potential of machine learning techniques as an alternative approach to address these problems.

By leveraging machine learning algorithms, known for their effectiveness in optimizing complex computational processes (SUN *et al.*, 2020a), this study aims to aid the computational limitation traditionally associated with SAM EZ simulations. Using a comprehensive dataset of pre-computed SAM EZ simulations, our proposed method integrates machine learning with a custom-designed simulation tool to train supervised algorithms – renowned for effectively addressing such issues. This approach, inspired by Dantas *et al.* (2021b), uses the extensive training phase to enable our model to accurately predict the EZ of a SAM for a new set of input parameters.

Despite the promising potential of this approach in addressing defense-related issues, it also poses distinct challenges. Specifically, when dealing with missile launch data, one has to navigate through obstacles such as the need for an extensive amount of training data, the risk of model overfitting, and the perpetual necessity for meticulous evaluation and

refinement to assure model precision and applicability (DANTAS *et al.*, 2021a). Nonetheless, these challenges also provide a route for further research and improvement, underscoring the continual need for innovation in this area.

The main contribution of this work is to highlight the transformative potential of machine learning techniques to speed up the response to SAM EZ simulations. It can contribute significantly to air defense strategy planning and potentially enhance SAM system performance by offering real-time insights. In doing so, we aim to combine the strengths of simulation and machine learning to bring a new level of dynamism and responsiveness to military strategy, enabling more informed and timely decision-making. Furthermore, our work includes a comparative analysis of widely recognized machine learning algorithms, emphasizing their performance metrics, training, and inference durations. This comparison offers a better understanding of these algorithms' capabilities and areas necessitating enhancement.

The structure of this chapter extends as follows: The subsequent section delves into a comprehensive review of the related work, setting the context for the study. This is followed by a detailed description of our research's simulation and analysis tools. Subsequently, we elucidate the methodology for developing machine learning algorithms to calculate the SAM EZ. After that, we present the results of our investigation and engage in a thorough discussion of their implications. Finally, we draw our conclusions and propose directions for future research.

6.3 Related Work

The simulation of missile-target engagements, essential for evaluating and developing defense systems, has been a research focus for several decades. Traditional simulation methods have been noted for their accuracy but are often constrained by high computational costs and lengthy processing times. This has spurred the investigation of alternative approaches, including machine learning techniques, to optimize efficiency.

Mathematical models for missile-target engagements are among this domain's foundational literature. For instance, Phillips (1991) presented a seminal work in which the Engagement Envelope Generator (EEG) was developed. The EEG, designed modularly, could be integrated with various mid-course missile trajectory simulations. The Rapid Estimation of Terminal Performance (RETP) algorithm, which utilizes linear adjoint techniques, was developed to estimate the miss distance and probability of hit rapidly. Philips' multilevel approach suggests the use of post-processing Monte-Carlo simulations for improved accuracy in analyzing intercepts along the EEG-generated engagement envelopes' boundaries.

Building on the earlier mathematical models, Farlik *et al.* (2017) investigated military modeling and simulation, emphasizing troops' training and military scenario verification. They highlighted the partial development of simulations for ground-based air defense systems and discussed an approach for modeling missile systems' firing capabilities. This involved simplifying missile systems' effective coverage zones for seamless integration into military simulators.

Furthermore, Li *et al.* (2020) developed a refined model for air-to-air missile attack zones by incorporating off-axis angles, allowing missiles to hit targets from different angles. They used the dichotomy search and fourth-order Runge-Kutta methods to calculate missile trajectories and determine hit conditions. Their results showed variations in the attack zone boundaries based on the target entry and missile off-axis angles, providing valuable insights for pilots in deciding whether to launch missiles.

Recent studies have made headway in employing machine learning techniques, particularly neural networks, to predict the Weapon Engagement Zone (WEZ). Yoon *et al.* (2010) used a Wavelet Neural Network to improve the accuracy of the weapon launch acceptability region, with promising results in terms of both accuracy and memory requirement reductions.

Similarly, Birkmire (2011) investigated a Multilayer Perceptron (MLP) with Bayesian Regularization to approximate the WEZ's maximum launch range. The network was trained on simulated data and showed improved approximation accuracy, demonstrating the feasibility of integrating ANNs into practical models.

Additionally, Dantas *et al.* (2021b) demonstrated using a Deep Neural Network (DNN) to estimate the WEZ's maximum launch range. By training the DNN on simulated launches, the model could efficiently predict the WEZ under various firing conditions. This study illustrated the synergy of simulation and machine learning in improving computational efficiency in missile-target engagement predictions.

In summary, the existing literature demonstrates a progression from mathematical models to the application of machine learning techniques in missile-target engagement simulations. This evolution highlights the trade-off between the accuracy of models and computational efficiency. Our research takes inspiration from the work of Dantas *et al.* (2021b) to integrate simulation with machine learning, focusing on the real-time prediction of SAM EZ under various firing conditions. In contrast to the aforementioned study, which focused on air-to-air missiles and analyzed missile models for a broad interval between 0 and 180 degrees, our work is tailored to SAM systems and conducts a more granular analysis by dividing models into sectors based on aspect angles. Additionally, we systematically compare three supervised machine learning algorithms in terms of performance and time response, as opposed to the sole use of a DNN in the prior study. This

multifaceted approach enables us to address the unique challenges associated with SAM systems.

6.4 Simulation and Analysis Tools

This study employs a simulation tool adapted from the model proposed by Dantas *et al.* (2021b), modified to model Surface-to-Air Missile (SAM) dynamics rather than air-air missiles. Constructed in the R programming language, the tool effectively leverages R's robust statistical capabilities and advanced graphical functions, making it suitable for simulating SAM dynamics (IHAKA; GENTLEMAN, 1996). Our tool features an intricately designed 5-degree-of-freedom (5DOF) model to accurately simulate a Fox 3 missile – an active radar-guided missile with an autonomous target-tracking seeker post-activation (DEPARTMENT OF DEFENSE, 1995).

Our tool emulates this guidance functionality, facilitating proportional navigation alongside an aggressive post-launch climb—commonly known as a loft maneuver (DEPARTMENT OF DEFENSE, 1995). Its adaptive capacity allows for the simulation of missile trajectories considering both stationary and maneuvering targets, thereby comprehensively exploring diverse engagement scenarios. These characteristics are illustrated in Figure 6.1.

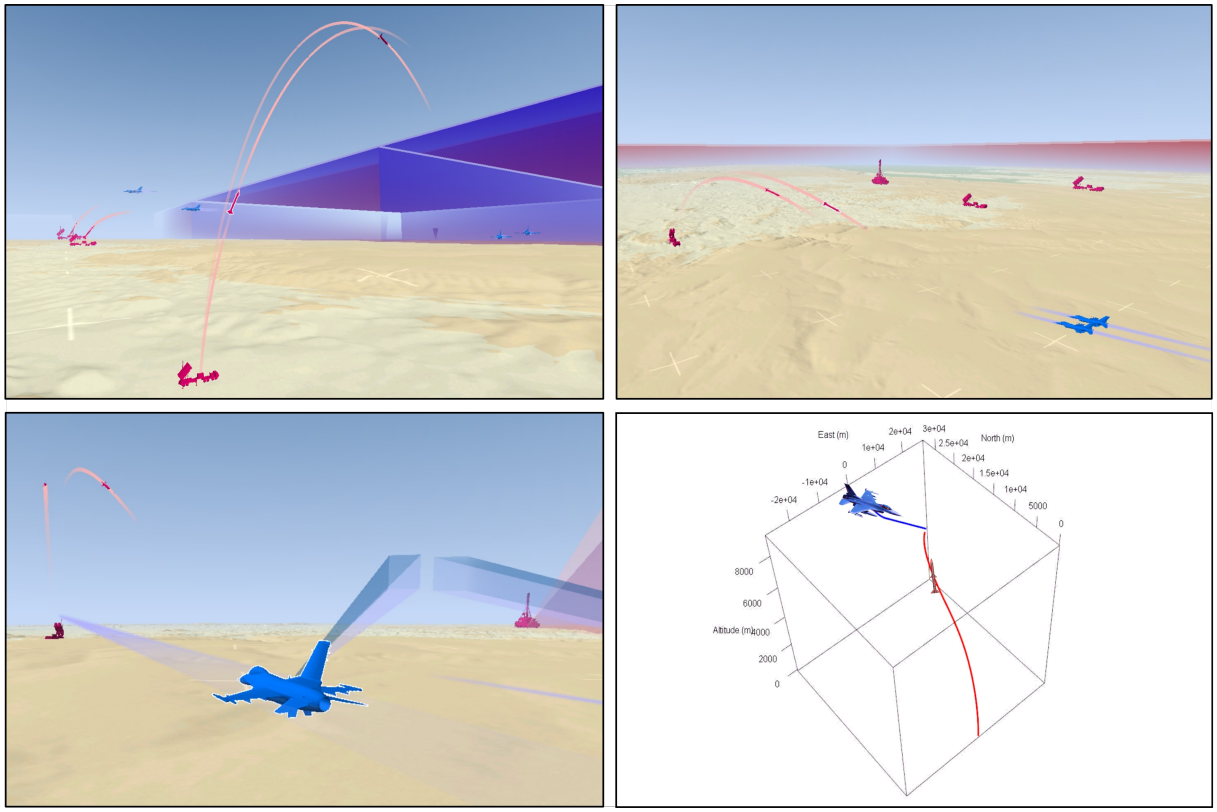


FIGURE 6.1 – Illustration of a red simulated missile engagement against a blue target.

This tool represents SAM dynamics with high fidelity and extends beyond numerical experiments to offer a scalable platform for creating, testing, and refining SAM models. Its potential for further expansion includes incorporating complex scenarios, diverse missile types, and environmental factors, making it an invaluable asset in EZ prediction and broader military strategy development.

We use the Aerospace Simulation Environment (*Ambiente de Simulação Aeroespacial – ASA in Portuguese*) to enhance launch visualization and debugging processes. This high-fidelity C++ simulation framework, developed specifically for the Brazilian Air Force (DANTAS *et al.*, 2022a; DANTAS *et al.*, 2023a), facilitates the modeling and simulation of diverse military operational scenarios. For the analysis of simulation data, we employ AsaPy (DANTAS *et al.*, 2024), a custom-made Python library associated with ASA, specifically designed to simplify and optimize the analysis of simulation data and is an integral component of the ASA technology. This library augments our data preprocessing, feature extraction, and machine learning algorithm training and evaluation stages, improving the overall efficiency of our study.

6.5 Methodology

In this study, we utilized a specialized simulation tool, as previously discussed, to estimate the maximum range of two distinct types of SAMs, which we have labeled SAM_1 and SAM_2. It is important to note that the actual models of these SAMs are not disclosed due to the sensitive nature of defense information. SAM_1 and SAM_2 are engineered for distinct operational requirements. Specifically, SAM_2 is optimally designed for long-range engagements and is especially effective against ballistic missiles, owing to its extended range capabilities. In contrast, SAM_1 excels in medium-range engagements, with high maneuverability and precision targeting offering distinct advantages. This differentiation uniquely suits each missile system to address varied threat profiles and operational contexts.

It is important to note that the maximum engagement range of a SAM, which is contingent on various situations between the missile launcher and the target, characterizes its EZ. For the models proposed in this study, we consider three main parameters as input:

(i) The elevation of the target relative to the launcher (x_1), spanning a range from -5,000 ft to 45,000 ft;

(ii) The speed of the target (x_2), measured in knots, encompassing values from 200 kt to 850 kt; and

(iii) The absolute value of the aspect angle of the target in relation to the launcher (x_3), varying within a spectrum from 0 degrees to 180 degrees.

When launched toward a target, our tool leverages these parameters to calculate a missile's maximum range (y). In this particular scenario, in each simulation run, the target is characterized as a passive aircraft, which maintains a state of non-reactivity during its flight—sustaining a constant speed and unwavering altitude throughout.

Initially, we generated 3,000 samples of the maximum range for each SAM type. Following their intervals, this was accomplished by applying Latin Hypercube Sampling (LHS) to the independent variables x_1, x_2, and x_3. Each resultant sample, referred to as an observed sample, comprises the values of the independent variables x_1, x_2, and x_3, along with a corresponding value of the dependent variable y.

Subsequently, we employed a simple polynomial regression to construct EZ models for SAM_1 and SAM_2 based on the generated sample sets. These models enable predictions of each SAM type's performance under varying conditions, as the input parameters specify.

We conducted a series of tests with the EZ models created and verified the occurrence of many estimation outliers in the results. We defined an estimation outlier as a value of the dependent variable y with a percentage error greater than 10% relative to its respective observed sample. The percentage error (e_{pi}) of a sample with index "i" is defined by:

$$e_{pi} = 100 \cdot \frac{|\hat{y}_i - y_i|}{y_i} \tag{6.1}$$

where: y_i is the observed value of the dependent variable of sample i, and $\hat{y}_i$ is the estimated value of the dependent variable relative to sample i.

Upon examining the results derived from the EZ models, we noted that most outliers occurred when the aspect angle absolute value was equal to or exceeded 144°. Consequently, we elected to generate additional maximum range samples for the two SAMs within the 144° to 180° range to reduce the number of outliers in the testing phase of the EZ models. To achieve this, we partitioned the domain of the aspect angle modulus into five distinct intervals or sectors: $[0°, 144°)$, $[144°, 153°)$, $[153°, 162°)$, $[162°, 171°)$, and $[171°, 180°)$. We then generated 5000 samples within each interval, utilizing LHS for this purpose. The experiments were executed using an Intel Xeon Gold 6230R CPU with 2.10GHz (4GHz of maximum turbo frequency) and 64 GB of RAM.

Finally, we created an EZ model for each sample set of the SAM_1 and SAM_2 through the application of methods based on Artificial Neural Network (ANN), Random Forest Regressor (RFR), and Polynomial Regression (PR). Additionally, we developed comprehensive EZ models that incorporated the entirety of the sample set—essentially, the 25,000

samples spanning the interval $[0°, 180°]$ for each SAM type.

When developing the EZ models for each sample set, we made the following considerations:

(a) We allocated 80% of the samples for training and validation purposes of the models;

(b) We employed the 5-fold cross-validation method for model training and validation; and

(c) The remaining 20% of the samples were reserved for testing the models.

In assessing the efficacy of the EZ models, our primary goal was to identify the model that exhibited the lowest mean percentage error. This measure would suggest a reduced incidence of outliers. We adopted a collection of metrics to examine these models, as detailed in the results section of our report. This in-depth analysis helped us determine whether using a singular EZ model for the entire interval of [0°,180°] would be more beneficial or if dividing models for each of the five sectors might yield better outcomes. Additionally, this approach allowed us to compare the models derived from previously mentioned supervised learning techniques: ANN, RFR, and PR.

Tables 6.1 and 6.2 offer an in-depth overview of the critical hyperparameters used in the ANN and RFR methodologies for constructing the EZ models, respectively. These configurations were meticulously chosen through exhaustive experimentation. We utilized the TensorFlow library (ABADI *et al.*, 2016) for the ANN methodology, with each hidden layer having a uniform number of units. For the RFR methodology, we employed the Scikit-Learn library (PEDREGOSA *et al.*, 2011a).

TABLE 6.1 – Detailed settings of key hyperparameters for the ANN model.

Hyperparameter	**Values**
Number of Hidden Layers	[2, 5, 10]
Number of Units in Each Hidden Layer	[32, 64, 128]
Batch Size	16
Patience Threshold for Early Stopping	10
Optimizer	Adam*
Activation Function in the Hidden Layers	Rectified Linear Unit (ReLU)

*Optimizer based on the method proposed by Kingma and Ba (2014)

Our study explored other machine learning techniques, such as the Support Vector Machine and AdaBoost. However, the results from these methods significantly underperformed compared to those from the ANN and RFR. As a result, we chose not to include these underperforming methods in this chapter to maintain conciseness and to highlight

TABLE 6.2 – Detailed settings of key hyperparameters for the RFR model.

Hyperparameter	**Values**
Number of Estimators	100
Maximum Depth of a Tree	None (Unrestricted)
Minimum Number of Samples to Split a Node	2
Minimum Number of Samples at a Leaf Node	1
Maximum Number of Features for Optimal Split	3
Bootstrap	True

the more promising approaches. We used the `polyFit` function from the R library to establish the PR model. Table 6.3 illustrates the hyperparameter configurations for this function. Additional parameters related to the ANN, RFR, and PR methodologies were left at their default settings, and we opted not to detail them further.

TABLE 6.3 – Detailed settings of key hyperparameters for the PR model.

Hyperparameter	**Values**
Maximum Degree for Polynomial Terms	[9, 10, 12, 13, 14, 15]
Maximum Degree of Interaction Terms	3

The next section describes the results of creating these EZ models from the 12 sample sets and comparing them.

6.6 Results

Each SAM system, specifically SAM_1 and SAM_2, incorporates five distinct sample sets, as previously mentioned. Additionally, we included a sixth sample set, which is functionally identical to the collective union of the original five sample sets, covering an interval range from 0° to 180°. This expanded dataset allowed us to construct 36 predictive models, evenly split with 18 dedicated to each SAM type.

The model development process utilized the aforementioned methodologies, resulting in an assortment of six models for each method, amounting to 18 models per SAM system.

We assessed these models using the following performance metrics:

- Coefficient of Determination (R^2);
- Root Mean Squared Error (RMSE) expressed in nautical miles (nm);
- Mean Absolute Percentage Error (MAPE); and
- Processing Time (PT) measured in seconds.

The evaluation process of these models consisted of training and testing phases. Subsequent subsections provide detailed information about these phases.

6.6.1 Training Phase

We analyzed and compared the performance of the training and validation phases of the EZ models by applying the 5-fold cross-validation method to 80% of their respective sample sets. We performed a comparative analysis among the EZ models created by the same method.

We verified that the PR method created EZ models that performed better for the sectors than for the whole interval $[0°, 180°]$, considering the SAM_1. The worst performance of PR was in the sector $[162°, 171°)$, presenting an RMSE value of 1.4429 nm ± 0.1765 nm and a MAPE value of 2.85% ± 0.23%. For the interval $[0°, 180°]$, PR showed an RMSE value of 2.0775 nm ± 0.0331 nm and a MAPE value of 8.54% ± 0.31%. Considering the SAM_2, we verified that the PR method created EZ models that also performed better for the sectors than for the whole interval $[0°, 180°]$, except the sector $[0°, 144°)$. PR presented an RMSE value of 3.5196 nm ± 0.1703 nm and a MAPE value of 14.05% ± 2.42% for this sector. Whereas for the whole interval, PR showed an RMSE value of 2.8757 nm ± 0.1688 nm and a MAPE value of 5.09% ± 0.75%.

Regarding the ANN models for the SAM_1, the model performance was notably superior in the $[162°, 171°)$ sector (5 hidden layers with 64 units), with the highest mean R^2 value of 0.9986 ± 0.0005 and the smallest mean RMSE of 0.5113 nm ± 0.0996 nm and MAPE as 1.71% ± 0.34%, suggesting reasonable predictions. The sector $[0°, 144°)$ (10 hidden layers with 32 units) also showed promising results, recording an R^2 of 0.9931 ± 0.0034 and a relatively low RMSE and MAPE, 0.2630 nm ± 0.0645 nm and 3.79% ± 1.15%, respectively. However, the performance significantly diminished for the more comprehensive sector range, especially in the $[0°, 180°]$ sector (2 hidden layers with 32 units), where it had the lowest R^2 of 0.9718 ± 0.0280 and the highest RMSE of 2.061 nm ± 1.157 nm and MAPE of 7.76% ± 1.95%, indicating a slight deterioration in the prediction process. Among the SAM_2 model sectors, the highest performing sector was the $[162°, 171°)$ range (5 hidden layers with 64 units), with a remarkable mean R^2 value of 0.9991 ± 0.0004 and a significantly lower RMSE of 0.2274 nm ± 0.0541 nm and MAPE of 0.24% ± 0.06%. The sector from $[171°, 180°)$ (5 hidden layers with 32 units) also showed strong performance with a mean R^2 value of 0.9975 ± 0.0016 and low RMSE and MAPE values. Conversely, the $[144°, 153°)$ (5 hidden layers with 64 units) exhibited a slightly lower mean R^2 value of 0.9644 ± 0.0274 and an RMSE of 2.0155 nm ± 0.7836 nm and MAPE of 1.64% ± 0.81%. The broadest sector, spanning $[0°, 180°)$ (2 hidden layers with 128 units), posted a mean R^2 of 0.9942 ± 0.0020 and an RMSE of 1.7500 nm ± 0.2713 nm,

indicating good overall performance. As a general trend, it can be inferred that models trained on narrower sector ranges tend to yield superior performance.

For SAM_1, we observed that RFR created EZ models that performed better for the whole interval, except for sectors $[0°, 144°)$ and $[144°, 153°)$. For the first sector, RFR presented an RMSE value of 0.2452 nm ± 0.0243 nm and a MAPE value of 3.22% ± 0.15%. For the second sector, RFR showed an RMSE value of 0.3462 nm ± 0.0687 nm and a MAPE value of 1.52% ± 0.60%. Whereas for the whole interval, RFR showed an RMSE value of 0.6197 nm ± 0.0668 nm and a MAPE value of 1.71% ± 0.14%. We also verified that, regarding MAPE values, the EZ model created for the whole interval performed better than the EZ model designed for the first interval. For SAM_2, we observed that RFR created EZ models that performed better for the sectors, except for sectors $[0°, 144°)$ and $[144°, 153°)$. For the first sector, RFR returned an RMSE value of 2.9785 nm ± 0.3371 nm and a MAPE of 6.75% ± 2.33%. For the second sector, RFR presented an RMSE value of 2.5176 nm ± 0.4481 nm and a MAPE value of 1.25% ± 0.24%. Whereas for the whole interval, RFR returned an RMSE value of 1.8977 nm ± 0.1755 nm and a MAPE value of 2.09% ± 0.53%. Regarding MAPE values, we also perceived that EZ models built for the second sector performed better than the EZ model of the whole interval.

6.6.2 Testing Phase

In this phase, we tested the EZ models using new sample sets not utilized during training. Subsequently, we analyzed and compared the performance of the EZ models using the remaining 20% of their respective sample sets. The metric results are presented in Table 6.4 for the SAM_1 and Table 6.5 for the SAM_2.

Regarding R^2, ANN and RFR consistently outperform PR across most sections in both SAM_1 and SAM_2. The highest R^2 observed for ANN was 0.9991 in SAM_2, and for RFR was 0.9997 in SAM_2. In contrast, PR peaked at 0.9996 in SAM_1 and 0.9999 in SAM_2 but frequently exhibited lower R^2 values than the other two models.

For RMSE, ANN performs better in SAM_1, while PR and RFR perform better in SAM_2.

The MAPE comparison reveals a mixed picture. PR shows the highest error in SAM_1 at 8.04% but performs well in SAM_2 with the lowest MAPE of 0.04%. ANN's performance varies, with the highest error observed at 14.21% in SAM_2 and the lowest at 0.09% in SAM_1. RFR consistently performs with lower errors, the best being 0.13% in SAM_2.

PT shows a clear trend: both ANN and RFR significantly outperform PR. RFR consistently shows the fastest processing time across all sections in both SAM_1 and SAM_2, indicating its computational efficiency. Through the results presented in Tables 4 and 5,

TABLE 6.4 – Evaluation of EZ models for SAM_1 during the testing phase using 20% of the sample sets.

Interval	**Metric**	**PR**	**ANN**	**RFR**
[0°, 144°)	R^2	0.9996	0.9979	0.9949
	RMSE	0.0646	0.1578	0.2278
	MAPE	0.92%	2.23%	3.32%
	PT	0.5070	0.2465	0.0255
[144°, 153°)	R^2	0.9939	0.9940	0.9919
	RMSE	0.2369	0.2288	0.2740
	MAPE	0.82%	1.21%	0.98%
	PT	1.8970	0.1926	0.0247
[153°, 162°)	R^2	0.9881	0.9944	0.9927
	RMSE	0.6578	0.4398	0.5333
	MAPE	1.62%	1.24%	1.25%
	PT	1.9120	0.2661	0.0251
[162°, 171°)	R^2	0.9873	0.9980	0.9947
	RMSE	1.4296	0.5523	0.9022
	MAPE	2.93%	0.09%	1.44%
	PT	1.9360	0.1812	0.0256
[171°, 180°]	R^2	0.9942	0.9976	0.9976
	RMSE	1.0533	0.6713	0.6779
	MAPE	2.41%	1.50%	0.96%
	PT	1.3440	0.1771	0.0255
[0°, 180°]	R^2	0.9764	0.9914	0.9389
	RMSE	2.0538	1.2869	3.4020
	MAPE	8.04%	7.27%	4.39%
	PT	8.6560	0.2763	0.0778

we found that the estimation times of all models are lower than the threshold of 0.01 seconds (L). The time to estimate the maximum engagement range of a single missile shot is equal to PT divided by the number of samples used in the testing phase of the respective model. For example, in Table 5, using the same hardware, the estimation time of the EZ model, created by the PR method for the interval $[0°, 180°]$, is $12.9340 \div 5000 = 0.0026$ seconds. This time limit, L, is significantly lower than the estimation time of the simulation tool. The simulation tool takes around 34 seconds to estimate the maximum engagement range of a single missile shot.

We also did a comparative analysis among the EZ models created by the same method. We observed that the EZ models produced by the PR method presented performances similar to those obtained by the EZ models developed in the training phase. The performance trend of the EZ models created by the ANN method is similar to that of the EZ models produced by the PR method. For SAM_1, the EZ models made by the ANN method performed better for the sectors than for the whole interval. For SAM_2, EZ models built by the ANN method also performed better for the sectors than for the entire interval, except for the sector $[0°, 144°]$. We noted different performances related to the

TABLE 6.5 – Evaluation of EZ models for SAM_2 during the testing phase using 20% of the sample sets.

Interval	**Metric**	**PR**	**ANN**	**RFR**
[0°, 144°)	R^2	0.9766	0.9800	0.9809
	RMSE	3.1722	3.1006	2.9824
	MAPE	14.51%	14.21%	6.08%
	PT	0.5170	0.1328	0.0267
[144°, 153°)	R^2	0.9583	0.9823	0.9675
	RMSE	2.2741	1.5595	1.9888
	MAPE	1.57%	0.98%	0.99%
	PT	2.9570	0.1836	0.0266
[153°, 162°)	R^2	0.9989	0.9778	0.9788
	RMSE	0.2432	1.1425	1.1029
	MAPE	0.17%	0.37%	0.30%
	PT	0.7480	0.2586	0.0262
[162°, 171°)	R^2	0.9999	0.9988	0.9995
	RMSE	0.0358	0.2695	0.1655
	MAPE	0.04%	0.29%	0.15%
	PT	0.5590	0.1843	0.0260
[171°, 180°]	R^2	0.9999	0.9991	0.9997
	RMSE	0.0418	0.2251	0.1401
	MAPE	0.04%	0.23%	0.13%
	PT	0.7450	0.1954	0.0255
[0°, 180°]	R^2	0.9849	0.9836	0.9993
	RMSE	2.8116	3.001	0.2078
	MAPE	4.63%	5.78%	0.21%
	PT	12.9340	0.2693	0.1027

RFR method. For SAM_1, EZ models built by the RFR method performed better for the sectors than for the whole interval. For SAM_2, EZ models built by the RFR method performed better for the whole interval, except sectors $[162°, 171°)$ and $[171°, 180°]$. However, the differences between the RMSE value returned by the EZ model of the whole interval and the RMSE values of these sectors are lower than 0.2 nm. Similarly, the differences between the MAPE value of the whole interval and the MAPE values of these sectors are lower than 0.1%. The magnitude of these differences is insignificant in the context of the problem addressed in this work. Therefore, we considered that the performance of the EZ model of the whole interval is equivalent to the performance of the EZ models of these two sectors.

Finally, we did a comparative analysis between the methods, i.e., we compared EZ models created by one technique with EZ models created by another.

We understood that the best composition of an SAM's EZ depends on the purpose of the simulation work. Therefore, we could represent an EZ of a SAM through the following:

- A Single EZ Model: This involves using one individual model to represent the SAM's

EZ;

- A Homogeneous Multimodel: This refers to a collection of EZ models created by the same method; and
- A Heterogeneous Multimodel: In contrast, this encompasses a set of EZ models produced by different methods.

If we only prioritized the most accurate EZ model, i.e., the performance in terms of the RMSE and MAPE values, we could represent the EZ for the SAM_1 as a heterogeneous multimodel. This multimodel would be formed by an EZ model built using the PR method for the sector $[0°, 144°)$ and EZ models created using the ANN method for the remaining sectors $[144°, 153°)$, $[153°, 162°)$, $[162°, 171°)$, and $[171°, 180°]$. Conversely, the EZ for the SAM_2 could be represented by a single EZ model built using the RFR method for the interval $[0°, 180°]$, with an RMSE value of 0.2078 and a MAPE value of 0.21%. The EZ models built using the PR method for the sectors $[162°, 171°)$ and $[171°, 180°]$ presented RMSE and MAPE values lower than 0.2078 and 0.21%, respectively. However, the differences between these RMSE and MAPE values and the mentioned values are lower than 0.2 nm and 0.1%. As we explained previously, these values are insignificant when compared to other methods. Thus, we considered that the EZ models built by the PR method for these two sectors are equivalent to the EZ model produced by the RFR method for the whole interval $[0°, 180°]$.

Suppose we prioritized the fastest EZ model, i.e., the performance in terms of the estimation time, with an RMSE limit of around 1.0 nm. In that scenario, we could represent the EZ for the SAM_1 as a homogeneous multimodel. Models constructed using the RFM method for the five sectors $[0°, 144°)$, $[144°, 153°)$, $[153°, 162°)$, $[162°, 171°)$, and $[171°, 180°]$ would form this multimodel. The EZ for the SAM_2 could be represented as a single EZ model for the whole interval $[0°, 180°]$.

In the context of this work, all three models exhibit strengths and weaknesses. In that case, PR tends to show the least prediction error but falls short in processing time efficiency. The RFR stands out for its processing time efficiency and presents a strong performance in error reduction. ANN also delivers a balanced performance, doing well in error reduction but lagging behind RFR in processing speed.

6.7 Outcomes

This study has illuminated the transformative potential of machine learning techniques in improving the efficiency and speed of SAM EZ simulations. Implementing machine learning algorithms can significantly mitigate the conventional computational

limitations that have consistently interfered with defense strategy planning. The integration of machine learning and custom-designed simulation tools within this study has yielded promising results, demonstrating the feasibility of predicting SAM EZs with precision and efficiency.

Our analysis of three machine learning models produced informative findings. PR showed the least prediction error, but its processing time efficiency could be improved. On the other hand, RFR brought its superior processing time efficiency and demonstrated commendable performance in error reduction. ANN also showed satisfactory performance in reducing error, although it lagged behind RFR in processing speed.

Despite the encouraging results, our study recognizes the inherent challenges in applying machine learning techniques in this context, such as the need for extensive training data, the risk of model overfitting, and the continual necessity for meticulous evaluation and refinement to maintain accuracy. Yet, these challenges provide pathways for future research.

Future work directions include further optimizing these machine-learning models and reducing processing time while maintaining or improving prediction accuracy. Moreover, researchers should explore incorporating other machine learning models not covered in this study. Considering the fast-paced evolution of algorithms, we anticipate novel methodologies may yield even more promising results in predicting SAM EZs. Also, a significant area of future research should include strategies to manage the challenges presented by data requirements and model overfitting, such as techniques for efficient data augmentation and model regularization.

In conclusion, our work affirms the potential of machine learning in advancing air defense strategies, offering a new level of dynamism and responsiveness to military decision-making. While the path toward full implementation of these techniques in the defense sector is not without its challenges, this study underscores the promise held by this field, paving the way for future innovative solutions.

6.8 Source Code

For further insight and access to the source code of this research, please visit our repository at `https://github.com/jpadantas/sam-ez`.

7 Probability of Kill for Air-to-Air Missiles

In modern aerial combat, the effectiveness of air-to-air missiles is important, with the Probability of Kill (P_{kill}) being a critical metric that quantifies the likelihood of a missile successfully neutralizing its target. An accurate estimate of P_{kill} is essential for operational planning and developing missiles. This chapter explores the methodologies and models used to estimate P_{kill}, including probabilistic approaches, statistical techniques, and machine learning algorithms. As indicated in Figure 1.2, this chapter primarily contributes to the "Weapon Systems" area within the proposed research framework.

The following work forms the basis for the development and insights presented in this chapter:

> **DANTAS, J. P. A.**; COSTA, A. N.; GERALDO, D.; MAXIMO, M. R.; YONEYAMA, T. PoKER: a probability of kill estimation rate model for air-to-air missiles using machine learning on stochastic targets. The Journal of Defense Modeling and Simulation, v. 0, n. 0, p. 15485129241309675, 2025. Available at: `https://doi.org/10.1177/15485129241309675`

7.1 Summary

This paper introduces PoKER, a novel probabilistic model engineered to optimize missile launch effectiveness in air-to-air scenarios, specifically within Beyond Visual Range (BVR) air combat. Unlike conventional Weapon Engagement Zone (WEZ) models that delineate zones based on static distances such as maximum, minimum, and no-escape ranges, PoKER applies machine learning to predict kill probabilities more accurately by integrating the stochastic behaviors of targets and missile miss distances. This model dynamically evaluates target behavior, greatly expanding the predictive capabilities of engagement analysis. By factoring in elements such as warhead lethality, target and shooter orientations, and the specific conditions of engagement, PoKER provides important in-

sights into engagement dynamics and quantifies success probabilities. Consequently, it can potentially be an important tool for BVR air combat pilots, improving operational decision-making within this specialized combat domain.

7.2 Introduction

Missile systems hold a critical position in contemporary military operations (LI *et al.*, 2020), providing an effective means to engage aerial targets over extensive distances, which is particularly significant in the context of Beyond Visual Range (BVR) air combat. BVR air combat refers to aerial engagements that occur at distances beyond the visual sight of the pilots, typically over tens of kilometers. In these scenarios, aircraft rely on advanced radar, sensors, and long-range missiles to detect, track, and engage opponents without the need for visual identification (DANTAS *et al.*, 2021a). The evolving landscape of BVR air combat, driven by recent advancements in detection and missile technology, underscores the necessity for sophisticated Weapon Engagement Zone (WEZ) models (LI *et al.*, 2022c). These models, essential for evaluating the operational efficacy of air-to-air missiles, bring about complex computations of maximum and minimum engagement ranges alongside the identification of No Escape Zones (NEZ) (DANTAS *et al.*, 2021b). Notably, these computations are usually dynamically executed in real-time during flight, factoring in the orientations of both the target and the shooter (HERRMANN, 1996).

WEZs are not simply geometric boundaries determined by orientations and positions; they also take into account the dynamics and maneuverability of both the target and shooter, along with their weapons systems (DANTAS *et al.*, 2021a). The insights derived from WEZ calculations may be relayed to pilots through heads-up or heads-down displays (YOON *et al.*, 2010), which may empower them to make more informed decisions regarding missile launches with increased precision and confidence (LING *et al.*, 2019).

The definition of WEZ can incorporate concepts such as long and short limits or maximum and minimum ranges, whose interpretations might slightly vary between countries (HOORN, 2019). To exemplify, we present two common nomenclatures, as depicted in Figure 7.1, to evidence their nuances, mostly with respect to enemy maneuverability. Some armed forces use Long Limit 0 (LL_0) to delineate the furthest kinematic reach of a missile, where it uses up all its energy just to activate its proximity fuse upon reaching the target. Similarly, Maximum Range ($R_{max,1}$) indicates the maximum distance a missile can effectively engage a passive target, with its success probability decreasing if the target maneuvers to increase flight distance. Long Limit 1 (LL_1) specifies the range within which a missile can effectively counteract a target's ultimate evasion efforts with maximum performance. Alternatively, $R_{max,2}$ introduces the beginning of the NEZ, which

represents a range within which the missile is expected to hit, barring specific defensive actions by the target, though escape is still possible through engine maneuvers or missile countermeasures. Lastly, the Short Limit $(LC)/R_{min}$ marks the closest distance at which a missile's fuze can arm.

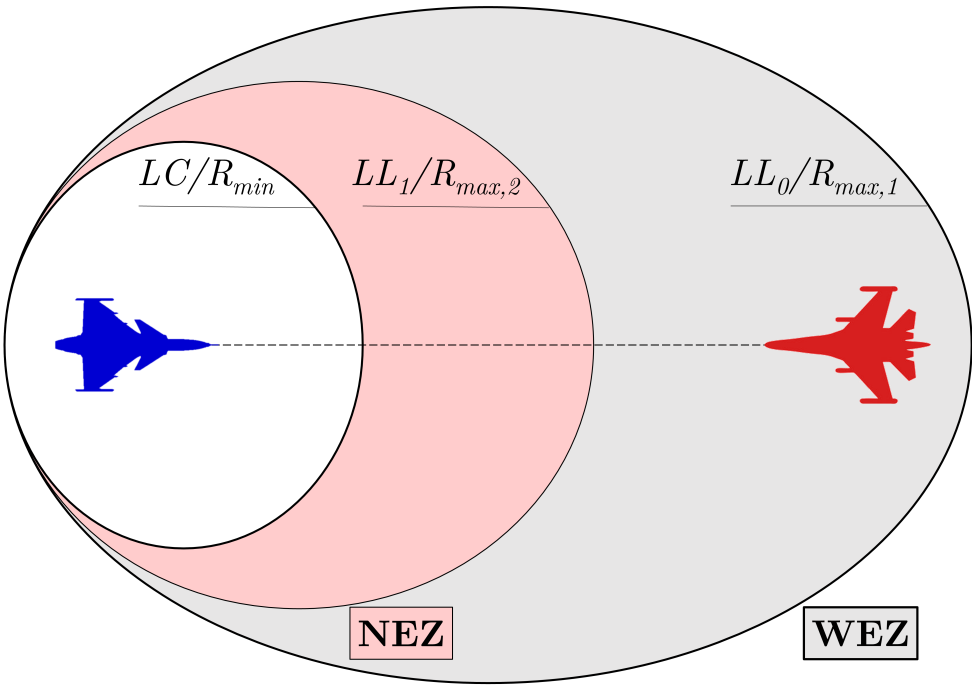


FIGURE 7.1 – Diagram illustrating aerial combat WEZ, detailing ranges for missile engagement from maximum kinematic reach to minimum arming distance, including the airspace where the target has limited evasion options.

Even though these concepts provide a general idea of where to engage a target, depending on rules of engagement and shot philosophies (DANTAS *et al.*, 2022b; KUROSWISKI *et al.*, 2023), these regions may still be quite extreme, not accounting for what happens in between them, which makes it difficult for pilots to choose the best moment to launch a missile in order to maximize the chances of success. This is aggravated by the fact that the opposing aircraft may maneuver unpredictably. Amid these complex considerations, it becomes essential to conduct detailed simulations to improve operational readiness and tactical accuracy, analyzing missile efficacy in diverse engagement scenarios, which include different adversarial reactions to missile threats. In this research, we introduce a novel WEZ model tailored to improve the assessment of missile performance after launch, specifically in the case of an Active Radar Homing (ARH) weapon system utilized in BVR air combat. Leveraging insights from prior work, we present an R-based simulator based on a 5 Degrees of Freedom (5DOF) model (DEPARTMENT OF DEFENSE, 1995).

ARH, or Active Radar Homing, refers to a guidance method used in missile systems,

where the missile itself is equipped with a radar seeker that actively emits radar signals to detect and track its target independently. This type of homing provides the missile with greater autonomy after launch, reducing the need for continuous guidance from the launching platform and enabling effective engagement of targets in BVR air combat scenarios.

Our simulator encapsulates the complexities of an ARH missile, including its radar seeker functionality and autonomous target-tracking capability, enabling rigorous analysis and prediction of its performance across various operational parameters. By exploring the nuances of missile-target interactions with an active radar seeker, our simulator provides valuable insights into BVR air combat dynamics, aiding in the formulation of strategic and tactical decisions.

Furthermore, we explore the domain of warhead lethality, offering insights into the calculation of the proximity-fuzed warhead's conditional probability of target destruction, a critical component in assessing missile effectiveness. Additionally, we focus on generating and preprocessing simulation data, proposing a supervised machine learning methodology to develop regression models capable of predicting missile kill probabilities. We carefully evaluate various algorithms' efficacy and computational efficiency, illuminating their suitability for real-world applications.

The main contribution of this work is as follows:

- **Development of the Probability of Kill Estimation Rate (PoKER) model**: Designed to provide enhanced situational awareness in dynamic air-to-air combat scenarios, particularly in BVR air combat settings.

The specific contributions that support the development and application of the PoKER model are:

- **Complementation of WEZ with kill probability estimation**: Addressing the gap in traditional weapon engagement analysis by providing an estimation of the probability of kill when launching air-to-air missiles, incorporating real-time conditions of both the launcher and the target for a more comprehensive understanding of engagement success.
- **Modeling target decisions**: Integration of potential adversary maneuvers using high-fidelity models of aircraft and missiles, allowing for flexibility rather than relying on fixed target profiles and predefined launcher strategies.
- **Simulation and machine learning integration**: Utilization of well-designed simulation tools and machine learning to deepen the understanding of missile engagement dynamics.

- **Warhead lethality model**: Inclusion of a warhead lethality model that estimates the probability of kill (P_{kill}) based on missile miss distance and a dynamic model of the opponent's behavior, substantially improving tactical decision-making and enhancing operational capabilities in BVR air combat.

In summary, the primary focus of the PoKER model is to enhance situational awareness in dynamic BVR engagements, marking an initial step towards the development of probabilistic models for weapon engagement analysis. This work offers a novel methodological contribution, advancing beyond traditional WEZ analysis by incorporating real-time probabilistic assessments and accounting for the dynamic interactions between the launcher and the target. Furthermore, the insights gained from this model could be leveraged in the future to support decision-making processes, enabling more informed tactical decisions in complex air combat scenarios.

The remainder of this document is organized as follows. Section 7.3 provides an overview of the literature pertinent to our study and introduces critical terminology. Section 7.4 delineates our approach, outlining each of its steps and the employed tools and techniques. Section 7.5 section showcases and discusses the findings derived from the application of our methodology. Section 7.6 summarizes our approach and its outcomes while also highlighting potential avenues for future research.

7.3 Related Work

The success of BVR air combat heavily relies on the seamless integration of offensive and defensive tactics supported by accurate and timely information. Consequently, numerous research efforts have focused on enhancing tactical decision-making through advanced air combat simulations, aiming to improve the effectiveness of engagements in dynamic scenarios (DANTAS *et al.*, 2023; DANTAS *et al.*, 2022; REINISCH *et al.*, 2022; UMMAH *et al.*, 2019; KANG *et al.*, 2019). These studies explore various aspects of BVR air combat, including weapon engagement, evasion strategies, and cooperative tactics, all of which contribute to a more comprehensive understanding of air combat dynamics.

7.3.1 Engagement and Escape Zone Strategies

The current study models predominantly integrate missile information to define Dynamic Escape Zones (DEZ), which refers to the region where the aircraft's survivability can be guaranteed through kinetic evasion maneuvers (YAGCI; NIKBAY, 2022), and WEZ, which are also referred to as Dynamic Launch Zones (KATUKURI, 2023), Missile Attack Zones (QIAN *et al.*, 2022), or Launch Acceptability Regions (OZDEMIR *et al.*, 2021). The

primary aim of WEZ models is to aid in decision-making processes concerning weapon launch and guidance support – through uplink updates – by outlining the boundaries and operational effectiveness of air-to-air missiles during target engagement. Two main methods are traditionally used to calculate WEZ for various missile configurations. The first method conducts ongoing flyout simulations of the missile model during the engagement period. The second, more widely used method performs simulated flyouts in an offline setting, creating a lookup table that incorporates the pre-simulated data. Upon employing the WEZ model, interpolation is applied to this data to produce outcomes that are not directly available in the table (BIRKMIRE, 2011).

Several techniques have been developed for both collecting the predefined data and for its subsequent interpolation. However, notable advancements have also been made in fields such as post-launch WEZ analysis, cooperative modeling of WEZ, and algorithms for WEZ estimation (HUI *et al.*, 2015; SHI *et al.*, 2018; YUE *et al.*, 2020; DANTAS *et al.*, 2021b). These studies tailor WEZ concepts to specific combat scenarios and maneuver types, often integrating them with three-degree-of-freedom (3-DOF) or six-degree-of-freedom (6-DOF) models (VENKATESAN; SINHA, 2015). Solution methods for WEZ calculations range from offline simulation algorithms to polynomial, interpolation, and neural network fitting algorithms (DANTAS *et al.*, 2023b).

On the other hand, the DEZ definition aims to encompass essential elements of tactical decision-making from the perspective of missile avoidance. Techniques such as differential game theory and principles like Minimum Evasive Range (KUNG; CHIANG, 2015) or NEZ (SCUKINS *et al.*, 2023) play a central role in differentiating between aggressive and defensive tactics, as well as in identifying the best timing for evasive actions (ALKAHER; MOSHAIOV, 2015; RYU *et al.*, 2018). This can be done either by analyzing missile miss distances in one-on-one engagements or by considering multiple incoming threats (SCUKINS *et al.*, 2024b).

7.3.2 Cooperative Engagement Models

While most of the earlier works focused on duel engagements, there has been a shift towards modeling cooperative engagement among multiple aircraft, particularly in scenarios involving Unmanned Combat Aerial Vehicles (UCAVs) (LI *et al.*, 2020). Cooperative engagement models extend WEZ and DEZ strategies by introducing coordination between multiple aircraft to enhance engagement effectiveness and survivability. These models aim to optimize WEZ and NEZ calculations using cooperative strategies, leveraging algorithmic approaches to increase effectiveness in group engagements (LI *et al.*, 2022c). Furthermore, optimization methodologies for attack positioning and cooperative tactics have been proposed, incorporating offensive and defensive factors for assessing the best

maneuvers (YANG *et al.*, 2014; GAO *et al.*, 2022; GAO *et al.*, 2023).

7.3.3 Probabilistic Modeling in BVR Air Combat

In recent years, there has been growing interest in incorporating probabilistic elements into BVR combat modeling, moving beyond traditional deterministic approaches. Some works have used Bayesian Networks to create probabilistic graphical models representing variables and their conditional dependencies through a directed acyclic graph (SUN *et al.*, 2020b). BVR combat can be divided into four phases: position occupy maneuver, launch maneuver, midcourse guidance, and terminal guidance. Utilizing Dynamic Bayesian Networks (DBN), these studies encapsulate the probabilistic relationships and uncertainties in each phase, predicting the likelihood of a successful missile hit by considering aircraft positions, velocities, maneuvers, missile specifications, and environmental conditions. Integrating expert knowledge and data from air combat simulations creates a temporal representation of the combat scenario. Through DBN analysis, key determinants of missile success probability are identified, and the Dynamic Attack Zone (DAZ) is delineated – representing the region where a missile launch is likely to result in a hit. This probabilistic framework facilitates simulation and tactical analysis, providing insights for optimizing missile deployment and supporting decision-making under uncertainty in BVR engagements.

7.3.4 Limitations in Existing Work

While the existing literature explores various dimensions of WEZ determination, a noticeable gap exists regarding incorporating the target's maneuvering capabilities into WEZ creation. Even some works that include probabilities through Bayesian Networks do not show clear variability in the target maneuvering, as the target makes a turn as soon as it detects the threat of the incoming missile (SUN *et al.*, 2020b). Our approach distinguishes itself by incorporating the concept of maneuver probability into the process of determining the WEZ, a feature absent in the studies we reviewed. This innovative method extends beyond only considering the initial dynamics and physical attributes of both the missile and the aircraft. It proactively forecasts the adversary's potential maneuvers, transforming the WEZ from a simple geometric entity into a dynamic probability distribution that reflects the likelihood of the missile's successful engagement with the target. Our model furthermore enhances this by incorporating the target's behavior in a probabilistic manner and providing a probability of success through studying warhead lethality.

7.4 Methodology

This section examines the methodology encompassing the missile launch simulator, the acquisition of simulation data, and the preprocessing techniques applied to ensure high-quality datasets. We thoroughly describe the input variables and the steps taken to analyze and undersample the data. We then elucidate the methods used to calculate warhead lethality, which is integral to determining the missile's probability of kill based on its miss distance. The development, training, and evaluation of supervised machine learning models are discussed in depth, concluding with an assessment of their performance. Finally, we explore the estimation of turn degree and delay, key factors in computing the probability of hitting a target with the missile.

7.4.1 Missile Launch Simulator

In this research, we have developed an enhanced version of the missile launch simulator, primarily based on our previous work (DANTAS *et al.*, 2021b). Our simulator, implemented in R, models the behavior of a Fox 3 missile, following a 5DOF framework as outlined in the Missile Handbook (DEPARTMENT OF DEFENSE, 1995). This type of missile, as defined in military terminology (AIR LAND SEA APPLICATION CENTER, 2020), is an ARH (Active Radar Homing) missile, meaning it is equipped with an autonomous seeker capable of tracking a target after activation at a specific range (DANTAS *et al.*, 2021b). Through this simulator, we are able to analyze and predict missile behavior and performance under a variety of operational conditions, providing valuable insights into missile-target interactions in BVR air combat. Due to confidentiality and sovereignty concerns, the missile model is under restricted access.

7.4.1.1 Guidance and Navigation

The model's core feature is its guidance system, designed to emulate proportional navigation. This ensures the missile maintains an optimal trajectory relative to the moving target, adjusting its course to align precisely according to its guidance law. Furthermore, the simulator incorporates the possibility of a loft maneuver - a sharp, upward trajectory immediately after launch. This maneuver is critical in extending the missile's range and enhancing its probability of intercepting distant targets (DANTAS *et al.*, 2021b).

7.4.1.2 Target Interaction

The simulation considers various target behaviors, including both stationary and maneuvering targets. One key aspect of the model is its ability to simulate high-performance maneuvers of the target, such as the +5 G maneuver depicted in Figure 7.2. The timing of these maneuvers can be varied, allowing for a comprehensive analysis of missile performance under different engagement scenarios. The data in Figure 7.3 offers insights into the target's behavior, illustrating sample metrics from the simulation, including velocity, acceleration, altitude, heading, and pitch, and showing how these maneuvers impact the engagement.

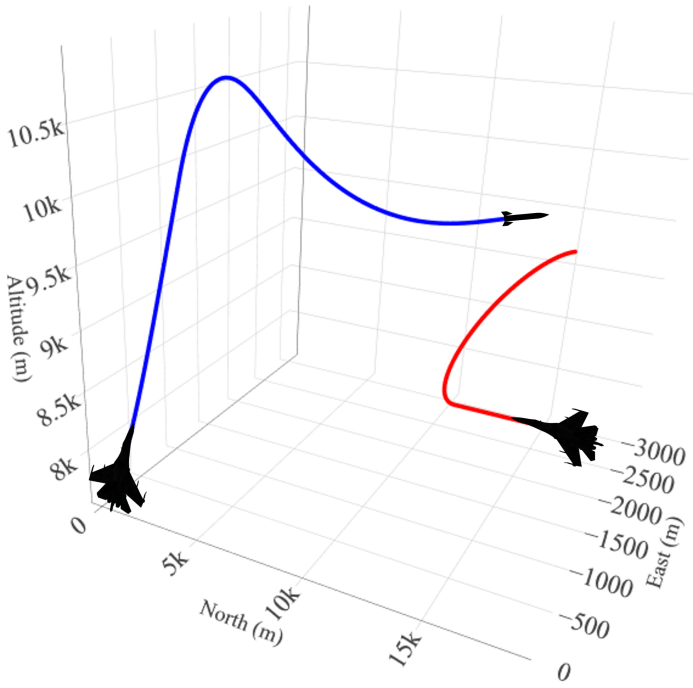


FIGURE 7.2 – Missile engaging a target performing a +5 G maneuver, illustrating the missile's adaptive trajectory in response to the target's high-G evasive actions.

7.4.1.3 Missile Trajectory Metrics

The simulation generates an analysis of missile trajectory metrics. One notable aspect is the missile's mass, which shows a consistent decrease almost linearly during the boost phase. This pattern aligns with the expected behavior of a dual-thrust rocket motor, a critical feature in missile design (DEPARTMENT OF DEFENSE, 1995; RAZA; LIANG, 2012). Additionally, the simulation reveals variations in the missile's angle of attack and pitch angle. These variations are especially pronounced during complex maneuvers such as the loft maneuver (MA *et al.*, 2015). The data depicted in Figure 7.4 highlights these dynamics, providing a view of the missile's performance metrics throughout the simulation.

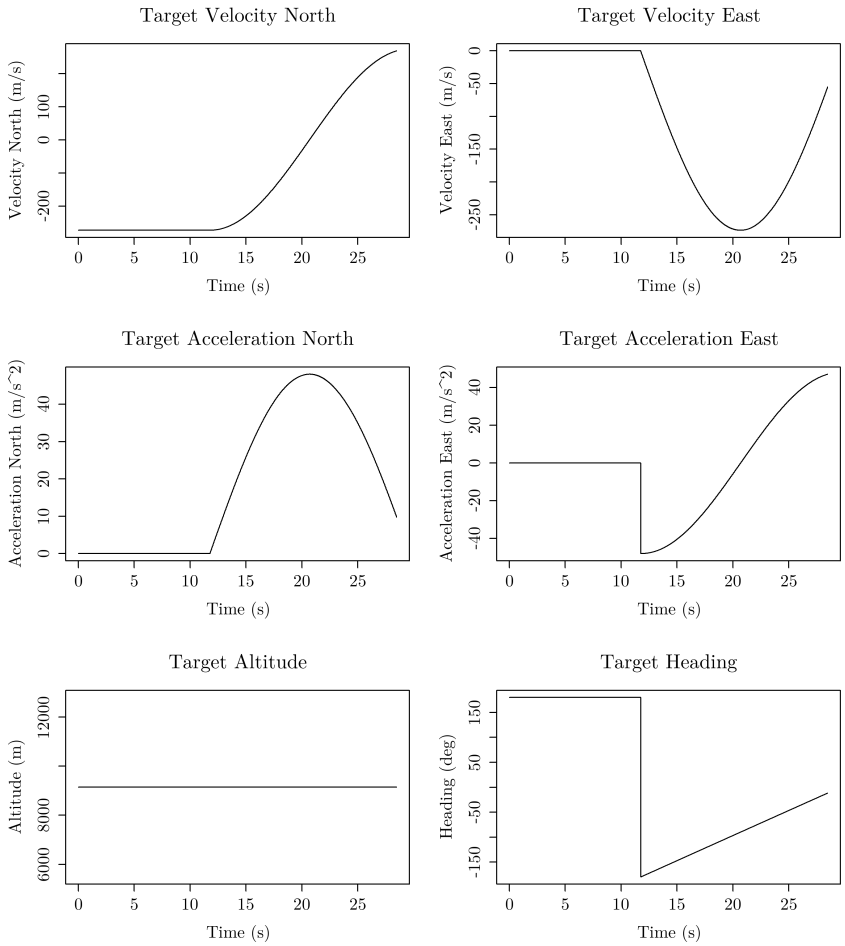


FIGURE 7.3 – Metrics from the simulation include target velocity (North, East), acceleration (North, East), altitude, and heading over time. Note that in this case, the target maneuver was considered to be leveled, and the target kept a constant altitude while evading the missile, resulting in a flat altitude profile. There is also a delay time in the pilot's reaction, causing all of the charts to present a flat profile for the first few seconds.

7.4.1.4 Flight Dynamics

In the domain of flight dynamics, the simulation includes detailed heading adjustments. These adjustments are fine-tuned to simulate responses to potential evasion tactics employed by a target, thereby evaluating the missile's adaptability in dynamic combat environments. The model also tracks accelerations along the East and North axes, adhering to the North-East-Down (NED) coordinate system. This aspect of the simulation is instrumental in assessing the missile's response to both loft maneuvers and evasive actions by the target. Another key aspect studied is the velocity profile of the missile. This profile demonstrates a characteristic increase during the boost phase followed by a gradual decrease during the sustain phase (KUMAR *et al.*, 2010). Figure 7.4 also illustrates these aspects, providing additional insights into the missile's adaptability and performance in various dynamic scenarios.

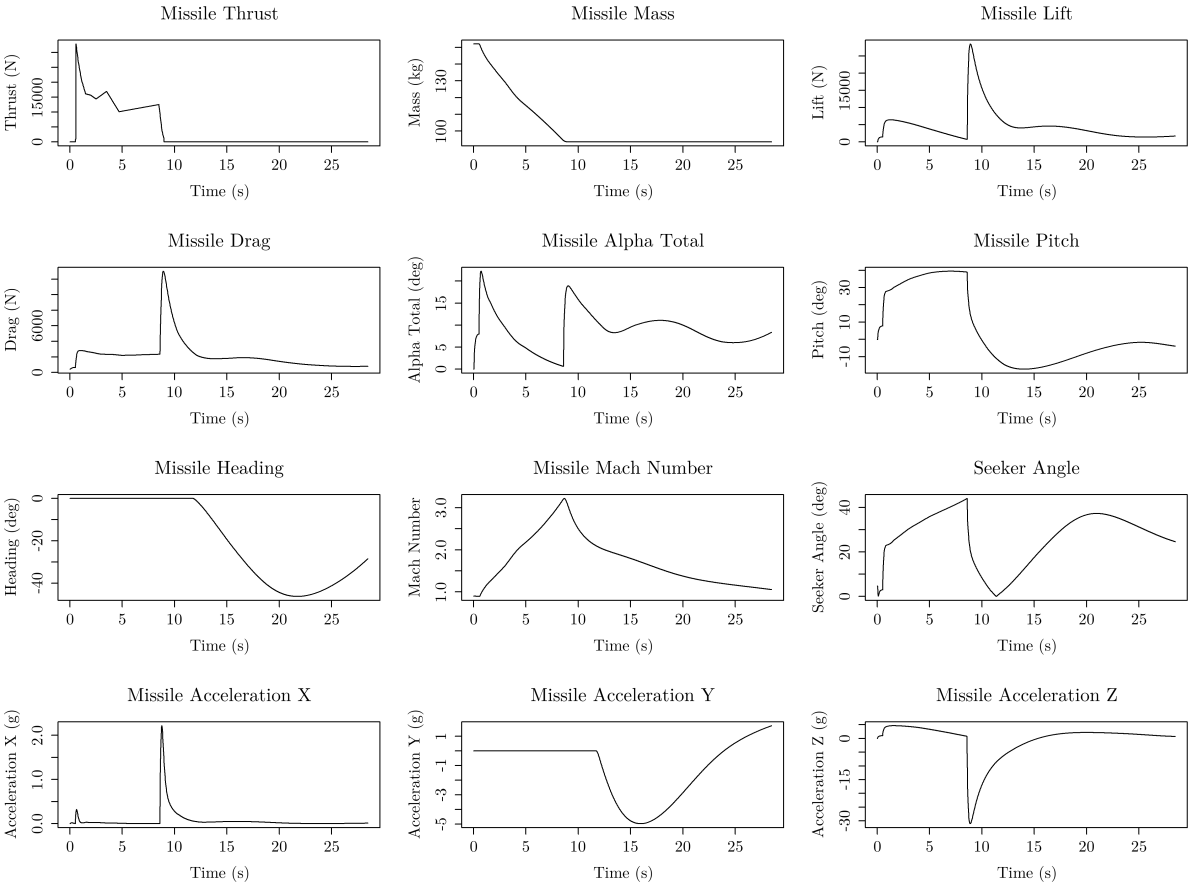


FIGURE 7.4 – Simulation metrics for the missile, showing thrust, mass, lift, drag, total alpha, pitch, heading, Mach number, seeker angle, and acceleration (X, Y, Z) over time.

7.4.1.5 Seeker Behavior

The final aspect covered by the simulation is the missile's seeker behavior. The seeker angle is particularly important, as it directly correlates with the missile's guidance system performance. This angle reflects the adjustments made by the missile's seeker in following a proportional navigation path. Analyzing deviations, especially from the off-boresight angle, provides critical insights into the missile's effectiveness, especially in scenarios where the target employs advanced evasive maneuvers (PARK *et al.*, 2016).

7.4.1.6 Visualization and Debugging Platform

The Aerospace Simulation Environment, or *Ambiente de Simulação Aeroespacial (ASA)* in Portuguese, serves as a critical resource for the Brazilian Air Force, facilitating enhanced missile launch visualization and debugging processes (DANTAS *et al.*, 2022a). This comprehensive C++ simulation framework, tailor-made for Brazilian military applications, supports detailed examinations and visualizations of various military operations, including missile launches. More than a tool for basic training, ASA is key for executing complex,

integrated combat scenarios and is essential in testing and validating tactical strategies. It provides a platform where analysts and engineers can perform comprehensive assessments and refine aerospace mechanisms and missions.

The missile model used in this research follows the same structural framework as the model used in ASA. This demonstrates the level of fidelity aimed for in this work, ensuring consistency with established simulation models in ASA. The ASA platform plays an essential role in validating and debugging this missile model, as it incorporates other elements and behaviors into the scenario, providing a more comprehensive environment for testing and refinement.

The advanced functionalities of ASA lead to improved accuracy in mission analysis and bolster reliability in operational execution (DANTAS *et al.*, 2023a). Looking forward, the insights gained from WEZ analyses in this research could be integrated into more complex simulations within ASA, where agents could make decisions based on a variety of available models, thus enhancing the overall operational readiness and tactical efficiency of the aerospace forces.

7.4.2 Simulation Data

Initially, to generate the simulation data, we created simulator input files using Latin Hypercube Sampling (LHS), an efficient statistical method for designing experiments. This method divides the space into a prespecified number of sections and randomly samples one point from each (MCKAY *et al.*, 1979; HELTON; DAVIS, 2003). Unlike factorial designs, LHS offers a superior alternative for populating the sample space more efficiently than purely random methods (HUSSLAGE *et al.*, 2011; DANTAS *et al.*, 2022b). We employed LHS using the AsaPy Library (DANTAS *et al.*, 2024), a custom-made Python library associated with ASA, specifically designed to optimize the analysis of simulation data.

Both the shooter's altitude (`alt_sht`) and velocity (`vel_sht`) significantly impact the energy transferred to the launched missile (HERRMANN, 1996). The launch altitude (`alt_sht`) influences the missile's drag, thus affecting flight performance (DANTAS *et al.*, 2021b). The shooter's pitch (`pit_sht`), which is defined with respect to the North-East plane, is important for determining the missile's initial angle post-launch, aiding in the loft maneuver, which is the missile's initial maneuver after launch. The target's velocity (`vel_tgt`) is significant, as it determines the defensibility of the launched missile. Two aspects are fundamental in terms of the angles between the shooter and the target: heading and off-boresight (Figure 7.5). The target's heading (`hdg_tgt`) indicates the direction taken to evade the missile with respect to the North, while the off-boresight angle (`rgt_tgt`), defined as the angle between the shooter's nose and the target, along with the target's heading, reveals whether the shooter aircraft is approaching the target and thus

the missile (VERGEZ; MCCLENDON, 1982; VERGEZ, 1998). Additionally, the distance between the shooter and the target (`dist`) is critical as it determines the missile's flight time and significantly influences the hit probability. The target's response time (`delay`) represents the time before initiating a defensive maneuver. A longer response time may force the target to execute more aggressive maneuvers, which are characterized by a higher turn degree (`turn_dg`) and greater acceleration (`load_factor`). All measurements are taken with respect to the center of mass of either the aircraft or the missile.

The target's response time (`delay`) represents the time before initiating a defensive maneuver. Depending on how long the target delays the response, it may need to perform more aggressive maneuvers, characterized by a higher turn degree (`turn_dg`) and greater acceleration (`load_factor`). All measurements are taken with respect to the center of mass of either aircraft or missile.

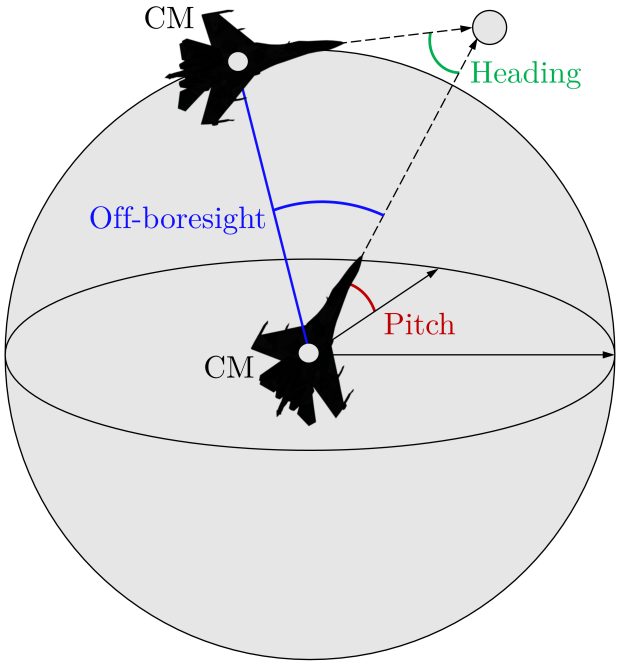


FIGURE 7.5 – Diagram illustrating the heading, off-boresight, and pitch angles between two aircraft in aerial maneuvering with respect to their centers of mass (CM).

Table 7.1 illustrates the range for each parameter considered during the generation of samples. These parameters were established based on the expertise of military fighter pilots, serving as Subject Matter Experts (SMEs) in military tactics, who identified critical values pertinent to the scenario. This approach aligns with methodologies previously reported in similar studies focusing on input variables within this domain (DANTAS *et al.*, 2021b; BIRKMIRE, 2011).

Since the LHS method allows the user to specify the desired sample points, we created 10 million different input files. The authors used this work as an opportunity to thoroughly test the simulator and work with a high-fidelity model, considering the large number of variations in input parameters and the potential for non-realistic combinations to arise.

TABLE 7.1 – Input simulation data with the respective intervals.

Parameter	Min	Max	Unit
Shooter altitude (`alt_sht`)	1,000	45,000	ft
Shooter velocity (`vel_sht`)	300	700	kt
Shooter pitch (`pit_sht`)	-30	30	deg
Target altitude (`alt_tgt`)	1,000	45,000	ft
Target velocity (`vel_tgt`)	300	700	kt
Target heading (`hdg_tgt`)	-180	180	deg
Target off-boresight (`rgt_tgt`)	-60	60	deg
Distance shooter-target (`dist`)	5	45	NM
Target response time (`delay`)	15	30	s
Target turn degree (`turn_dg`)	0	180	deg
Target acceleration (`load_factor`)	1.5	5	G (m/s^2)

Additionally, previous works by the authors using the same simulator provided indications of the necessity of a large volume of data to achieve reliable and meaningful results. These 10 million input files generated an equal number of missile launch simulations, which were executed using an Intel Xeon Silver 4210R CPU with 2.40GHz and 128 GB of RAM. It took approximately 40 days to execute all the simulation runs. Each run generated an output file containing the missile's miss distance (`miss_dist`), measured in meters by the simulator, under the respective input conditions. The miss distance is measured at the closest point of approach to the target aircraft and represents the minimum separation distance between the missile and its target (BALL, 2003). The large number of simulations also allows for an effective downsampling process, as described in the subsection 7.4.3, ensuring that a representative subset can be selected while maintaining the fidelity of the original data.

While the LHS method ensures a well-distributed sample across the input parameter space, the simulation data generated in this study is subject to certain limitations due to assumptions and simplifications inherent in the model. One such limitation is the idealized behavior of both the shooter and target, which may not fully capture the complexity of real-world engagements, such as pilot decision-making under uncertainty or environmental factors like varying weather conditions. Additionally, some parameters, such as missile dynamics, were modeled based on a consistent structure, but without accounting for potential variations in performance due to system degradation or other operational constraints.

The simplifications in target response and maneuvering behavior might also limit the accuracy of the model when applied to diverse combat scenarios. Moreover, the computational resources limited us to certain ranges and resolutions of parameters, which might not entirely cover all possible operational scenarios. These factors must be taken into account when interpreting the results, as they could impact the generalizability of the findings to different environments and conditions. Despite these limitations, the insights gained from this analysis provide a valuable foundation, which could be further tested and validated using more complex scenarios and comprehensive simulation environments.

7.4.3 Preprocessing

Concerning the preprocessing step, we initially performed an Exploratory Data Analysis (EDA) to identify the dataset's general behaviors and obtain an introductory understanding of the data. This understanding allowed us to formulate hypotheses and assess the need for new data collections. The methods employed for this primary analysis included descriptive statistics, histograms, box plots, correlation, and outlier analysis.

In our study, we employed histograms and box plots to understand the output variable's distribution and behavior across simulations and identify initial outliers in the samples. A essential aspect of our analysis was the investigation of correlations among variables to detect multicollinearity, which can significantly impair the efficacy of supervised machine learning algorithms (ALIN, 2010). Notably, we observed a substantial correlation between the Target turn degree (`turn_dg`) and Target acceleration (`load_factor`). This finding is particularly relevant, suggesting the redundancy of one of these variables. Therefore, we propose eliminating `load_factor` from the set of input variables to optimize the performance of our future machine learning models.

Furthermore, we employed a downsampling technique on the initial LHS design to examine potential outliers in the data collected. Identifying outliers and inadequate data in a dataset is arguably one of the most challenging parts of the preprocessing stage, and it is always a topic to be explored cautiously (KAZIL; JARMUL, 2016). Based on subject matter expert knowledge, we defined the variable intervals but did not control the combination of values generated for each parameter. As a result, we could produce some improbable input variables. For instance, an aircraft at 1,000 ft launching a missile at a target at 45,000 ft is extremely rare from an operational perspective since a pilot would most likely increase altitude before shooting (DANTAS *et al.*, 2021b). We manually removed these disfavored samples, such as the one presented, to prevent confusion in the model. For more details, check the code associated with the EDA methods *.

7.4.4 Warhead Lethality

The lethality of a proximity-fuzed warhead, represented by the probability of kill (P_{kill}), depends on the conditions at the time of detonation and the relative position between the warhead and the target aircraft. The P_{kill} function estimates the likelihood that an aircraft is destroyed given the detonation of the warhead (BALL, 2003). This probability is influenced by several factors, including fragment velocity, density of fragment dispersion, and the miss distance.

Figure 7.6 illustrates how increasing the detonation distance reduces the effectiveness

*EDA code: `https://github.com/jpadantas/poker/tree/main/eda`

of the warhead. The ideal lethal radius represents the optimal distance for maximum lethality, while the declining effectiveness curve shows how the kill probability decreases as the detonation distance deviates from this optimal value (BALL, 2003).

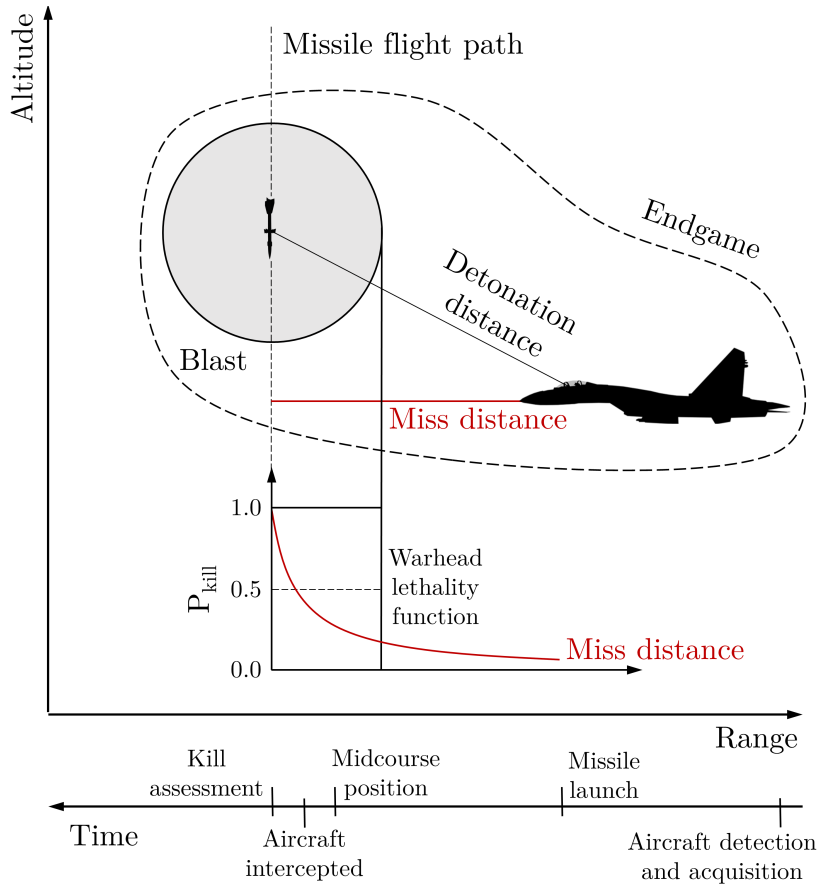


FIGURE 7.6 – Impact of Detonation Distance on Warhead Lethality. The figure shows the relationship between detonation distance and P_{kill} for a proximity-fuzed warhead. The lethal radius represents the optimal distance for maximum lethality, contrasted with the decreasing kill probability as the distance increases.

7.4.4.1 Factors Affecting Warhead Lethality

The lethality of a proximity-fuzed warhead is determined by various factors that influence how effectively the warhead fragments impact the target. Key factors include:

- **Miss Distance**: The proximity of the warhead at detonation to the target is the primary factor affecting the probability of kill. A smaller miss distance generally leads to a higher likelihood of target destruction, while a larger miss distance reduces effectiveness.
- **Fragment Distribution and Density**: The arrangement and density of the fragments around the explosive affect how many fragments are likely to impact the target. In

this study, the pre-formed fragments are arranged in a brickwork pattern, which optimizes coverage and ensures a consistent fragment distribution.

- **Fragment Initial Velocity**: The initial velocity of the fragments post-detonation, which depends on the Gurney constant and the explosive characteristics, determines how far the fragments travel and their ability to penetrate the target structure.
- **Warhead and Explosive Properties**: The material composition, dimensions, and explosive characteristics (e.g., detonation velocity, explosive mass) collectively impact the lethality.

7.4.4.2 Modeling the Physical Properties of the Warhead

To calculate P_{kill}, it is necessary to accurately model the warhead's physical properties. Table 7.2 presents the structural, fragment, and explosive properties of the warhead used in this study.

TABLE 7.2 – Missile's warhead data including structural, fragment, and explosive properties.

Variable	Value	Unit
Explosive Data		
Explosive Type	PBX-9404	-
Explosive Density (ρ_e)	1710	kg/m^3
Explosive Mass (M_e)	8.94	kg
Detonation Velocity	9500	m/s
Gurney Constant (G)	2895	m/s
Initiation Point (X_2)	100	mm
Charge-to-Mass Ratio (C/M)	0.65	-
Fragment Initial Velocity (V_0)	2028	m/s
Material Data		
Material	Mild Steel	-
Density (ρ)	7850	kg/m^3
Geometric Data		
External Diameter	178	mm
Skin Thickness	2	mm
Structural Diameter (D_s)	154	mm
Internal Diameter (D_i)	144	mm
Structural Thickness (T_s)	5	mm
Effective Length (L_e)	321	mm
Structural Mass (M_s)	5.90	kg
Pre-formed Fragment Data		
Type	Sphere	-
Pattern	Brickwork	-
Diameter or Distance ($\overline{AO}$)	10	mm
Distance ($\overline{OB}$)	8.66	mm
Sphere Mass (m_{sph})	0.0041	kg
Number of Spheres		
Total Number (N_{sph})	1909	-
Total Mass of Spheres (M_{sph})	7.85	kg

The properties presented in Table 7.2 form the foundation for understanding the warhead's behavior during detonation, which is essential for modeling its effectiveness. Each key parameter contributes to the warhead's ability to distribute fragments and ultimately destroy the target. The data is organized as follows:

- **Explosive Data**: The warhead uses PBX-9404, a high-energy explosive with a density (ρ_e) and detonation velocity (D). The explosive mass (M_e), Gurney constant (G), and the initiation point (X_2) are key factors in determining the velocity and energy imparted to the fragments upon detonation.

- **Material Data**: The warhead casing is made from mild steel, providing sufficient strength to contain the explosive until detonation. The density (ρ) of the steel is used to calculate structural mass and other characteristics relevant to energy transfer during detonation.

- **Geometric Data**: Geometric properties such as structural diameter (D_s), internal diameter (D_i), and effective length (L_e) determine the size and volume of both the warhead structure and the explosive content. These dimensions influence explosive power and fragment distribution.

- **Pre-formed Fragment Data**: The fragments are pre-formed as spherical mild steel pieces and arranged in a brickwork pattern. This arrangement ensures uniform fragment dispersion. Parameters include fragment diameter ($\overline{\mathrm{AO}}$), distance between fragment centers ($\overline{\mathrm{OB}}$), and sphere mass (m_{sph}). The total number of fragments (N_{sph}) and their combined mass (M_{sph}) are essential for assessing lethality.

- **Fragment Initial Velocity (V_0)**: Using the Gurney equation, the initial velocity of the fragments (V_0) is estimated. This velocity is critical for determining how far and how fast the fragments will travel toward the target, which directly impacts P_{kill}.

In summary, the combination of explosive, material, geometric, and fragment properties forms the foundation of the warhead's lethality model. These data points will be integrated into subsequent probabilistic models to evaluate P_{kill} under varying combat scenarios, providing a comprehensive assessment of the warhead's effectiveness.

7.4.4.3 Warhead Performance Metrics

The initial velocity (V_0) of the warhead fragments post-detonation is determined using the Gurney equation:

$$V_0 = G \cdot \sqrt{\frac{C/M}{1 + 0.5 \cdot C/M}}, \tag{7.1}$$

where G is the Gurney constant of the explosive, and C/M is the ratio of the explosive mass to the total mass of the warhead's structural components and fragments.

Additionally, static ejection angles are necessary for determining the initial dispersion of the fragments. These angles represent the directional pattern of fragment ejection relative to the warhead, influenced by the warhead's design and explosive-structure interaction. Understanding these angles helps optimize fragment spread for maximum target coverage.

7.4.4.4 Endgame Condition Determination

In this stage, the model considers operational flight conditions at the moment of detonation. The conditions include:

- **Missile Velocity (V_m)**: Determines the additional kinetic energy imparted to the fragments, influencing their relative velocity to the target.
- **Target Velocity (V_t)**: Affects the relative speed and encounter geometry, impacting the likelihood of fragment impact.
- **Miss Distance**: The predicted closest distance between the missile and the target at detonation, directly affecting P_{kill} by determining how many fragments can reach the target.
- **Altitude**: Influences air density, affecting drag on the fragments and their ability to penetrate the target.

7.4.4.5 Endgame Dynamics Computation

The endgame dynamics involve computing dynamic ejection angles and fragment velocities relative to the missile's trajectory to predict the interaction with the target. Unlike static ejection angles, dynamic ejection angles consider missile velocity at detonation, modifying the trajectory of each fragment.

The relative velocity of each fragment with respect to the moving target is calculated to assess the likelihood of impact. This velocity depends on the initial velocity of the fragment (V_0), missile velocity (V_m), and target velocity (V_t). The computed relative velocity and ejection angles allow for the estimation of the distance each fragment will travel post-detonation, which is essential for assessing their ability to reach and penetrate the target.

7.4.4.6 Fragment Spray Density and Lethality Assessment

The fragment spray density represents the concentration of fragments within the target zone, a key factor in determining lethality. The density of fragment saturation directly

influences the number of impacts on the target, affecting P_{kill}.

Several factors are combined to evaluate lethality, including:

- **Drag Coefficient (C_D)** and **Air Density (ρ)**: These parameters affect how quickly fragment velocity decreases due to air resistance. A higher air density or drag coefficient results in greater velocity reduction, reducing penetration likelihood.
- **Impact Velocity**: The impact velocity of the fragments determines their ability to inflict damage upon reaching the target. This velocity is influenced by the drag experienced during flight.

The lethality assessment combines these factors to estimate the probability of kill per fragment hit, ultimately leading to the overall kill probability for each fragment.

7.4.4.7 Computation of Final Probability of Kill

For the calculation of P_{kill} in this work, it was assumed that all missile launches used consistent values for key parameters. The intrinsic missile parameters, which are assumed to remain constant across different simulations, include: fragment initial velocity $V_0 = 2028$ m/s, effective length $L_{\text{e}} = 321$ mm, internal diameter $D_{\text{i}} = 144$ mm, structural diameter $D_{\text{s}} = 154$ mm, and fragment diameter $\overline{\text{AO}} = 10$ mm.

On the other hand, the parameters that may vary depending on the simulation context are the missile velocity $V_m = 592$ m/s, target velocity $V_t = 250$ m/s, target wingspan of 8.13 m, and the angle between trajectories, which was set to 0 radians for this study. This distinction helps clarify which parameters are intrinsic to the missile type and which are subject to change depending on the specific conditions of the simulation.

Since these parameters remain constant for all launches, the miss distance becomes the primary factor influencing the variation in P_{kill}. This assumption simplifies the model, allowing us to focus on optimizing warhead lethality based on detonation distance.

The final P_{kill} integrates the structural and explosive characteristics of the warhead with dynamic ejection analysis and fragment lethality assessment. This probability provides a statistical measure of the warhead's effectiveness and its likelihood of achieving its intended destructive outcome upon the target.

This model integrates principles from materials science, ballistics, aerodynamics, and probability theory to predict warhead performance, from detonation to target impact. By incorporating the miss distance obtained from the simulation, we transform this distance into a P_{kill} value, enhancing the model's predictive capacity. Such an approach is important for optimizing warhead configurations to maximize their efficacy. Interested readers

can refer to the code for lethality calculations available at the following repository [†].

7.4.5 Supervised Machine Learning Models

We created supervised machine learning models using a subset of the initially proposed input variables, excluding the `load_factor` variable, as mentioned previously, to develop a regression model capable of predicting the missile P_{kill}, which was calculated from the dataset based on the miss distance from the target, using the lethality theory described in the previous subsection. Notably, variables such as `turn_dg` and `delay` are not directly available to the launcher and require assumptions for their estimation. These variables will be addressed in detail in a later subsection. We employed three distinct algorithms to build the models: Polynomial Regression (PR), Artificial Neural Networks (ANN), and Extreme Gradient Boosting (XGBoost).

Before starting creating and training the models, we performed a stratified train-validate-test split for all supervised machine learning algorithms regarding 34 different probability blocks (bins) using Doane's formula (DOANE, 1976), allocating 80% for training and validation using a 5-fold cross-validation technique and 20% for testing, following the approach proposed in previous works (DANTAS *et al.*, 2021b; DANTAS *et al.*, 2023b). This dataset separation will allow the evaluation of the machine learning models later.

PR extends the linear model by adding extra predictors obtained by raising each original predictor to a determined degree. Adding polynomial terms to the linear model can effectively allow the model to identify nonlinear patterns (KUHN; JOHNSON, 2019). The degree of the polynomial controls the number of features added. Using a degree greater than four is unusual because the polynomial curve can become overly flexible and take on some unusual shapes (JAMES *et al.*, 2013). We proposed three different models with degrees from one to three using the Scikit-learn library (PEDREGOSA *et al.*, 2011b). For additional information, refer to the code associated with the PR models [‡].

ANN seeks to approximate the function represented by the dataset, calculating the error between the predicted outputs and the expected outputs and minimizing this error during the training process, working as function approximation machines that are designed to achieve statistical generalization (GOODFELLOW *et al.*, 2016). In this work, we used a multi-layer perceptron using the backpropagation algorithm, adjusting the weights of the connections between ANN neurons to minimize the mean-squared error loss (REED; MARKS, 1998). We performed data scaling to equally distribute the importance of each input in the ANN learning process (PRIDDY; KELLER, 2005). We used a min-max scaler, transforming all data features to a range from 0 to 1 (BONACCORSO, 2017). For the ANN

[†]Lethality code: `https://github.com/jpadantas/poker/tree/main/lethality`
[‡]PR code: `https://github.com/jpadantas/poker/tree/main/pr`

model, we utilized TensorFlow (ABADI *et al.*, 2016) to run 50 different hyperparameters configurations, changing the number of hidden layers in {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} and the number of units in {16, 32, 64, 128, 256}. All nodes have a rectified linear activation function (ReLU) (GOODFELLOW *et al.*, 2016). In addition, we included the Adaptive Moment Estimation (Adam) optimizer, a well-known training algorithm for ANN (BOCK; WEIß, 2019). Adam is a stochastic gradient descent method based on adaptive estimation of first- and second-order moments function (KINGMA; BA, 2014). We typically should choose a batch size between one and a few hundred. For a given computational cost, small batch sizes achieve the best training stability and generalization performance across diverse experiments (MASTERS; LUSCHI, 2018). We selected 32 as the model's batch size, considered a reasonable default value (BENGIO, 2012). For more details, check the code associated with the ANN models [§].

XGBoost, a leading gradient boosting framework, is engineered to construct a robust ensemble of decision trees by iteratively correcting errors from previous trees, thereby enhancing the model's accuracy for regression, classification, and ranking tasks and is considered the state-of-the-art for tabular data, as evidenced by its widespread adoption and superior performance in various competitions and applications (CHEN; GUESTRIN, 2016). In our study, we leveraged an extensive array of hyperparameters, including `n_estimators` set to an ambitious 1,000,000 trees for comprehensive data exploration, and `max_depth` options ranging in {10, 12, 14, 16, 18, 20}, enabling the model to uncover complex patterns without excessively increasing complexity. The learning rate was varied across {0.01, 0.1, 0.2, 0.3, 0.4}, balancing the convergence speed with the solution's accuracy. Alongside this, subsampling and feature sampling rates were set using `subsample`, `colsample_bytree`, and `colsample_bylevel`, which were varied in the range from 0.5 to 1.0 with a step size of 0.1, to enhance model generalization. Additionally, `min_child_weight` and `gamma` parameters were finely adjusted to regulate the model's growth and complexity, with `min_child_weight` in {1, 3, 5} and `gamma` in {0.0, 0.1, 0.2, 0.3, 0.4}. For further insights, please consult the code for the XGboost models [¶].

For both the ANN and XGBoost models, we implemented an early-stopping mechanism to oversee the training phase efficiently. This strategy entailed continuous monitoring of performance metrics on the validation set to detect any stagnation in improvement. We defined "patience" as the maximum number of epochs to continue training without observing any advancement in the validation set's metrics, setting this threshold at 10 epochs. This specific interval proved to be balanced for mitigating the effects of noise within the model optimization processes, ensuring that both models ceased training at a juncture that prevented overfitting while maximizing performance (GOODFELLOW *et al.*, 2016).

[§]ANN code: `https://github.com/jpadantas/poker/tree/main/ann`
[¶]XGBoost code: `https://github.com/jpadantas/poker/tree/main/xgboost`

The choice of machine learning models in this study was made to provide a balanced comparison between simplicity and performance. We included PR as a simpler baseline model, which we anticipated would be faster to train but likely result in lower performance compared to more complex models. This choice allowed us to understand the trade-offs between model complexity and accuracy in our context. The ANN and XGBoost models were selected as they are well-known for their effectiveness with tabular data and are frequently used for achieving high predictive metrics in such scenarios. This combination of models ensured a comprehensive evaluation of different approaches, ranging from simpler to more sophisticated techniques, thereby providing valuable insights into model performance under different levels of complexity.

7.4.6 Models Evaluation

Since this work analyzed a regression problem, we evaluated all the supervised machine learning models using the following well-known metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2). Additionally, training and inference time are two critical aspects of a model's assessment. We analyzed each cross-validation fold's training time and calculated each model's inference time on the test dataset. While training time is important, inference time is even more important as it directly impacts the model's practical applicability in real-time scenarios. We can estimate which model offers the best trade-off between performance and computational cost by examining these factors.

7.4.7 Estimating Turn Degree and Delay

Estimating the `turn_dg` and `delay` for a model presents significant challenges due to the inherent unpredictability of a target's response to an incoming missile. This unpredictability is largely attributed to the unique characteristics and decision-making processes of the target, which can vary widely in real-world scenarios. Understanding and predicting how a target will maneuver in response to a perceived threat is critical for accurate simulation and assessment of missile engagement outcomes.

To address this challenge, we employ the expertise of SMEs in military tactics and missile defense systems to estimate the potential `turn_dg` and `delay` of a target upon missile detection. The approach leverages the deep understanding of SMEs regarding the behavior of various targets under threat, allowing for a more informed and realistic set of estimations.

These parameters are essential for modeling the target's evasive maneuvers and the subsequent trajectory adjustments of the missile. We represent the target's response

variability using normal distributions for both `turn_dg` and `delay`. The choice of a normal distribution is based on the assumption that the target's response will vary around a mean value (median) with a certain degree of predictability, as influenced by the target's characteristics and situation. These distributions' median and standard deviation are determined through a collaborative process with SMEs, ensuring that the estimations reflect a realistic range of target behaviors.

The estimation process involves the identification of specific scenarios in which a target might find itself when a missile is launched toward it. Each scenario is carefully analyzed to determine the likely response of the target, focusing on two key parameters: the `turn_dg` and `delay` before initiating the turn.

To estimate the `turn_dg` of the target, we assume that the target will adopt the most defensive posture possible, always attempting to move in the direction opposite to the missile launcher. Regardless of the distance between the missile launcher and the target, the target aircraft will aim to maximize the separation by performing a defensive turn of up to 180 degrees, whenever feasible. This approach ensures that the target consistently prioritizes evasion by moving away from the threat to the greatest extent possible.

The initial value of the target's turn degree is defined by the following function:

$$\texttt{turn\_dg}_0 = ((\texttt{rgt\_tgt} + 180) \bmod 360) - 180. \tag{7.2}$$

To illustrate the initial value of the defensive behavior of the targets, Figure 7.7 presents ten diverse and well-spread scenarios, where each target attempts to perform a defensive maneuver aimed at maximizing separation from the launcher. The distances, initial headings, and off-boresight angles were chosen to ensure a good distribution in space, avoiding overlap and facilitating visualization of the maneuvers. Each target, represented by the red point, performs a turn to move away from the launcher (the blue point at the center) according to the suggested direction indicated by the red arrow, in an effort to maximize evasion. The turn degree ($\texttt{turn\_dg}_0$) is represented by the blue arc and varies according to the relative position of the target to the launcher.

To better reflect realistic behavior, the final value of `turn_dg` is modeled with added variability as follows:

$$\texttt{turn\_dg} = \texttt{turn\_dg}_0 \, (1 + 0.05 \cdot w), \tag{7.3}$$

where $\texttt{turn\_dg}_0$ is the value calculated from Equation 7.2, and $w \sim \mathcal{N}(0, 1)$ represents a standard normal random variable. This introduces a 5% variation, which accounts for natural fluctuations in the target's maneuver.

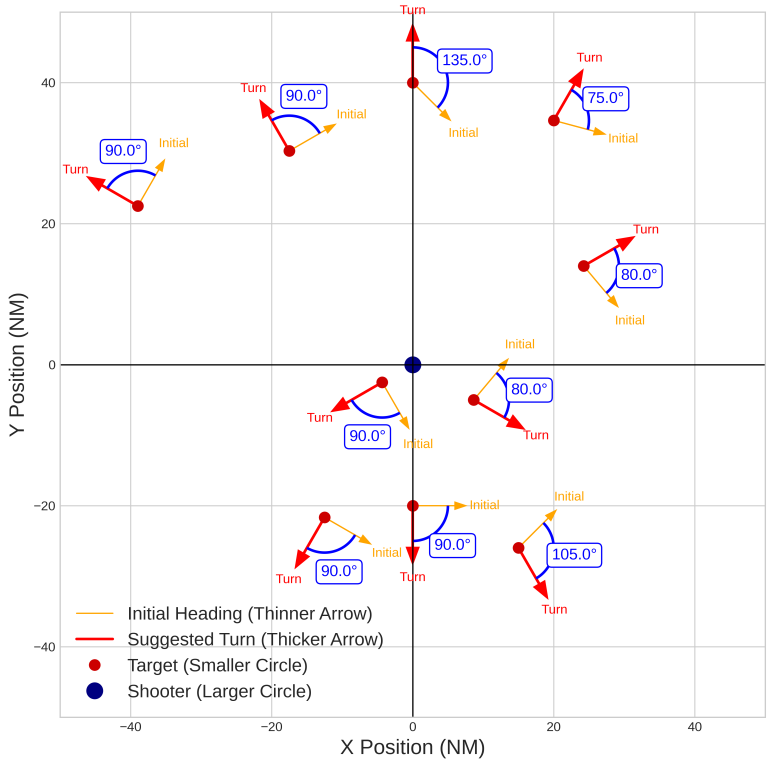


FIGURE 7.7 – Illustration of ten diverse scenarios of defensive turn maneuvers. Each scenario shows a target attempting to maximize distance from the launcher. The dark red points (smaller) indicate the initial position of the target, while the orange (thinner) and red arrows (thicker) represent the initial heading and the suggested turn direction, respectively. The dark circle (larger) represents the shooter. The blue arc indicates the turn degree (`turn_dg`) performed by the target, which varies between 0° and 180°.

Moreover, regarding the variable `delay`, we anticipate specific behaviors based on the target's orientation relative to the missile launcher. While we initially assumed that shorter reaction times would occur when the target directly faces the launcher due to the immediate perception of the threat, we acknowledge that this is not always the case. The reaction time depends on various factors, including the target's sensor capabilities and engagement strategy. For instance, even when the launcher is positioned behind the target, a Radar Warning Receiver (RWR) may immediately detect the radar, leading to a prompt response.

To simplify this context, we made certain assumptions about the target's behavior. Specifically, we structured the estimated reaction times within a range of 15 to 30 seconds. This range is further segmented into three distinct intervals, each spanning 5 seconds, to reflect varying degrees of threat awareness and response times. These intervals are divided into 120 degrees each to cover the full 360-degree range, ensuring that all possible orientations are accounted for. Each interval is associated with a standard deviation value of approximately 5%, which ensures the three curves are almost connected, allowing for comprehensive sampling of all potential values. Please refer to Figure 7.8 for a detailed

visualization of the considered scenarios and the corresponding distributions for `delay`.

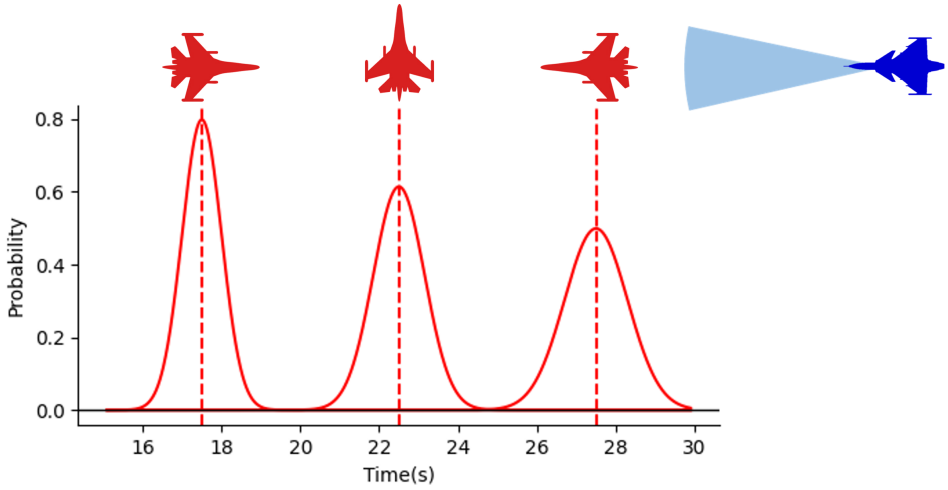


FIGURE 7.8 – Graph illustrating the target's delay time as a function of its orientation with respect to the shooter.

To validate the chosen values for the median and standard deviation, we employ the face validation method (BALCI, 1998). This method involves a review by an independent group of experts who assess the plausibility and accuracy of the estimations based on their own experience and knowledge. The face validation process ensures that the estimations are not only based on a solid theoretical foundation but also resonate with the practical insights of seasoned professionals in the field.

It is important to note that the parameters and distributions outlined in this section adhere to the intervals and assumptions previously described in this work. By leveraging SME knowledge and employing rigorous validation techniques, we aim to provide a robust framework for estimating `turn_dg` and `delay`, thereby enhancing the accuracy of missile engagement simulations. The validation process was conducted through face validation by SMEs with extensive experience in missile engagements. Specifically, several simulated scenarios were presented to the SMEs, covering a range of engagement distances, target maneuvers, and missile launches. The SMEs evaluated whether the estimated `turn_dg` and `delay` values aligned with their operational experience and expectations under different combat situations. This process ensured that the estimated values were realistic and consistent with real-world operational behavior.

7.4.8 The Probability of Hitting a Target

Determining the final probability of hitting a target is intricately linked to the analysis performed by machine learning algorithms. These algorithms are designed to process a comprehensive set of input variables derived from the aircraft's subsystems, providing

the pilot with actionable intelligence to enhance missile targeting accuracy. However, a critical component of this targeting process is the estimation of the target's `turn_dg` and `delay`, which are not directly observable and must be inferred through the methodology outlined previously.

This estimation plays an essential role in calculating the hit probability, accounting for the variability in the target's response behavior. By incorporating the target's expected maneuvers, the machine learning model can adjust its predictions to better reflect the dynamic nature of aerial engagements. The aim is to achieve a conservative estimation of the target's performance, ensuring that the probability of a hit considers the best possible maneuvers the target could execute to evade the missile.

To this end, the model leverages the variability in the target's behavior as an essential factor, allowing for a better understanding of potential evasive actions. This approach is grounded in the principle of anticipating optimal target performance, where the target utilizes its capabilities to the maximum extent to evade incoming threats. By assuming a high level of performance from the target, the model inherently considers the target to perform well on average, thereby minimizing the risk of underestimating the target's evasive capabilities.

Integrating this realistic estimation into the machine learning algorithm enhances the reliability of the hit probability calculation. It enables the algorithm to account for the widest possible range of target behaviors, including those that may occur under extreme conditions. This approach ensures the pilot has a probability of hit that reflects the most challenging engagement scenarios, facilitating informed decision-making during missile launch sequences.

In summary, the probability of hitting a target is not merely a static figure derived from straightforward inputs; it is a dynamic estimation that incorporates the complexity and unpredictability of aerial combat. By integrating conservative estimations of the target's `turn_dg` and `delay` into the machine learning response, we aim to improve the accuracy and realism of missile engagement outcomes. However, we acknowledge the limitations of the current approach, and further empirical validation is required to fully demonstrate the practical benefits of these estimations. This ongoing work lays the foundation for future studies that will more comprehensively assess the impact of these estimations on aerial defense strategies.

7.5 Results and Analysis

This section examines the metrics of the test dataset for the proposed machine learning models. Additionally, we provide a Multi-Function Display (MFD) representation,

incorporating the traditional maximum range and no-escape zone indicators along with the probability of kill in the WEZ indication based on the proposed probabilistic model.

7.5.1 Comparative Analysis of the Machine Learning Models

Table 7.3 summarizes the mean performance metrics for each model across the five folds of the three distinct proposed models: PR, ANN, and XGBoost. These metrics include MAE, MSE, RMSE, and R^2, reflecting their respective best hyperparameter settings, chosen based on performance on the validation dataset during training. It is important to note that the predicted variable, P_{kill}, is expressed as a percentage. Consequently, all the error metrics (MAE, MSE, RMSE) are dimensionless, as they are derived from percentage values. Additionally, R^2 is also dimensionless. The analysis further includes computational efficiency metrics, such as training and inference time, both measured in seconds, using the training/validation and test datasets, respectively.

TABLE 7.3 – Comparative evaluation of machine learning models based on the proposed metrics. The best results for each metric are highlighted in bold red text.

Model	PR	ANN	XGBoost
Hyperparameters	degree=4	hidden_layers=3, units=128	max_depth=14, learning_rate=0.01, subsample=0.9, colsample_bytree=0.9, colsample_bylevel=0.6, min_child_weight=5, gamma=0.1
MAE	11.52	**2.33**	5.04
MSE	296.26	**51.90**	117.14
RMSE	17.21	**7.20**	10.82
R^2	67.94	**94.38**	87.31
Training Time (s)	**532.23**	4722.17	6098.33
Inference Time (s)	9.57	**6.65**	8.20

7.5.2 Statistical Analysis

To determine whether the observed differences in model performance are statistically significant, we performed the Friedman test (FRIEDMAN, 1937) followed by the Nemenyi post-hoc test (NEMENYI, 1963) for multiple comparisons. Interested readers can refer to the code for the statistical analysis at the following repository ‖.

7.5.2.1 Friedman Test

The Friedman test was used to detect differences in performance across the three models using the results from 5 folds of each model. The null hypothesis is that there

‖Statistics code: `https://github.com/jpadantas/poker/tree/main/statistics`

are no differences in the performance metrics among the models. Table 7.4 presents the results of the Friedman test for each metric.

TABLE 7.4 – Friedman Test Results. Significant p-values are highlighted in bold red text.

Metric	Friedman Statistic	p-value
MAE	10.0	**0.0067**
MSE	10.0	**0.0067**
RMSE	10.0	**0.0067**
R^2	10.0	**0.0067**
Training Time	8.4	**0.0150**
Inference Time	4.8	0.0907

The results indicate that for all metrics except inference time, there are statistically significant differences among the models (p-value < 0.05). Notably, the p-values for several metrics (MAE, MSE, RMSE, and R^2) are identical, suggesting that the models perform similarly across these related metrics. This similarity could be due to the fact that these metrics are strongly correlated, and the relatively small sample size (5 folds) may limit the ability of the Friedman test to distinguish nuanced differences between models. Additionally, the identical Friedman statistics for these metrics imply that the rankings across models were consistent, reflecting similar performance trends.

7.5.2.2 Nemenyi Post-hoc Test

To identify which models differ significantly from each other, we conducted the Nemenyi post-hoc test. Table 7.5 shows the p-values for pairwise comparisons among the models for the metrics where the Friedman test was significant. Note that the inference time metric did not show significant differences in the Friedman test; therefore, the Nemenyi test was not conducted for this metric.

TABLE 7.5 – Nemenyi post-hoc test p-values for pairwise comparisons among models. Significant p-values are highlighted in bold red text.

Metric	Comparison	p-value
MAE	PR vs ANN	**0.0045**
	XGBoost vs PR	0.2541
	XGBoost vs ANN	0.2541
MSE	PR vs ANN	**0.0045**
	XGBoost vs PR	0.2541
	XGBoost vs ANN	0.2541
RMSE	PR vs ANN	**0.0045**
	XGBoost vs PR	0.2541
	XGBoost vs ANN	0.2541
R^2	PR vs ANN	**0.0045**
	XGBoost vs PR	0.2541
	XGBoost vs ANN	0.2541
Training Time	XGBoost vs PR	**0.0123**
	PR vs ANN	0.1394
	XGBoost vs ANN	0.6008

The Nemenyi post-hoc test results, shown in Table 7.5, indicate that significant differences primarily occurred between PR and ANN. For XGBoost, its comparisons with both PR and ANN resulted in similar p-values across multiple metrics. This suggests that XGBoost's performance was relatively stable and consistently fell between the other two models. This pattern might reflect limited differentiation capability given the inherent relationships among the metrics and the modest sample size, highlighting that larger samples might be needed to detect finer performance differences.

7.5.3 Discussion

The statistical analysis confirms that the choice of model has a significant impact on performance metrics and training time. The Friedman test indicated statistically significant differences among the models for MAE, MSE, RMSE, R^2, and Training Time (p-value < 0.05). The Nemenyi post-hoc test further identified that:

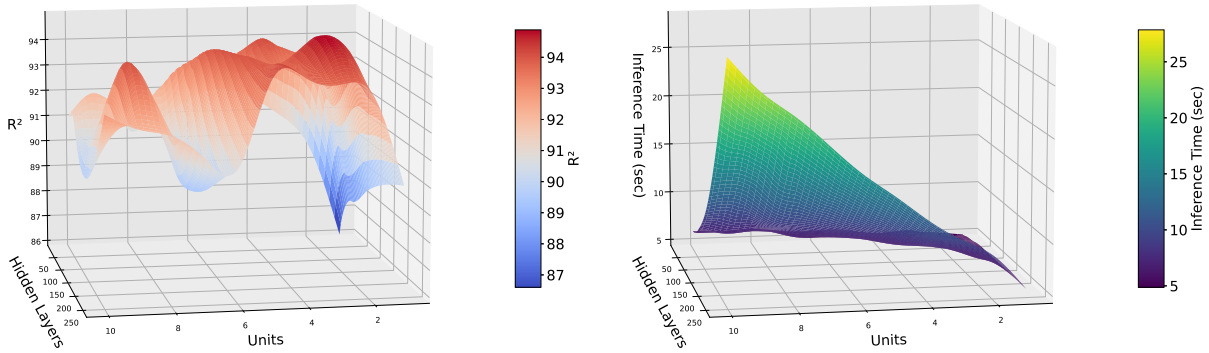
- For MAE, MSE, RMSE, and R^2, ANN significantly outperformed PR (p-value = 0.0045).
- No significant differences were found between XGBoost and PR, or XGBoost and ANN for these metrics.
- For Training Time, PR had a significantly shorter training time than XGBoost (p-value = 0.0123).
- No significant differences were found between PR and ANN, or XGBoost and ANN for Training Time.

These results suggest that ANN provides superior predictive accuracy compared to PR, with a statistically significant improvement in error metrics. Although ANN's training time was longer than PR's, the difference was not statistically significant between ANN and PR, indicating that the increased computational cost is acceptable given the performance gains.

Among the models tested, ANN exhibited the lowest error metrics across all considered measures. Specifically, ANN achieved the lowest MAE (2.33), suggesting its predictions were closer to the actual values on average compared to PR and XGBoost. Similarly, ANN recorded the lowest MSE (51.90) and RMSE (7.20), indicating a strong capability to minimize both variance and bias in its predictions. The R^2 value for ANN stood at 94.38%, demonstrating that it could explain a substantial portion of the variance in the dataset, which indicates strong predictive power.

Regarding computational efficiency, PR had the shortest training time (532.23 seconds), which was significantly shorter than that of XGBoost (6098.33 seconds) as indicated by the Nemenyi test (p-value = 0.0123). ANN's training time (4722.17 seconds) was not significantly different from either PR or XGBoost. In terms of inference time, ANN was the fastest (6.65 seconds), slightly quicker than XGBoost (8.20 seconds) and PR (9.57 seconds).

Specifically for the ANN, Figure 7.9 illustrates a comparative analysis of the neural network's performance metrics, demonstrating the best results among the evaluated models. The three-dimensional surface plots capture how various configurations of model units and layers affect both model accuracy, measured by the mean R^2 value, and the inference time required for predictions. Plot (a) demonstrates that the relationship between the number of units and layers significantly impacts accuracy, indicating an interplay between model depth and performance. Plot (b) shows an increase in inference time with more complex models, emphasizing the need to consider computational efficiency alongside model precision. These visualizations highlight the balance that must be struck in neural network architecture design to optimize both performance and efficiency.



(a) (b)

FIGURE 7.9 – Comparative analysis of neural network performance metrics. (a) illustrates how model configurations, varying in units and hidden layers, impact accuracy through the mean R^2 value. (b) delineates the effect of model complexity on inference time, measured in seconds.

XGBoost achieved moderate accuracy metrics with an MAE of 5.04, MSE of 117.14, RMSE of 10.82, and an R^2 of 87.31%. While its error metrics were higher than those of ANN, the differences were not statistically significant according to the Nemenyi test (p-value = 0.2541 for XGBoost vs PR and p-value = 0.2541 for XGBoost vs ANN). XGBoost's performance was acceptable and it may serve as a robust alternative, especially in contexts where ensemble methods are preferred. Its inference time (8.20 seconds) was shorter than PR's but slightly longer than ANN's, with no significant differences detected.

The simplicity of PR resulted in lower predictive performance compared to more advanced models. Its error metrics were significantly higher than those of ANN, as confirmed by the Nemenyi test (p-value = 0.0045 for PR vs ANN). PR lagged behind, with the high-

est MAE (11.52), MSE (296.26), RMSE (17.21), and the lowest R^2 value (67.94%). These results highlight the limitations of PR in capturing complex patterns within the dataset, especially when compared to models like ANN and XGBoost, which are better suited for such complexity.

Note that discrepancies in computational times may be due to differences in library implementations, as PR and ANN were implemented using different libraries, which could impact their computational efficiency.

In summary, the statistical analysis confirms that ANN provides significantly better predictive accuracy compared to PR, without a significant increase in training time. While XGBoost did not show significant differences compared to the other models, it offers a balance between performance and computational efficiency. These findings can guide practitioners in selecting the appropriate model based on the specific requirements of accuracy and computational efficiency, considering both statistical significance and practical performance metrics.

7.5.4 Enhanced Weapon Engagement Zone Representation

We introduce an enhanced MFD visualization for the WEZ, integrating a probabilistic model that encompasses the traditional maximum range and no-escape zone indicators and a novel probability of kill aspect within the WEZ indications. By applying the proposed probabilistic model, we adjusted the target's off-boresight angles from $-60°$ to $+60°$. The enhanced WEZ visualization, depicted in Figure 7.10, now incorporates P_{kill}, offering a more comprehensive understanding of the engagement zone. Incorporating this metric alongside traditional WEZ metrics may improve a pilot's situational awareness and decision-making capabilities, particularly in controlled scenarios. However, this potential benefit is heavily dependent on model assumptions and requires further validation for broader applicability. This integration can possibly support pilots in making more informed decisions regarding missile launches against potential targets by providing a clearer assessment of the likelihood of successfully engaging a target under various conditions. Including a probabilistic kill probability metric may aid in optimizing the use of onboard weaponry by evaluating the tactical viability of a missile launch, thus enhancing operational effectiveness and minimizing resource loss.

7.6 Outcomes

In conclusion, this work presents PoKER, an innovative probabilistic model for WEZ analysis in BVR air combat scenarios. By extending beyond traditional WEZ metrics

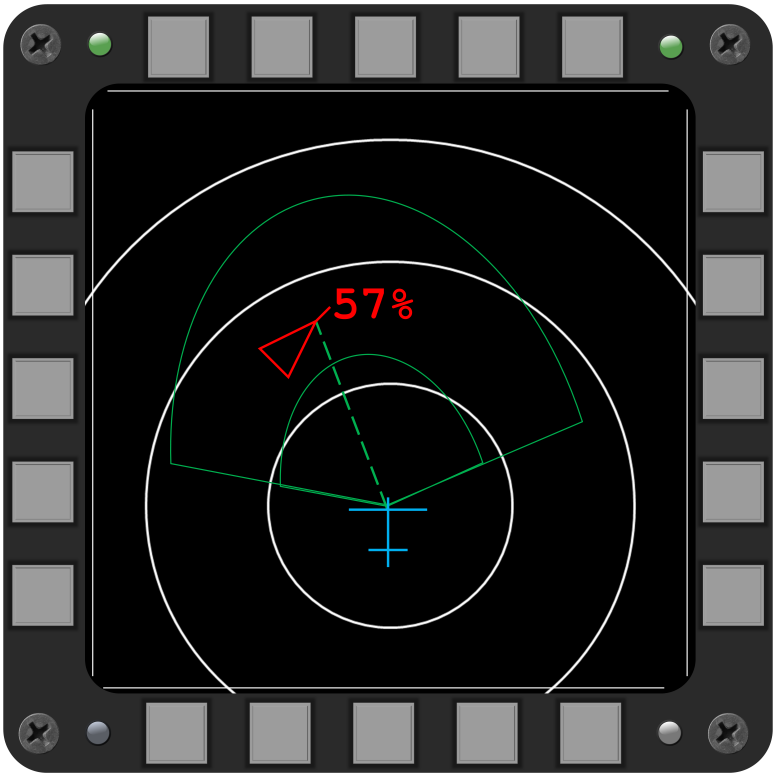


FIGURE 7.10 – Enhanced MFD representation incorporating the integration of P_{kill} alongside the maximum range and no-escape zones, providing a multifaceted view of the engagement zone to support informed decision-making in missile launch scenarios.

to include a probability of kill calculation, PoKER enhances the precision and relevance of missile engagement assessments. This integration may improve the pilot's situational awareness and decision-making capabilities by providing a better understanding of engagement outcomes. The capability of PoKER to offer real-time, accurate estimations of missile effectiveness based on dynamic combat conditions and target behavior represents a notable advancement in air combat strategy and planning. Its application not only can enhance operational efficiency and effectiveness but also contributes to the optimal use of onboard weaponry, thus minimizing resource loss and maximizing engagement success.

For future work, we advocate the utilization of data derived from real training missile launches, whether from virtual simulations or live exercises, to refine the modeling of target behavior. The introduction of human-in-the-loop simulations could significantly enhance the model's practical application by aligning theoretical predictions more closely with real-world decision-making processes. Furthermore, the adoption of advanced machine learning techniques promises to make WEZ modeling more adaptive to changing combat conditions. Enhancements in missile simulation accuracy, alongside investigations into the impact of environmental variables and electronic warfare, could offer deeper insights into engagement strategies. Moreover, developing methods to predict adversary tactics and

countermeasures accurately will greatly improve PoKER's strategic utility. Additionally, creating a representation of how to maneuver into regions of higher probability could inform pilots on effective maneuvers before a missile launch.

We also agree that evaluating the dataset size is an important contribution, especially for the scalability of future work. While 10 million simulations were feasible for this study due to available computational resources, we acknowledge that this may not be practical for all scenarios. Therefore, we suggest exploring methods to reduce the dataset size without compromising model fidelity, such as using efficient sampling techniques or selecting a subset of representative points. This could help balance computational costs while maintaining the quality of the analyses.

Furthermore, we plan to conduct empirical studies to validate the overall effectiveness of the proposed model across different scenarios. These studies will provide valuable insights into the model's strengths and limitations in varied operational contexts. While these studies will help assess the model's applicability, incorporating such estimations into real-world systems will require further investigation, which we suggest as a direction for future work as well. The ultimate goal is to demonstrate the utility of PoKER in enhancing situational awareness and decision-making for both autonomous agents and human pilots, particularly in scenarios involving AI-driven adversaries.

By integrating such predictive capabilities, pilots can make more informed decisions, enhancing their engagement strategies. Such efforts can solidify the model's foundational strength and expand its applicability in addressing the complexities of modern air combat. By pushing the boundaries of current WEZ models, this study contributes to academic understanding and offers practical insights for military strategists and defense technology developers.

7.7 Source Code

The complete source code and detailed instructions for reproducing the results are available at `https://github.com/jpadantas/poker`.

Part IV

In-Flight Tactical Systems

8 Air Combat Engagement

This chapter explores the intricate process of engagement decision support for BVR air combat. The research presented highlights methodologies for enhancing decision-making in BVR scenarios. By leveraging machine learning techniques, the study aims to optimize engagement strategies, providing resources to increase efficiency in air combat operations. As indicated in Figure 1.2, this chapter primarily contributes to the "In-Flight Tactical Systems" area within the proposed research framework.

This chapter draws upon the following work:

> **DANTAS, J. P. A.**; COSTA, A. N.; GERALDO, D.; MAXIMO, M. R. O. A.; YONEYAMA, T. Engagement Decision Support for Beyond Visual Range Air Combat. In: Proceedings of the 2021 Latin American Robotics Symposium, 2021 Brazilian Symposium on Robotics, and 2021 Workshop on Robotics in Education. Proceedings [...]. Natal, RN, Brazil: IEEE, 2021. p. 96–101.

8.1 Summary

This work aims to provide an engagement decision support tool for Beyond Visual Range (BVR) air combat in the context of Defensive Counter Air (DCA) missions. In BVR air combat, engagement decision refers to the choice of the moment the pilot engages a target by assuming an offensive stance and executing corresponding maneuvers. To model this decision, we use the Brazilian Air Force's Aerospace Simulation Environment (*Ambiente de Simulação Aeroespacial - ASA* in Portuguese), which generated 3,729 constructive simulations lasting 12 minutes each and a total of 10,316 engagements. We analyzed all samples by an operational metric called the DCA index, which represents, based on the experience of subject matter experts, the degree of success in this type of mission. This metric considers the distances of the aircraft of the same team and the opposite team, the point of the Combat Air Patrol, and the number of missiles used. By defining the engagement status right before it starts and the average of the DCA index throughout the engagement, we create a supervised learning model to determine the

quality of a new engagement. An algorithm based on decision trees, working with the XGBoost library, provides a regression model to predict the DCA index with a coefficient of determination close to 0.8 and a Root Mean Square Error of 0.05 that can furnish parameters to the BVR pilot to decide whether or not to engage. Thus, using data obtained through simulations, this work contributes by building a decision support system based on machine learning for BVR air combat.

8.2 Introduction

Air combat may occur in two primary forms: Within Visual Range (WVR) and Beyond Visual Range (BVR) (KURNIAWAN *et al.*, 2019), with the latter being recently more developed in the operational context, due to more extensive availability of more advanced weapons and sensors (HIGBY; COL, 2005). Notice that, even though modern air combat may still end WVR, through a series of complex decisions and maneuvers, it usually begins BVR, which frequently is the most critical phase of the combat since it may provide advantages and drawbacks for succeeding phases (YUAN *et al.*, 2016). There is no clear definition of the distance to differentiate these two forms of air combat since this may be subject to the conditions in which the air combat happens.

BVR conditions force pilots to rely more on a series of systems to compose their situational awareness, allowing them to make tactical decisions during combat, such as whether to fire a missile or not. Especially in BVR combat, the missile launches and the circumstances around these events are critical (ARONSSON *et al.*, 2019) since these weapons are the main form of engaging the opponent. Since BVR combat is rarely observable in practice, with low availability of historical data, much of its possibilities must be assessed through simulation (STILLION, 2015). It is also true due to the high costs of flying, air space regulations, and limited availability of platforms representative of those used by opposing forces (KÄLLSTRÖM; HEINTZ, 2020).

This work contributes by developing a decision support tool for pilots in BVR combat situations based on simulated data, with a particular focus on solving the problem of deciding when to engage a specific enemy aircraft, which is commonly based solely on pilot experience. There are several previous approaches that relate to this decision, mainly providing different forms of modeling pilot behavior. Many of them use game theory to model air combat (KARELAHTI *et al.*, 2006; VIRTANEN *et al.*, 2006; MUKAI *et al.*, 2003; HA *et al.*, 2018). Other approaches found in the literature are as follows: Bayesian Networks (POROPUDAS; VIRTANEN, 2007; FU *et al.*, 2021; RAO *et al.*, 2011a; DU; LIU, 2010), fuzzy logic (AKABARI *et al.*, 2005; PRABHU *et al.*, 2014), agent-based modeling (HEINZE *et al.*, 1998), influence diagrams (LIN *et al.*, 2007), reinforcement learning (HU

et al., 2021; TOUBMAN *et al.*, 2016; PIAO *et al.*, 2020; WEILIN *et al.*, 2018), artificial neural networks (DANTAS, 2018; YAO, 2021), evolutionary algorithms (LI *et al.*, 2020; YANG *et al.*, 2020), minimax method (KANG *et al.*, 2019), and behavior trees (YAO *et al.*, 2015a).

Among all of these approaches, Ha *et al.* (2018) is the one that more directly focuses on estimating the firing moment, which is done through a probabilistic function of the target's evasive maneuverability, the missile's speed on the final approach, and the accuracy of the target's information to guide the missile. However, as with many of the other cited methods, the methodology proposed in Ha *et al.* (2018) has not been tested in simulations with a higher degree of fidelity concerning an actual BVR air combat.

As an alternative, our work uses supervised machine learning models based on decision trees, using the XGBoost library (CHEN; GUESTRIN, 2016), to provide pilots with parameters that improve their situational awareness in air combat from data collected from simulations of operational scenarios, allowing them to better decide when to engage a target. Compared to most of the approaches, this study was conducted through higher fidelity simulations, providing systems and subsystems that resemble their natural counterparts through our simulation environment, such as 6 degrees of freedom (6DOF) multi-role combat aircraft, electronic warfare (EW) devices, datalink communications, and active radar-guided missiles.

This study is, therefore, developed around four main subjects: **(i)** determination of BVR air combat scenarios that will be analyzed, **(ii)** carrying out simulations based on the chosen scenarios in the agents' configurations are varied, **(iii)** collecting and analyzing the data generated in the simulations, and **(iv)** using machine learning techniques on this data to improve the pilot's situational awareness by providing information about the analyzed air combat, fulfilling the role of a decision support system.

The remainder of this chapter is organized as follows. In Section 8.3, the main characteristics of the aircraft in the BVR air combat are described. We also explain the Defensive Counter Air (DCA) index, the operational metric used to perform the engagement analysis. Besides, the sampling process of the simulation inputs is discussed. In Section 8.4, we show the methodology used to create the solution approach. Section 8.5 investigates the results and analysis concerning the exploratory data analysis and model results. Finally, in Section 8.6, we present the conclusions and future works.

8.3 Proposal Description

This section shows the Fighter Agent model, describing the simulation agent's reasoning that we propose. Next, the DCA index, the operational metric that we created to evaluate the air combat engagements, is presented. Finally, we discuss our process of

sampling simulation input parameters.

8.3.1 Fighter Agent

The Institute for Advanced Studies from the Brazilian Air Force develops the Aerospace Simulation Environment (*Ambiente de Simulação Aeroespacial - ASA* in Portuguese) to provide a computational solution that enables the simulation of operational scenarios, allowing users to establish scenarios, parameters, and command decisions to support the development of tactics, techniques, and procedures. Since there is so much variability in the nature of military scenarios, it may be hard to define what the scenarios of interest are. Thus, the development of ASA is not limited to predefined layouts; instead, it poses itself as a flexible solution that may be tailored to the user's needs. Its modularity contributes to this flexibility since one can configure components (models) in diverse ways and combine them freely, which enables the creation of the most varied scenarios (COSTA, 2019).

ASA is a custom-made, object-oriented (C++), and high-fidelity environment that generates the simulations and the data to analyze the problem. The simulation concept addressed in this context is one in which the scenario elements are represented as agents capable of making decisions based on artificial intelligence models or arbitrary rules previously stipulated. Simulations of this nature, termed constructive, can be used in the decision-making process, for example, to predict possible outcomes of engagements between opposing forces and assist in the definition of lines of action.

The agent's actions are managed by rules describing a pilot's main tactical behaviors during a BVR combat. This set of rules is modeled as a Finite State Machine (FSM), or finite automaton, which is a mathematical computer model that, at any time, can be in only one state of a limited set of states (MAMESSIER *et al.*, 2014). Each state is one of the possible tactics that can be taken during a BVR combat. The following behaviors will be analyzed to understand better the presented problem: Combat Air Patrol (CAP), Commit, Abort, and Break. Figure 8.1 shows the FSM that manages all the agent's tactics.

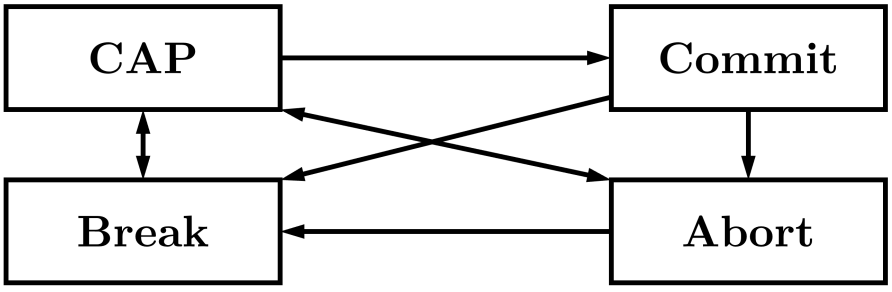


FIGURE 8.1 – FSM of agent tactics.

The CAP tactic consists of performing a flight pattern, describing an orbit, that can be any tactical maneuver, such as a circle or an oval, around a specific position called CAP point, with a defined heading and direction (clockwise or counterclockwise). The Commit tactic consists of the agent engaging a target detected by its radar or shared by data link by its allies. In the Abort tactic, the agent performs a defensive maneuver to move away from his priority threat, which is also the priority target in many situations. Finally, when the agent's sensors detect a missile threat fired in its direction, it performs the Break tactic. This tactic consists of a sudden defensive maneuver, describing a curve and a dive with great acceleration.

At the engagement moment, the FSM assumes the Commit behavior and will keep performing offensive maneuvers until the time it is desirable within combat. The offense and vulnerability indices are essential variables that guide the change from one state to another within the FSM. Then, when the agent needs to do a defensive movement, it assumes Break or Abort. Therefore, the engagement time in a simulation is defined as the time between the first Commit carried by the agent and its first Break or Abort. We created an operational index that determines the quality of the actions taken by the agent in a BVR scenario, which was calculated during the engagement period and can be extended to any moment in the simulation. This metric is referred to as the DCA Index, which will be detailed in the following subsection.

8.3.2 DCA Index

We defined the index as a probability of success, ranging from 0% to 100%, for BVR combat on DCA missions whose objective is to establish a CAP. These missions have the goal of defending a point of interest, which is done by ensuring that the opposing aircraft are kept far away from it. In addition, it aims to do that while launching the least number of missiles possible, which is interesting both from economic and operational standpoints. Furthermore, from the doctrine perspective, one may consider it good practice for the defending aircraft to stay close to its CAP point since it is easier to employ tactics to defend the point of interest.

From these considerations with respect to the DCA mission context, we defined three basic principles (depicted in Figure 8.2) for the elaboration of the DCA index:

1. Minimize the number of missile launched in the mission: $(m_{total} - m_{avail})$.
2. Minimize the reference distance from its CAP point: $D(r, CAP)$.
3. Maximize the distance of each enemy (e_n) to the CAP point: $D(e_n, CAP)$.

Firstly, at the beginning of the engagement, each aircraft has a fixed number of missiles

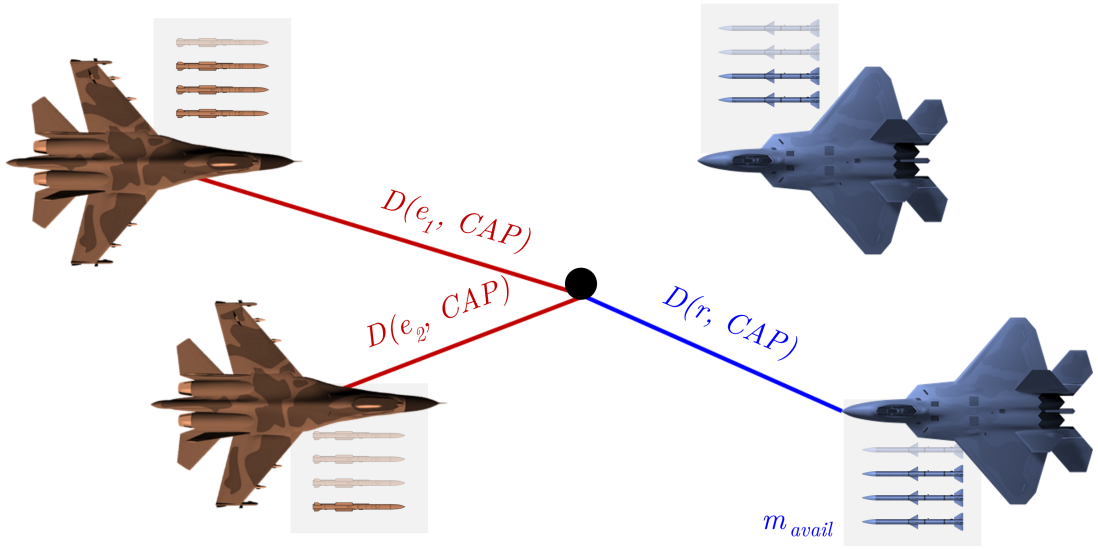


FIGURE 8.2 – Representation of the factors that form the DCA index.

(n_{total}), which at that moment is also the number of missiles available (n_{avail}). When the aircraft launches a missile, n_{avail} decreases to keep track of the currently available missiles. The ratio between n_{avail} and n_{total} is one of the factors of the DCA index, so that, when maximizing this ratio, $n_{total} - n_{avail}$ is minimized.

Secondly, regarding minimizing the distance between the reference aircraft and the CAP point, i.e., $D(r, CAP)$, we considered that the decay of this effect is not linear since, as the distance increases, its influence becomes much less relevant to fulfilling the mission goals. Therefore, we chose a sigmoid function to generate such decay to encompass this non-linearity. We defined the sigmoid limits, considering operational experience with respect to the maximum range of the available missile, as 8,000 meters ($x_{99\%,r}$) for the 99% output value, which corresponds to $y_{99\%,r} \approx 4,5951$, and as 12,000 ($x_{1\%,r}$) meters for the 1% output value, which stands for $y_{1\%,r} \approx -4,5951$. These limits must be used in a linear interpolation (8.1) to convert the current distance value (D_r) to be input (d_r) in the sigmoid equation (8.2).

$$d_i = \frac{(y_{99\%,i} - y_{1\%,i})}{(x_{99\%,i} - x_{1\%,i})} \cdot [D(i, CAP) - x_{1\%,i}] + y_{1\%,i} \tag{8.1}$$

where: i = r (reference) or e_n (enemy)
D_i = measured distance from the CAP point
d_i = interpolated distance for sigmoid input

Lastly, when considering the opposing aircraft distances ($D(e_n, CAP)$), for all N enemies, we used a similar sigmoid function, but with opposite characteristics since the idea was to increase the enemies' influence when they were closer to the CAP point. Therefore, the sigmoid limits for this factor were instead 12,000 ($x_{99\%,e}$) meters for the 99% output

value, which corresponds to $y_{99\%,e} \approx 4,5951$, and 8,000 ($x_{99\%,e}$) meters for the 1% output value, which stands for $y_{1\%,e} \approx -4,5951$. Applying the enemy sigmoid limits in the interpolation equation (8.1), we are able to convert the measured distance value $D(e_n, CAP)$ to be input (d_{e_n}) in (8.2) for each of the enemy aircraft.

With the composition of the previously calculated factors, it is possible to obtain the DCA index's final calculation. The index factors have coefficients to allow prioritization of the three principles presented, as seen in (8.2). We defined the weights ($w_1 = 0.2$, $w_2 = 0.4$, and $w_3 = 0.4$) for each factor of the DCA index (I_{DCA}) based on the operational knowledge from subject matter experts.

$$I_{DCA} = w_1 \cdot \frac{m_{avail}}{m_{total}} + w_2 \cdot \frac{1}{1 + \exp(-d_r)} \quad + w_3 \cdot \frac{1}{N} \sum_{n=1}^{N} \frac{1}{1 + \exp(-d_{e_n})} \tag{8.2}$$

8.3.3 Sampling of Simulation Input Parameters

Latin Hypercube Sampling (LHS) is a method that can be used to produce, in a distributed way, the set of input values to be used in the simulations according to the desired intervals (WANG, 2019). This statistical method generates a random sample of parameter values from distribution in multiple dimensions. LHS consists of subdividing the sample universe into several disjoint subsets and extracting a representative element for each subsets, chosen at random.

The generated and stored simulations used in the supervised learning model were carried out in packages (batches). Thus, for the same type of scenario, it was possible to vary some parameters at the beginning of the simulation, which may lead to different outcomes. The simulations parameters changed during the sampling were: (a) latitude and longitude, determining the initial positions of the agents around the CAP points (adopting fixed CAP positions), (b) the flight level blocks to determine their altitudes, (c) the commit distance (the minimum distance that an agent is from a possible target that allows it to leave the CAP tactic and commit), (d) the thresholds of the offense and vulnerability indices before and after firing a missile (represent the level of risk acceptance that the agent is willing to withstand), (e) the shot philosophy (orientation defined during mission planning, before the flight, referring to the moment when, within the Weapon Engagement Zone (WEZ), which is an estimation of the missile maximum launch range (DANTAS *et al.*, 2021b), the agent must launch a missile), (f) the shot distance (the minimum distance that an agent is from a possible target that allows, during an engagement, fire a missile, i.e., the WEZ), and (g) the presence or absence on the aircraft of a Radar Warning Receiver (RWR), an EW system that detects electromagnetic emissions from opposing radar systems.

Using the LHS algorithm, 3,729 constructive simulations were generated in ASA, and

a total of 10,316 engagements were observed. Each simulation corresponds to a 12-minute scenario executed three times faster than real-time, lasting approximately ten days in total.

The scenario consists of two opposing formations with two aircraft each, which are initially approaching, disengaged, and outside the radar range of each other. Their main goal is to establish a CAP at the same CAP point, invariably leading to a confrontation. When they enter the limits of the opponents' radars, the engagement phase begins. In the modeling proposed in this work, each aircraft is equipped with four of the same type of medium-range missile, i.e., up to 40 nautical miles in range.

8.4 Methodology

After sampling the variables, respecting the intervals chosen for each one, we executed the simulations through ASA. The engagement events between the four agents are extracted with the data generated from these simulations. The average DCA index is calculated for all engagements extracted from the interval under analysis, generating the output variable that we intend to predict later for new samples. With the input data of the simulations and the output variable already defined, we build a supervised machine learning model based on eXtreme Gradient Boosting (XGBoost), predicting the average value of the agents' DCA index in future engagements. XGBoost represents a class of algorithms based on Decision Trees with Gradient Boosting (CHEN; GUESTRIN, 2016). Its performance is analyzed with the test dataset after the model's training process is completed. This section discusses the input and output model variables, preprocessing procedures, hyperparameters tunning, evaluation metrics, and cross-validation processes.

8.4.1 Model Input and Target Variables

The main variables that coordinate the simulation of a BVR air combat were analyzed, and seventeen input variables were determined to be the most important to define the progress of the simulations based on the described scenarios. In addition, there are categorical and numerical variables with different ranges of coverage, and the definition of these sampling intervals was made based on the operational knowledge of BVR pilots and combat specialists. Next, in Table 8.1, the description of each of the input variables of the simulations is carried out, presenting its unit when the variable is not dimensionless.

Concerning the target parameter, the DCA index will be averaged between the start of the agent's Commit maneuver and the beginning of either Break or Abort maneuvers since both will make the agent disengage. Therefore, given a sequence of input parameters

TABLE 8.1 – Variables at the beginning of the engagement.

Parameter	Description
distance [m]	Distance between the reference and the target
aspect [deg]	Angle between the longitudinal axis of the target (projected rearward) and the line-of-sight to the reference
delta_head [deg]	Angle between the longitudinal axis of both aircraft
delta_alt [m]	Difference of altitude between the reference and the target
delta_vel [kn]	Difference of absolute velocity between the reference and the target
wez_max_o2t [m]	Maximum range of the reference's weapon (non-maneuverable target)
wez_nez_o2t [m]	No-escape zone range of the reference's weapon (target performing high performance maneuver)
wez_max_t2o [m]	Estimated maximum range of the target's weapon (non-maneuverable reference)
wez_nez_t2o [m]	No-escape zone range of the target's weapon (reference performing high performance maneuver)
vul_thr_bef_shot	Level of risk acceptance before shooting
vul_thr_aft_shot	Level of risk acceptance after shooting
shot_point	Missile firing point between the maximum range and the no-escape zone range of the reference
rwr_warning	Boolean indicating whether the aircraft is equipped with an active RWR
hp_tgt_off	High priority target offense index of the reference
hp_thr_vul	High priority threat vulnerability index of the aircraft that is threatening the reference
own_shot_phi	Reference shot philosophy
enemy_shot_phi	Estimated enemy's shot philosophy

that define the agent's state, the model must predict the average value of the DCA index in that interval, improving the agent's situational awareness.

8.4.2 Data Preprocessing

Unlike what is done in artificial neural networks, there is no need to carry out data normalization procedures to employ the XGBoost algorithm since it is based on decision trees. Thus, using this type of learning method, the model benefits from one of the significant advantages of these trees in artificial intelligence problems related to the low amount of preprocessing required for it to be applied. It is necessary to transform the model's categorical variables into numerical ones to be appropriately processed. For this purpose, two preprocessing steps will be performed on the data: Label Encoding and One-Hot Encoding (COHEN *et al.*, 2013). Also, feature engineering will be carried out to facilitate the training process of the proposed regression model.

8.4.3 Hyperparameters Tuning

GridSearch is a hyperparameter adjustment process to determine the ideal values for a given model, based on searching throughout a grid (PUTATUNDA; RAMA, 2018). The performance of the entire model is based on the specified hyperparameter values. Some functions have been implemented, such as the GridSearchCV of the sklearn library, to automate finding the best of these values for the model. We performed an adjustment of the XGBoost model hyperparameters based on a variation of the library default values as observed in Table 8.2.

TABLE 8.2 – GridSearch parameters for the prediction model.

Parameters	**Values**
n_estimators	[100, 1000, 5000]
learning_rate	[2, 3, 6, 10, 15, 20]
max_depth	[0.1, 0.01, 0.001]
gamma	[0.0, 0.1, 0.2, 0.3, 0.4, 0.5]
subsample	[0.6, 0.7, 0.8, 0.9]
colsample_bytree	[0.6, 0.7, 0.8, 0.9]
reg_alpha	[0.001, 0.01, 0.1, 1, 10, 100]
min_child_weight	[1,3,5,7,9,10,13,15]

8.4.4 Metrics

The Root-Mean-Square Error (RMSE) is used to measure the differences between the values predicted by a model or an estimator concerning the actual values, providing the average magnitude of the error. As the errors are squared, the RMSE places a relatively high weight on significant errors, which means that RMSE should be most useful when large errors are particularly undesirable. In addition, the advantage of using the RMSE is that it has the same magnitude as the target variable, which helps in the interpretation of the average of the model errors found. In the analysis of the predictive models carried out in this work, the coefficient of determination (R^2) will also be used to evaluate the best architectures of the supervised machine learning models for BVR combat modeling. The use of R^2 is well established in classical regression analysis and it generally describes how the model is adapted to reality when making predictions.

8.4.5 Cross-Validation

After performing a dataset train-validation-test split, allocating 80% for the training and validation and 20% for the testing, we propose to conduct cross-validation to evaluate the model's ability to predict new data that was not used in the estimate with the benefits

of using the whole train dataset for training and validation and to highlight problems such as overfitting (HYNDMAN; KOEHLER, 2006). We employ 10-fold cross-validation to address this problem after considering the trade-off between processing time and the generalization of the model results.

8.5 Results and Analysis

This section presents a whole dataset exploratory data analysis to provide an initial understanding of the variables in the model, followed by 10-fold cross-validation results.

8.5.1 Exploratory Data Analysis

Exploratory data analysis began with a check of the main descriptive statistics of the model's input and output data. The input variables of the model follow a uniform distribution since they were sampled using LHS. The model's output variable, the DCA index, is a probability, and in this case, it ranges from a minimum value of 0.21 to a maximum value of 0.99. with a mean of 0.53 and a standard deviation of 0.12. Moreover, the mean and the median are almost the same (0.53 and 0.51), indicating a low number of outliers for this variable at the top of the distribution. A histogram and a boxplot were generated to visualize the distribution, as shown in Figure 8.3. The data of the target variable follows an approximately normal distribution around the average value.

Regarding the aspect and heading difference, these two angular variables are transformed into four numerical variables to calculate their respective sine and cosine values. Note that the variables referring to WEZ have minimum values of -1. These values are model adjustments for when it is not possible to estimate the WEZ.

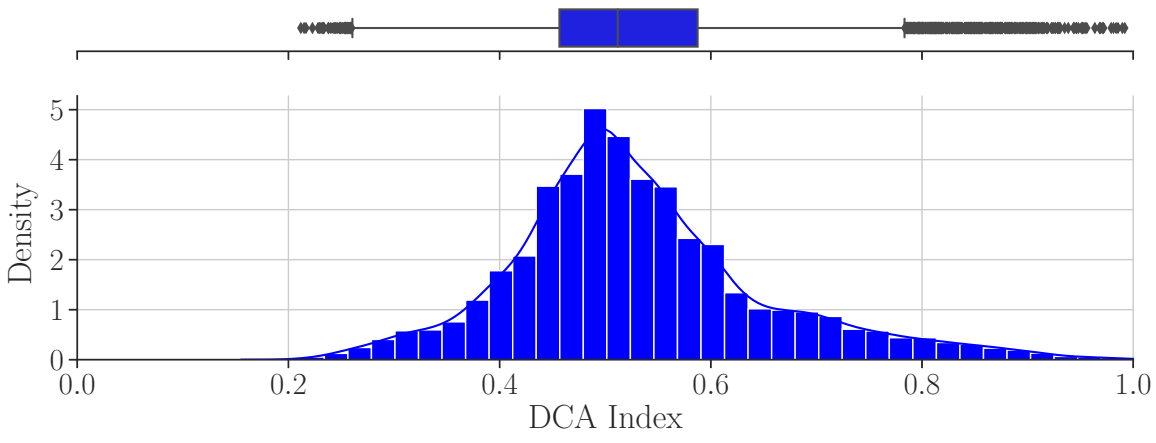


FIGURE 8.3 – Histogram and boxplot of the target variable.

8.5.2 Model Results

The means of all 10-fold metrics, namely RMSE and R^2, used to evaluate the best model at the end of the grid search training process after performing the cross-validation are, respectively, 0.0543 and 0.8020, while their standard deviations are 0.0009 and 0.0077. The coefficient of determination is approximately 80%, and an RMSE, which penalizes the effects of outliers, is close to 0.05. Considering that this is a regression problem and that we are trying to predict the DCA index, which indicates a probability of success in this type of mission, the results are satisfactory for this type of problem since they would be making predictions with errors in the range of 5% with a practically instantaneous inference time, which is desirable for a real-time application.

8.6 Outcomes

This work presented a supervised machine learning model through the XGBoost library to develop an engagement decision support tool for BVR air combat in DCA missions. The model represented some of the primary dynamics of the BVR combat, allowing the analyst to evaluate the parameters that influence this combat. Additionally, we analyze in advance the performance of an engagement made by a pilot in air combat through the agent's DCA index, predicting the outcome of this confrontation. This kind of prediction may be used as an innovative decision support system for the pilots in this air combat modality concerning whether to engage an opponent or not. Although the index does not inform which is the better action to take, it measures the performance of the actions taken by the agent through the proposed modelings.

The modeling considered the characteristics of the aircraft and their armaments, along with the beliefs about the opposing aircraft. In addition, we also used the shot philosophy for the aircraft and the pilot's level of risk aversion. The model showed an R^2 of 0.802 and an RMSE of 0.054. Assuming the average values of the DCA index as 0.53 and standard deviation of 0.12, the results showed relatively consistent values and good predictive power of the DCA index. With this degree of confidence in the model, it is possible to predict future pilot's conditions, even with a few samples. The regression model could calculate the average values of the DCA index in each engagement in a coherent manner, providing quick answers to the results of this air combat phase that could provide the pilot with an improvement in his situational awareness in real-time. Thus, through simulation data, it is possible to improve the employment of the best operational tactics for each situation in the complex context of BVR air combat. Furthermore, it contributes by avoiding the incorrect and careless use of weapons, improving DCA mission effectiveness. Finally, since the pilot's survival is a determining factor, it could decrease the number of friendly aircraft

lost in real-life BVR air combat.

Future work should move towards using a more significant number of simulations since we only analyzed 10,316 engagement cases due to the substantial computational cost to generate them. In addition, the search for more variables to define the agent state at the beginning of the engagement, or to do a feature engineering to generate new variables from the existing ones, could improve the performance of the proposed model. Also, comparing the results of XGBoost with other regression algorithms would be interesting to understand which supervised learning method best suits this type of problem. Finally, the conception of other operational metrics, like the DCA index, or the combination of several of them, to assess the level of performance in an engagement could contribute to bringing more information to the model.

8.7 Source Code

The source code used in this chapter is available at `https://github.com/jpadantas/lars2021`. This repository contains all the necessary code and detailed instructions to reproduce the results and experiments discussed.

9 Enhance Situational Awareness

Situational awareness is fundamental for achieving strategic superiority in air combat's dynamic and continually evolving landscape, particularly in BVR engagements. This chapter explores integrating simulation technologies and machine learning techniques to enhance situational awareness. As indicated in Figure 1.2, this chapter primarily contributes to the "In-Flight Tactical Systems" area within the proposed research framework.

This chapter is founded on the following work:

> **DANTAS, J. P. A.**; MAXIMO, M. R. O. A.; COSTA, A. N.; GERALDO, D.; YONEYAMA, T. Machine Learning to Improve Situational Awareness in Beyond Visual Range Air Combat. IEEE Latin America Transactions, v. 20, n. 8, 2022. Available at: `https://latamt.ieeer9.org/index.php/transactions/article/view/6530`.

9.1 Summary

This chapter presents an artificial intelligence model using artificial neural networks that provide parameters to improve the situational awareness of a Beyond Visual Range (BVR) air combat pilot. In this combat modality, it is necessary to make decisions based on information from sensors, mainly radars. Furthermore, since information regarding enemy aircraft systems is sometimes unknown, pilots' decisions are usually based on beliefs regarding the opponent. The presented model proposes to deal with such characteristics, generating behaviors for entities represented in a constructive simulation environment, i.e., simulated people operating simulated systems. We created BVR air combat simulations between two aircraft, with only one missile each, through Latin Hypercube Sampling (LHS) to choose input variables to cover almost homogeneously all their ranges. The aircraft have similar behaviors, and their parameters may change only at the beginning of the simulation. The simulation environment generated ten thousand air combat scenarios, varying thirty-six input parameters, for the analysis proposed in the case study. From this data, we could create supervised machine learning models that substantially improve

the BVR air combat pilot's situational awareness regarding offensive situations in which the reference aircraft employs a missile against a target or defensive positions, in contrast to when the same reference aircraft tries to avoid a possible enemy's missile launched in its direction. The offensive and defensive models were consistent with the accuracy of 0.930 and 0.924 and the F1-score of 0.717 and 0.678, respectively. Thus, the contribution of this work is to use machine learning algorithms to generate responses concerning the tactical state to improve the pilot's situational awareness and, therefore, the in-flight decision-making process.

9.2 Introduction

In modern air combat, Air Forces worldwide have been enhancing their aircraft and weaponry (HIGBY; COL, 2005). The development of more effective combat tactics and the constant training of pilots are indeed essential to ensure a nation's sovereignty. Modern air confrontations are categorized into visual combat, known as Within Visual Range (WVR), and beyond visual combat, known as Beyond Visual Range (BVR) (KURNIAWAN *et al.*, 2019). The latter, being the most critical part of the combat (YUAN *et al.*, 2016), involves combatants using airborne detection equipment to search for the enemy target and missiles that use both the launching aircraft's onboard radar and the missile's internal radar. BVR combat is characterized by its dynamism due to the influences that the actions of one aircraft have on others (MARQUES *et al.*, 2015).

Real BVR air combat training is quite costly. With an estimated cost of $4,700.00 per hour, the Swedish aircraft JAS 39 Gripen has a very favorable comparison to the Block 40/50 F-16, its closest competitor, which has an estimated hourly cost of $7,000.00 (IHS Jane's, 2012). Other aircraft such as the F-16, Rafale, Eurofighter, and F-35 have even higher costs, considering factors such as aviation fuel, consumables, operation and maintenance, labor, system upgrades, capital charges, depreciation, and amortization.

Therefore, the use of simulations, especially constructive ones—simulated agents in a simulated environment—has become a practice adopted by Armed Forces in various countries to reduce costs (COSTA *et al.*, 2022). Such simulations can be used for the analysis of operational scenarios, which can generate conclusions at strategic, operational, and tactical levels (BRUZZONE; MASSEI, 2017).

Based on the results of a large number of simulations, it is possible to propose models that describe the tactical situation of the pilot in flight (DANTAS *et al.*, 2021a). However, this process can be very time-consuming and costly, especially when it comes to high-fidelity simulations. This work proposes the use of a supervised machine learning model to provide a quick and reliable response regarding the tactical situation of a BVR pilot

without the need for new simulation executions. This model is based on the results of previously conducted simulations used to train, validate, and test the proposed model. Such responses can assist in the decision-making process of a BVR pilot in situations like the ideal missile launch moment or performing a defensive maneuver.

Various studies in the field focus on the decision-making processes of pilots, employing diverse methodologies to simulate and understand their behavior. For instance, agent-based models simulate interactions among multiple autonomous pilots, capturing complex dynamics and strategic behaviors in air combat (HEINZE *et al.*, 1998). Fuzzy logic has been applied to refine decision-making processes, providing a framework for handling the inherent uncertainties in air combat (AKABARI *et al.*, 2005; TRAN *et al.*, 2002; PRABHU *et al.*, 2014). Bayesian networks have been used to analyze and predict pilot decisions, leveraging probabilistic relationships to update and refine models as new information becomes available (POROPUDAS; VIRTANEN, 2007; FU *et al.*, 2021; RAO *et al.*, 2011a; DU; LIU, 2010). Influence diagrams provide a structured approach to sequential decision-making, allowing researchers to map out decision points, uncertainties, and objectives in a coherent model (LIN *et al.*, 2007).

Game theory continues to be fundamental in modeling air combat scenarios, capturing the strategic interplay between adversaries, and identifying optimal tactics under various conditions (KARELAHTI *et al.*, 2006; VIRTANEN *et al.*, 2006; MUKAI *et al.*, 2003; MA *et al.*, 2020; HA *et al.*, 2018). Reinforcement learning has been increasingly utilized to enhance pilot strategies through continuous learning from simulated experiences (HU *et al.*, 2021; TOUBMAN *et al.*, 2016; PIAO *et al.*, 2020; WEILIN *et al.*, 2018). Evolutionary algorithms have shown promise in optimizing pilot maneuvers and tactics, using principles of natural selection to iteratively evolve superior strategies (LI *et al.*, 2020; YANG *et al.*, 2020). The minimax method has been applied to develop strategies that are resilient under adversarial conditions, minimizing potential losses even in the worst-case scenarios (KANG *et al.*, 2019). Behavior trees have been employed to construct adaptable and hierarchical decision-making frameworks, enabling the decomposition of complex tasks into simpler, modular components (YAO *et al.*, 2015a).

When analyzing these works, no approaches using Artificial Neural Networks (ANNs) were identified for predicting the pilot's tactical situation and improving situational awareness. For this reason, and due to their versatility and efficiency in problems with many input variables (SILVA *et al.*, 2017), this work opted for this type of supervised machine learning algorithm to address the problem of BVR air combat.

Therefore, the main contributions of this work are: identifying the most important variables to be analyzed in a BVR combat that contribute to the analysis of similar problems in this context, even when dealing with data from other simulation platforms; proposing a methodology for the use of machine learning to create decision support tools

using data from simulations; and providing responses to the BVR pilot regarding their tactical situation using supervised learning techniques, specifically ANNs, which can enhance their situational awareness in air combat.

The remainder of this research is organized as follows: in Section 9.3, the methodology to train two ANN-based models, one for an attack situation and another for a defense situation, is presented; in Section 9.4, the main results and discussions regarding the proposed models are demonstrated; and in Section 9.5, the final considerations and suggestions for future work are presented.

9.3 Methodology

This section describes the BVR combat case study analyzed in this work, as well as the input variables used to generate the scenarios and the output variables obtained after the simulations were executed. Additionally, the use of artificial neural networks and the metrics used to improve the situational awareness of the BVR pilot are detailed.

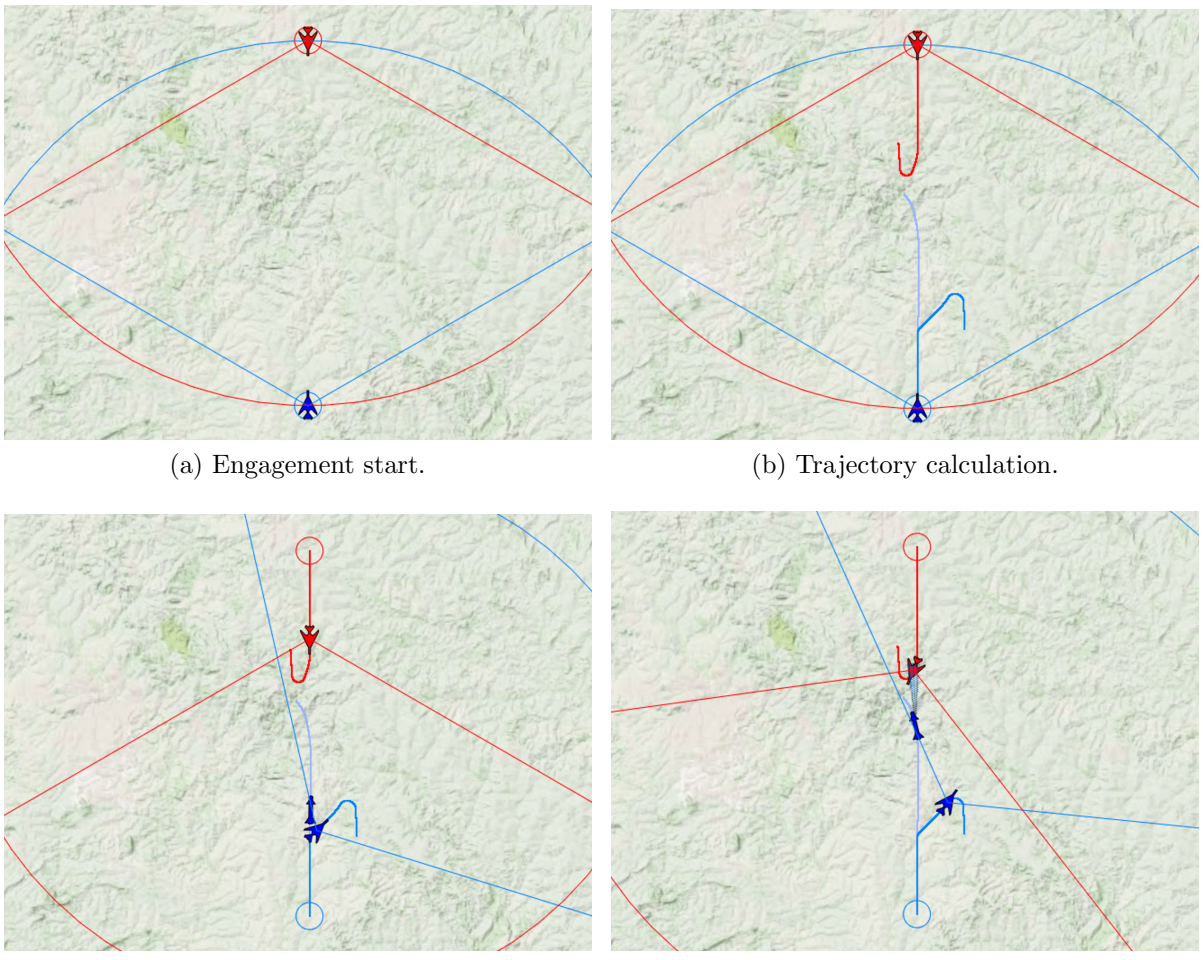
9.3.1 Case Study

To obtain the data used in this study, the prototype of the Aerospace Simulation Environment (ASA)* (DANTAS *et al.*, 2022a) was utilized, a simulation tool developed by the Brazilian Air Force (FAB) for operational scenario analysis. ASA currently has restricted access as it is a military-use platform incorporating high-fidelity models, with systems and subsystems present in military scenarios such as onboard radars with different search and tracking modes, active radar missiles, and aircraft modeled with 6 degrees of freedom. The simulation concept addressed in this context involves scenario elements represented by autonomous agents capable of making decisions based on artificial intelligence models. Such simulations, termed constructive, can be employed in decision-making to predict possible outcomes of engagements between opposing forces and aid in defining courses of action (COSTA, 2019).

Figure 9.1 describes the main phases of the case study scenario, which consists of a one-on-one BVR combat scenario where initially both aircraft are approaching, disengaged, and out of each other's radar range. Engagement begins when they enter the opposing aircraft's radar limits (Figure 9.1a). Figure 9.1b presents the trajectory calculation of both aircraft until the end of this engagement. In the proposed modeling in this chapter, a scenario is considered where each aircraft is equipped with a medium-range active radar missile. It is known that the use of a missile typically involves three phases (DANTAS,

*https://youtu.be/o3huZycjdDA

2018). Initially, when the enemy aircraft is within the effective engagement zone of the weapon, also known as the Weapon Engagement Zone (WEZ) (DANTAS *et al.*, 2021b), the launching aircraft fires and transmits the target information to the missile during the support phase (Figure 9.1c). In the second phase, the aircraft performs an evasive maneuver, *cranking*, and the missile continues towards the expected intercept point. Finally, the missile activates its own radar, becoming active, and attempts to reach the target's future position (Figure 9.1d). It is important to note that despite the relative simplicity of the proposed case study, this engagement can be considered the basic situation for constructing more complex air combat scenarios.



(a) Engagement start. (b) Trajectory calculation.

(c) Missile launch. (d) Disengagement maneuver.

FIGURE 9.1 – BVR combat engagement phases.

9.3.2 Input and Output Variables

The most important variables were determined to define the progress of the simulations in the proposed case study using the operational knowledge of FAB pilots. Additionally, from the variables described by the experts, variable selection techniques using mutual

information analysis were used to determine those utilized in this study. There are categorical and numerical variables with different coverage ranges. The simulator's definition of these operational ranges was based on discussions with BVR combat pilots and experts.

9.3.2.1 Relative Position

A numerical variable representing the positional relationship between the blue and red aircraft. This variable (`relative_position`) is defined in the range $[0, 60]^\circ$. The use of this range is based on the radial radar range limit, also defined as 60° in the simulator. The blue aircraft was used as a base for calculating the relative position for the proposed simulations. Therefore, the base aircraft was fixed, and the red aircraft varied its position from 0° to 60°, always pointing towards the line connecting the two aircraft.

9.3.2.2 Distance Between Aircraft

A numerical variable that depicts the relative distance between the two aircraft, calculated in nautical miles (NM). The range for this variable (`distance`) was defined between 25 NM and 50 NM.

9.3.2.3 Altitude

This is a very important numerical variable in BVR air combat, as it determines the amount of energy each aircraft possesses. A higher amount of energy allows for better performance in air combat (FEDERAL AVIATION ADMINISTRATION, 2021). The altitude range for the blue team (`altitude_blue`) and red team (`altitude_red`) was defined between $20,000$ ft and $35,000$ ft.

9.3.2.4 Speeds

A numerical variable that influences various aspects of BVR combat. Higher speeds can translate into more effective attacks and defenses but result in higher fuel consumption, reducing the aircraft's range. The speed range for the blue aircraft (`velocity_blue`) and red aircraft (`velocity_red`) was defined between Mach 0.8 and Mach 1.00.

9.3.2.5 Radar Warning Receiver (RWR)

A categorical variable that indicates the use of RWR, which is widely used in BVR combat and is based on detecting electromagnetic waves emitted by radar systems (PRABHU *et al.*, 2014). As the opposing aircraft are at a distance that does not allow for visual

identification, BVR combat relies heavily on information from these sensors. The variable takes the values 1 when the blue (`blue_rwr`) or red (`red_rwr`) aircraft has the operating equipment and 0 when inoperative. The presence of RWR greatly influences BVR combat as it enables the pilot to perform evasive maneuvers that ensure survival.

9.3.2.6 Missile

A categorical variable that represents the type of missile used in combat, which can be, for the blue and red aircraft, AM-3AL (`missile_blue_a` and `missile_red_a`), AM-3BL (`missile_blue_b` and `missile_red_b`), and AM-3CL (`missile_blue_c` and `missile_red_c`), listed in descending order of lethality and represented respectively by $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$. Variables related to the estimated type of missile that the enemy aircraft possesses will also be computed for both the blue (`missile_est_blue_a`, `missile_est_blue_b`, and `missile_est_blue_c`) and red (`missile_est_red_a`, `missile_est_red_b`, and `missile_est_red_c`) sides.

9.3.2.7 Shooting Philosophy

A categorical variable that represents the minimum distances at which a missile should be fired. These distances are defined based on the adopted doctrine and available resources since, if there is not a large availability of armaments, for example, they must only be used when there is a higher certainty of effectiveness (DANTAS, 2018). The shooting philosophy can be classified in increasing order of distance from the target as follows: short (`shotphi_blue_s` and `shotphi_red_s`), medium (`shotphi_blue_m` and `shotphi_red_m`), and long (`shotphi_blue_l` and `shotphi_red_l`). Variables related to the estimated shooting philosophy adopted by the enemy aircraft will also be computed for both the blue (`shotphi_est_blue_s`, `shotphi_est_blue_m`, and `shotphi_est_blue_l`) and red (`shotphi_est_red_s`, `shotphi_est_red_m`, and `shotphi_est_red_l`) sides.

9.3.2.8 Thresholds

Numerical variables that represent the pilot's risk aversion in different phases of the combat. Two values are considered: the first is used for the pre-launch phase (`threshold_bfr_blue` and `threshold_bfr_blue`), and the second for the post-launch phase (`threshold_aft_blue` and `threshold_aft_red`). This division was made to increase the pilot's risk propensity after making the shot, as they have already committed to the launch. These values should be calibrated to represent behaviors close to what is expected from a BVR combat pilot, considering the current doctrine and the political-strategic conditions in which the simulated combat is inserted.

9.3.2.9 Final Missile Distance to Target

Numerical variables refer to the shortest distances that the blue (`BM_RA`) and red (`RM_BA`) aircraft missiles could reach during the simulation. Taking the blue aircraft as a reference, the `BM_RA` distance represents the attack situation, as it refers to the minimum distance reached by its missile to the target. On the other hand, the `RM_BA` distance refers to the defense situation, being the minimum distance reached between the missile launched by the red aircraft and the reference (blue aircraft). These parameters allow for analyzing the mission's success in terms of offensiveness and survival. These distances relate to how much the BVR pilot has strived to ensure a successful attack or defense. By obtaining the minimum distances reached by the missiles from the simulations conducted, a filter is used to classify the launches that were successful or not. The minimum distance of 10 meters was standardized as the minimum operational distance of the modeled missile's proximity fuse, considering the type of mission to be accomplished. This distance is important to determine whether the missile will effectively fulfill its objective of hitting the desired target. Defining a distance limit to determine the success or failure of a mission substantially increases the number of scenario analysis possibilities. If the missile is intended to function merely as a warning and change the enemy aircraft's trajectory, reaching a distance of 10 kilometers, for instance, would likely be sufficient to achieve the desired effect. This work focuses only on the case of the missile neutralizing its target. The binary variables of the attack and defense models that define the success or failure of the missiles launched toward their targets are the outputs of the models and the parameters to be predicted.

Therefore, based on the described variables, 36 predictor attributes and 1 output variable were defined for each of the proposed combat situations (attack and defense) to be used in the machine learning models.

9.3.3 Sampling and Simulation

Using the simulator in a non-automated manner becomes unfeasible for generating a large number of input samples for the BVR combat simulations. To make the process more efficient in terms of processing time, a data generation method was adopted to cover all the ranges described in subsection 9.3.2 so that the chosen values for the variables were appropriately distributed. This method is called Latin Hypercube Sampling (LHS) (MCKAY *et al.*, 1979).

Using the LHS algorithm, $10,000$ different configurations for the input variables of the simulations were generated, one for each simulation, and with the use of the BVR air combat simulator, it was possible to generate, after seventeen days, $10,000$ simulations.

From the input variables and the results obtained in these simulations, it was possible to create a supervised machine learning model based on ANN to provide a quicker response with a good level of confidence in the simulation results.

9.3.4 Improvement of Situational Awareness

Situational Awareness (SA) is defined as the perception of elements in an environment within a volume of time and space, the comprehension of their meanings, and the projection of their status in the near future (ENDSLEY, 1995). The importance of situational awareness, in the operational aspect, emerged from the operators and pilots themselves, who realized how difficult it is to achieve and maintain it at a high level, a central aspect in their performances and mission fulfillment (ENDSLEY, 1995).

Given the importance of high situational awareness for the BVR air combat pilot, the creation of a means to improve it during combat is proposed. It is possible to obtain a quicker response from the BVR simulator with information regarding the possible final distances that the missiles launched by the enemy aircraft and the own aircraft reached from their respective targets. Therefore, based on these distance values, it is possible to constantly predict the final situation of the BVR air combat between a friendly aircraft and an enemy aircraft, both equipped with only one missile. The information related to these distances increases the pilot's situational awareness by providing a prediction of the final outcome of an air combat.

The implementation of an ANN model requires the definition of input and output variables, the choice of the architecture to be employed, its topology, and the relevant training and testing to perform supervised learning.

9.3.5 ANN Models

The two classification ANNs, referring to the attack and defense scenarios, have the same architecture with an input layer with thirty-six neurons ($I_n, n \in Z : 1 \leq n \leq 36$), which are related to the predictor attributes, two hidden layers, which according to Lippmann (1988) are sufficient to create classification regions of any desired shape, with eighteen and nine neurons respectively ($H_{1,18}$ and $H_{2,9}$), and an output layer with one neuron (O_a for the attack model and O_d for the defense model), as shown in Figure 9.2. The choice of the number of neurons in the hidden layers was made by testing different topology configurations, with the best performance achieved when successive halving of the number of neurons in the previous layer was used.

All layers have the Rectified Linear Unit (ReLU) activation function (GOODFELLOW *et al.*, 2016), except for the output layer which uses sigmoid for the desired binary classifi-

cation. The Dropout technique, which eliminates some neurons/weights in each iteration based on a probability of 20% (HINTON *et al.*, 2012), was also used to avoid overfitting during model training. The Adaptive Moment Estimation (Adam) optimizer was also employed, an extremely popular training algorithm for ANNs (BOCK; WEIß, 2019). Adam is a stochastic gradient descent method based on adaptive estimation of first and second-order moments (KINGMA; BA, 2014), which, in our case, aimed to minimize cross-entropy, monitored by an early stopping method responsible for verifying if the chosen validation set metric continued to improve throughout the training process (patience, i.e., the number of waiting periods before early stopping if no progress on the validation set, was set at 25 epochs, which proved adequate for the noise level in the model optimization process).

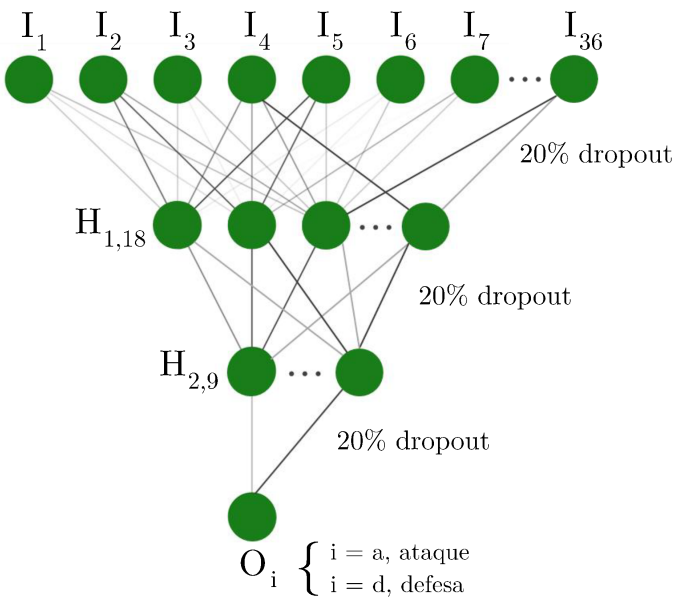


FIGURE 9.2 – Proposed ANN architecture for attack and defense classification models.

For the ANN analysis, proportions of 85% for training and validation and 15% for testing the 10,000 sample datasets were used, where each represents a simulation for the attack and defense scenarios simultaneously. In the training and validation data, k-fold cross-validation was used so that all observations were used for training and validation, avoiding possible occurrences of overfitting. According to Kuhn *et al.* (2013), the choice of the number of folds does not have a well-established formal rule, with a higher value of k leading to lower biases at the cost of increasing the computational time of the training. Therefore, $k = 5$ was used, considered a usual value in the literature and which presented good results. The performance metrics used to evaluate the classification ANNs were accuracy, precision, sensitivity, and F1-score.

9.4 Results and Discussions

This section demonstrates the problem data's correlation analysis and the proposed models' metrics.

9.4.1 Correlation

As shown in Table 9.1, due to the complexity of the BVR air combat problem, there are no input variables of the model that have a strong correlation (positive or negative) with the simulation output variables, except for those related to the presence or absence of RWR in the aircraft, which have slightly moderate negative correlation values (around -0.38) due to their influence on missile detection when active and directed towards the target.

TABLE 9.1 – Correlation between the model's input and output variables.

	BM_RA*	RM_BA†		BM_RA*	RM_BA†
relative_position	-0.028	-0.034	blue_rwr	-0.019	-0.377
distance	-0.103	-0.093	red_rwr	-0.376	0.005
altitude_blue	0.051	-0.029	shotphi_blue_s	-0.233	0.135
altitude_red	-0.032	0.058	shotphi_blue_m	0.005	0.016
velocity_blue	0.031	-0.002	shotphi_blue_l	0.226	-0.149
velocity_red	0.012	0.030	shotphi_est_red_s	0.002	0.062
missile_blue_a	-0.096	0.034	shotphi_est_red_m	0.003	-0.015
missile_blue_b	-0.070	0.021	shotphi_est_red_l	-0.005	-0.047
missile_blue_c	0.165	-0.055	shotphi_red_s	0.141	-0.234
missile_est_red_a	0.001	0.030	shotphi_red_m	0.018	0.009
missile_est_red_b	-0.004	0.007	shotphi_red_l	-0.158	0.223
missile_est_red_c	0.004	-0.037	shotphi_est_blue_s	0.068	-0.005
missile_red_a	0.032	-0.092	shotphi_est_blue_m	-0.005	0.014
missile_red_b	0.038	-0.073	shotphi_est_blue_l	-0.063	-0.010
missile_red_c	-0.070	0.164	threshold_bfr_blue	0.022	0.036
missile_est_blue_a	0.019	0.003	threshold_aft_blue	-0.012	0.029
missile_est_blue_b	0.020	-0.001	threshold_bfr_red	0.047	0.035
missile_est_blue_c	-0.039	-0.002	threshold_aft_red	0.039	-0.026

* Minimum distance reached between the blue missile and the red aircraft.
† Minimum distance reached between the red missile and the blue aircraft.

In light of these findings, the fact that it is a nonlinear complex modeling problem contributed to the choice of the ANN algorithm to represent this type of simulated operational scenario, as this type of supervised machine learning model can function as a universal function approximator (GOODFELLOW *et al.*, 2016).

9.4.2 Metrics

Using the standard value of 0.5 as the threshold for the sigmoid of the output layers of the two ANNs, we can observe in Table 9.2 the average accuracy results of the attack and defense models, which were 0.930 and 0.924, respectively. The dataset was observed to be unbalanced, with the minority class represented by successful missile attacks, accounting for 15% of the cases in both models. Thus, accuracy may not be the most appropriate metric for model evaluation (MENARDI; TORELLI, 2014). However, if the model always predicted the minority class, an accuracy of 0.851 and 0.848 would be obtained. Therefore, it is still possible to verify an improvement in this metric with the proposed model of 0.079 and 0.076. On the other hand, precision, sensitivity, and the harmonic mean of these two metrics (F1-score) can better evaluate this dataset (JENI *et al.*, 2013).

TABLE 9.2 – Metrics of the attack/defense models in each fold with their means and standard deviations.

	Accuracy	Precision	Sensitivity	F1-score
1st fold	0.931 / 0.925	0.735 / 0.735	0.707 / 0.639	0.721 / 0.683
2nd fold	0.925 / 0.919	0.686 / 0.735	0.745 / 0.565	0.714 / 0.639
3rd fold	0.931 / 0.926	0.749 / 0.730	0.681 / 0.655	0.713 / 0.696
4th fold	0.937 / 0.921	0.782 / 0.734	0.686 / 0.592	0.731 / 0.655
5th fold	0.927 / 0.931	0.712 / 0.751	0.697 / 0.681	0.704 / 0.714
Mean	0.930 / 0.924	0.733 / 0.737	0.703 / 0.628	0.717 / 0.678
Standard deviation	0.004 / 0.005	0.036 / 0.008	0.025 / 0.049	0.010 / 0.030

Precision averaged 0.733 and 0.737 for the attack and defense models, respectively. Low precision refers to cases where the models erroneously indicate that the missile would be successful if launched, which would not happen in reality. Analyzing in terms of associated combat costs, models with high precision are advisable to avoid unnecessary missile launches and maintain the aircraft's operational capacity, as there is a limit to the number of missiles that can be carried in air combat.

Sensitivity averaged 0.703 for the attack model and 0.628 for the defense model. Low sensitivity values are related to cases where the model suggests that the missile would not hit the target if launched in a given condition, recommending that the shot not be fired; however, in reality, that missile would have successfully hit the target if fired. Since target neutralization is very important in BVR combat, not launching a missile that would likely hit a specific target is a problem that should be avoided, highlighting the importance of having high sensitivity. It is observed that the defense model achieved a lower sensitivity than the attack model, demonstrating the difficulty of predicting the possible actions of the enemy aircraft in air combat.

The F1-score provides a broader synthesis of the model's performance. The values obtained were 0.717 and 0.678 for the attack and defense models, respectively, and seem

promising for analyzing this type of problem, considering the complexity of BVR combat. Finally, the standard deviation values of all metrics were very low, with a maximum coefficient of variation of 0.078, below the limit of 0.3 defined by Brown (1998), showing the low variability of results among the analyzed folds.

9.5 Outcomes

Analyzing the demonstrated application, the proposed model could represent the fundamental dynamics of BVR combat, allowing the analyst to evaluate the parameters that influence the outcome of the confrontation between the aircraft. Additionally, it is possible to provide the pilot with an estimate of their performance, in terms of attack or defense, in BVR combat. This perspective of how the combat is progressing and the quality of the actions taken is measured by calculating the distances reached by the missiles during the combat. The modeling considered both the characteristics of the aircraft and the weapons possessed, as well as beliefs about the opposing aircraft. We also analyzed the influence of the shooting philosophy established for each aircraft and the pilot's risk aversion level.

The ANN models were developed for the situation of the missile reaching minimum distances of 10 meters from the target, achieving accuracy values above 0.92 and F1-score values of 0.67 for the proposed supervised learning models. Thus, it is demonstrated that it is possible, through data obtained from constructive simulations, to develop decision support tools using ANNs that can improve the quality of flight in a BVR air combat. Such contributions help in the effectiveness of attack missions, ensuring that weapons are not used erroneously and carelessly, and aid in defense missions, where the pilot's survival is a determining factor, reducing the number of aircraft shot down in real combat.

For future work, the model could be applied to more complex scenarios with multiple aircraft and missiles, and its sensitivity and robustness could be further analyzed. Improving the simulator's efficiency would also help generate more data and samples. Deep learning techniques may enhance ANN performance, and new output variables or combinations could be explored to create metrics that better capture key aspects of the BVR scenario and support pilot situational awareness.

9.6 Source Code

The source code utilized in this chapter is available at `https://github.com/jpadantas/machine-learning-bvr-air-combat`. This repository includes all the scripts and instructions required to reproduce the experiments and results presented in this chapter.

10 Missile Hit Prediction

This chapter explores the methodologies and approaches used to predict missile hits in air combat scenarios, particularly in BVR air engagements. Accurately predicting missile hits is essential for enhancing situational decision-making in combat situations. Our discussion is based on contemporary research that leverages machine learning techniques to improve predictive accuracy and operational effectiveness. We also address the problem of unbalanced datasets and explore techniques to mitigate this issue. As indicated in Figure 1.2, this chapter primarily contributes to the "In-Flight Tactical Systems" area within the proposed research framework.

The insights and developments presented in this chapter are based on the following work:

> **DANTAS, J. P. A.**; COSTA, A. N.; MEDEIROS, F. L. L.; GERALDO, D.; MAXIMO, M. R. O. A.; YONEYAMA, T. Supervised Machine Learning for Effective Missile Launch Based on Beyond Visual Range Air Combat Simulations. In: Proceedings of the Winter Simulation Conference. Proceedings [...]. Singapore: IEEE, 2022. (WSC '22).

10.1 Summary

This work compares supervised machine learning methods using reliable data from beyond visual range air combat constructive simulations to estimate the most effective moment for launching missiles. We employed resampling techniques to improve the predictive model, and we could identify the remarkable performance of the models based on decision trees and the significant sensitivity of other algorithms. The models with the best f1-score brought values of 0.379 and 0.465 without and with the resampling technique, respectively, which is an increase of 22.69% and an appropriate time inference. Thus, if desirable, resampling techniques can improve the model's recall and f1-score with a slight decline in accuracy and precision. Therefore, through data obtained through constructive simulations, it is possible to develop decision support tools based on machine

learning models, which may improve the flight quality in BVR air combat, increasing the effectiveness of offensive missions to hit a particular target.

10.2 Introduction

In Beyond Visual Range (BVR) air combat, since pilots do not have visual contact with their opponents, the targets are detected using sensors, such as radar or Radar Warning Receiver (RWR). Lately, due to technological advances in sensors and weapons, BVR air combat has become one of the fundamental elements to achieving air superiority (HIGBY; COL, 2005). Constructive computer simulations have been widely used to emulate the most diverse BVR air combat situations to assess the effects of new combat tactics, sensors, and armaments, at a reduced cost compared to live exercises (DANTAS *et al.*, 2022a). In a constructive simulation, an aircraft is modeled as an autonomous agent, which is a computational entity that makes decisions based on data obtained from the environment in which it operates and interacts with other agents (RUSSELL; NORVIG, 2021). One of constructive BVR simulation's main challenges is mimicking the complex behaviors in all combat phases. A pilot can perform decision-making processes, such as adapting to new combat situations and conducting collective tactics with other aircraft (COSTA *et al.*, 2022). Some research addressed the application of artificial intelligence, game theory, and heuristics in modeling decision-making for autonomous agents in different phases of simulated BVR air combat.

Concerning threat assessment and target selection, some of the available methods found in the literature are: Bayesian optimization algorithm in Fu *et al.* (2021); back-propagation neural network in Yao (2021); combination of a genetic algorithm with deep learning in Li *et al.* (2020); zero-sum game in Ma *et al.* (2020); deep neural network in Dantas *et al.* (2021a); Case-Based Behavior Recognition proposed in Borck *et al.* (2015a); and an algorithm that combines fuzzy logic and Bayesian network in Rao *et al.* (2011a).

In the phase of selection and execution of tactical maneuvers, there are also many approaches in the literature: deep reinforcement learning in Hu *et al.* (2021); Bayesian networks in Du and Liu (2010); zero-sum differential game in Garcia *et al.* (2021); genetic algorithm in Kuroswiski (2020); reinforcement learning, artificial neural network and Markov decision processes in Piao *et al.* (2020); Hierarchical Multi-Objective Evolutionary Algorithm in Yang *et al.* (2020); decision algorithm based on the minimax method in Kang *et al.* (2019); Markov decision processes, reinforcement learning, a multilayer perceptron (MLP) neural network, and simulated annealing in Weilin *et al.* (2018); reasoning based on objectives in Floyd *et al.* (2017b); finite state machines, rule-based scripts, and reinforcement learning in Toubman *et al.* (2016); and grammatical evolution and behavior

trees in Yao *et al.* (2015a).

Our work addresses the decision-making process of estimating the most effective moment for firing a missile in BVR air combat in a specific scenario. The most decisive moment of firing guarantees the elimination of the target.

In Ha *et al.* (2018), the authors modeled the BVR air combat between two aircraft swarms as a zero-sum stochastic game. The firing moment estimation process was designed as a probabilistic function of the target's evasive ability, the missile's speed on the final approach, and the accuracy of the target's information to guide the missile. However, the methodology proposed has not been tested in simulations with higher fidelity concerning real BVR air combat. Also, significant simplifications were made in the process of modeling a BVR air combat: all aircraft in the same swarm fire at the same time, the missiles can pursue targets without the need for support from the radar aircraft that launched them, and the evasive ability of each aircraft is an input parameter.

In Dantas (2018), Dantas *et al.* (2022), the authors designed an MLP neural network using data from constructive simulations to be employed in an embedded device to enhance the pilot's situational awareness in the in-flight decision-making process. Although bringing promising results, only a single machine learning method was employed to analyze a reduced number of air combat scenarios simulation runs with only one aircraft on each team. Lima-Filho *et al.* (2021) adopted a similar methodology but used data collected in real air combat exercises with computationally simulated shots.

The main contribution of our work is to compare the application of different machine learning methods to estimate the most effective moment for launching missiles during a BVR air combat based on data from constructive simulations, which are run through a commercial off-the-shelf framework. Since running the simulations is computationally demanding, machine learning methods can streamline the missile success predictions for real-time applications. Furthermore, it was observed that, during the simulations, the missiles could not hit their targets in most of the scenarios due to challenging shooting conditions within our experiment design, which led to an imbalanced dataset. Therefore, we employed resampling techniques to improve the predictive model, analyzing accuracy, precision, recall, and f1-score. To the best of our knowledge, this is the first work to address the imbalance in the missile launch results within air combat simulation.

The rest of this chapter is organized as follows. The definition of a simulated BVR air combat scenario is presented in Section 10.3. Section 10.4 describes the methodology proposed to solve the problem of estimating the best moment for firing missiles using machine learning models. The analysis of the results obtained with the application of the machine learning models is presented in Section 10.5. Finally, Section 10.6 describes the conclusions of this work and shares ideas for future work.

10.3 Scenario Description

The scenario is formed by two forces that engage each other in a tropical area. The blue side performs a Combat Air Patrol (CAP), classified as a Defensive Counter Air mission. In contrast, the red side takes part in a fighter sweep (Offensive Counter Air) to neutralize the blue patrol, opening space for other types of tasks to be performed later, such as an air-to-ground attack. The red side always employs a 4-aircraft squadron formed by multi-role fighters, while the blue side can have either a 4- or a 6-aircraft squadron, which may also vary between two fighter concepts (defined as type 1 and 2). The concepts represent different classes of fighters; type 1 is a smaller single-engine aircraft (fourth-generation), while type 2 is a larger double-engine stealth aircraft (fifth-generation). These entities were defined within a framework for composable simulations called FLAMES (TERNION, 2023), which provides several aspects of constructive simulation development and use, including customizable scenario creation, execution, visualization, and analysis, as well as interfaces to constructive, virtual, and live systems.

We modeled the BVR air combat behaviors of the aircraft, i.e., autonomous agents, through behavior trees structured within states of a finite state machine (COLLEDANCHISE; ÖGREN, 2020). These behaviors include CAP, target selection, engagement, evasion, and missile firing. Also, the aircraft is equipped with several subsystems: controllers, trackers, datalinks, fusers, radars, RWR, antennas, and weapon systems. We will focus on the ones directly related to the variables used in our experimental design.

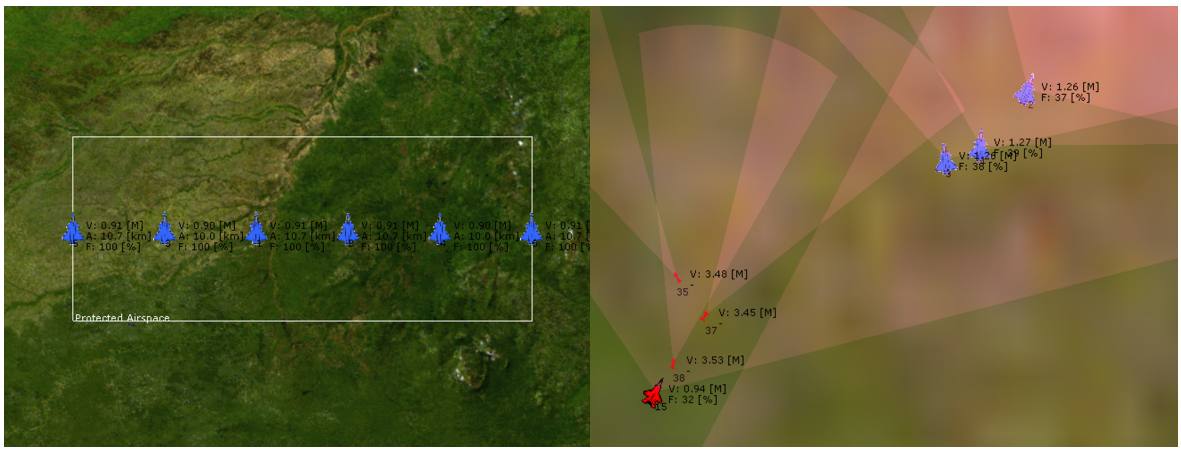
The red side employs a squadron formed by four twin-engine, mid-size fifth-generation fighters. These aircraft have internal weapons bays carrying up to 4 BVR air-to-air missiles (BVRAAMs). The blue side can have 4- and 6-aircraft squadrons of either type 1 or 2 aircraft. Type 1 is a light aircraft with a reduced cost, whereas type 2 is much larger. Both are single-engine multi-role fighters with stealth capabilities. The red and blue aircraft's main parameters can be found in Table 10.1.

TABLE 10.1 – Main characteristics of the red and blue (type 1 and 2) aircraft.

Feature	Red	Blue 1	Blue 2	Feature	Red	Blue 1	Blue 2
Wing area	49.5 m^2	36 m^2	43 m^2	Mission fuel weight	6000 kg	3070 kg	5200 kg
Wing span	11.52 m	8.9 m	9.5 m	Maximum weight	26205 kg	13300 kg	22214 kg
Wing incidence	1°	0°	1°	Dry weight	17805 kg	9085 kg	14638 kg
Wing 25% sweep	27°	38.4°	42°	Maximum payload	8250 kg	6000 kg	8000 kg
Wing dihedral	-4°	0°	-2°	Radius of action	648 NM	1000 NM	700 NM
Wing twist	2.5°	2°	2.5°	Cruise altitude	11000 m	10668 m	11000 m
Length fuselage	17.28 m	15 m	18.96 m	Cruise Mach	0.8	0.9	0.8
Aspect ratio	2.68	2.2	2.1	Max Mach (sea level)	1.1	1.1	1.1
Taper ratio	0.21	0.2	0.2	Thrust dry	59038 N	64874 N	109642 N
Ultimate load factor	13.5	14.5	13.5	Thrust aferburner	91781 N	98022 N	173550 N
Number of missiles	4	3	6	Front RCS	-10 dBm^2	-10 dBm^2	-25 dBm^2

Even though each aircraft type had its own baseline front Radar Cross Section (RCS), an additive factor in dBm^2 was included throughout the design of experiments (DoE) so

that different RCSs could be evaluated. Even though just the frontal value was provided, the RCS for each aircraft was determined by a table with values at each angle. Therefore, we applied the additive factor to all values within these tables. These aircraft types are both equipped with BVRAAM missiles with similar characteristics to those employed by the red side. However, different from what was modeled for the red missiles, the activation distance of the blue missiles varied to assess whether this would influence the shot's effectiveness. Additionally, during the different experiment cases, we changed the initial formation of the blue and red aircraft (distance between their CAP points, as in Figure 10.1a). Moreover, in the case of blue aircraft, we modified the radar ranges and the shot philosophy within their behavior trees. We altered flying altitudes and speeds according to the DoE. Figure 10.1b shows a representation of the radars and missiles employed during a simulation execution.



(a) Initial setting. (b) Intermediate frame.

FIGURE 10.1 – Scenario depictions with (a) 6 blue aircraft in wall formation within protected airspace and (b) radar and missile representations during the simulation execution.

The engagement distances for the aircraft's missiles are dynamically calculated based on factors such as the launcher's altitude and speed and the target's altitude, speed, and azimuth. Therefore, the aircraft's artificial pilot may decide whether to fire or not. The shot philosophy is defined as a percentage of the Weapon Engagement Zone (WEZ), calculated by the earlier parameters, stored on a table. For the red aircraft, this percentage is fixed at 60%. In simple terms, the WEZ represents the range of a given weapon to be employed in a particular shooting stance (DANTAS *et al.*, 2021b).

Besides this percentage, the behavior considers other factors to choose to fire, such as whether the target is already engaged by another friendly aircraft. Additionally, it defines defensive maneuvers and any other action performed by the aircraft. The data that feed the behavior trees come mainly from the aircraft's sensors. The most important ones are the RWR and the radar, which provide the unit's tracklist, i.e., the combined list containing all its possible targets.

The radar could operate in two distinct modes: scan and track. In the scan mode, the radar can cover larger areas with a smaller range. This is used to detect the aircraft within the theater of operations. Diversely, the track mode focuses on a particular target (or set of targets) to follow its movements and prepare to fire. The track mode range is usually larger than the scan mode, which was therefore considered 60% of the track mode range. While tracking, the radar power is concentrated on more specific regions, providing more extensive ranges at the expense of situational awareness concerning the surrounding areas.

10.4 Methodology

We model the estimation of the most effective moment for firing missiles in BVR air combat simulations as a classification problem, a classic machine learning challenge. This problem consists of classifying a set of instances of features, independent variables, into two classes defined by a single dependent variable: value 1 indicating an effective shot; and value 0 meaning an ineffective shot. We employed some of the most relevant supervised machine learning methods to estimate the most effective moment for firing missiles in BVR air combat simulations: Logistic Regression (LR), K-Nearest Neighbors (KNN), Support Vector Machines (SVM), Artificial Neural Networks (ANN), Naive Bayes (NB), Random Forest (RF), and Extreme Gradient Boosting (XGBoost); to the interested reader, we refer to Geron (2019). Concerning the resampling techniques, we applied oversampling methods such as Synthetic Minority Oversampling Technique (SMOTE) (CHAWLA *et al.*, 2002) and Adaptive Synthetic Sampling Approach (ADASYN) (HE *et al.*, 2008). Also, we introduced undersampling methods such as Tomek Links (TL) (TOMEK, 1976) and Edited Nearest Neighbor (ENN) (WILSON, 1972). Besides, we analyzed the datasets using hybrid techniques that use oversampling and undersampling together: SMOTE with Tomek Links (SMOTE-TL) (BATISTA *et al.*, 2004) and SMOTE with Edited Nearest Neighbor (SMOTE-ENN) (BATISTA *et al.*, 2004).

We proposed the following steps to solve this problem: design of a set of BVR air combat experiments (Subsection 10.4.1); acquisition of the shooting dataset, i.e., the training dataset, collected by performing the set of simulations (Subsection 10.4.2); specifying the variables used in the predictive models (Subsection 10.4.3); hyperparameters tuning process of all supervised machine learning models applied to the problem (Subsection 10.4.4); preprocessing of the training dataset (Subsection 10.4.5); and training and evaluation process of the machine learning models (Subsection 10.4.6).

10.4.1 Design of Experiments

Considering these scenarios and behavior characteristics, we create an experimental design using the Latin Hypercube Sampling (LHS) (MCKAY *et al.*, 1979). This method is an alternative to using factorial strategies, presenting a solution for filling the sample space in a better fashion than what is done by a purely random process, such as the Monte Carlo sampling (HUSSLAGE *et al.*, 2011). LHS divides the space into a prespecified number of sections from which it randomly samples one point. The variables in Table 10.2 form the sample space in this work, and each has a predefined interval to design the samples properly. This sampling method allows the user to specify the number of sample points desirable. In our work, we selected 240 cases as simulation inputs, each executed 30 times with different random seeds, which mainly interfered with missile probability of kill (set as 90% for both sides) and behavior evaluation delays since both were functions of these seeds. This probability value means that if the missile can reach a distance of at least 10 meters from the target, it has a 90% chance of destroying the target.

TABLE 10.2 – Model input and output variables.

Description	Min	Max	Unit
Blue maneuvering altitude	27.5	42.5	kft
Red maneuvering altitude	27.5	42.5	kft
Blue maneuvering speed	0.9	1.5	Mach
Red maneuvering speed	0.9	1.5	Mach
Blue radar track range	150	300	km
Blue missile range multiplicative factor	1	2	-
Blue RCS	-10	10	dBm^2
Blue missile activation distance	15	30	km
Blue shot philosophy (in percentage of WEZ)	50	70	%
Initial distance between blue aircraft (in longitude)	0.1	1.0	°
Initial distance between red aircraft (in longitude)	0.1	1.0	°
Blue CAP speed	0.7	0.75	Mach
Red CAP speed	0.7	0.75	Mach
Blue aircraft concept (type)	1	2	-
Whether there are six blue aircraft (0) or not (1)	0	1	-

10.4.2 Simulation Results

With the 240 cases defined by the LHS, 7,200 simulations were carried out, generating the data reports for analysis. Each run stopped after all aircraft had been destroyed or a simulation time of 30 minutes had been reached, which is a regular time for BVR air combat engagements. Since the simulations ran faster than in real-time, these 30 minutes took only an average of 5 seconds to be run, totaling 10 hours for the whole experiment. The first dataset generated concerns the number of surviving aircraft after each run. It

also contained the total number of missiles fired by the design case but no data concerning the firing conditions. In the second dataset, on the other hand, the focus was primarily on the firing conditions, presenting the positions of the shooter and the target at the moment of launch and the distance and the off-boresight angle between them (line of sight). Since the models used for the simulation are still restricted, the resulting datasets are also subject to restraints that preclude us from sharing them.

10.4.3 Variables

Using mutual information for selecting features in supervised models (BATTITI, 1994), we chose 12 variables from the 28 initial variables in the dataset after considering their distribution, correlation analysis, and operational importance as input and output variables of the proposed models. Table 10.3 describes the 11 features and 1 target variable used in this work's supervised machine learning models, considering the moment that the shooter aircraft launches the missile towards the target aircraft.

TABLE 10.3 – Description of the model's input and output variables.

Variables	Unit	Description
`radar_track_range`	km	Radar track range
`distance`	km	Distance between aircraft
`missile_act_dist`	km	Missile activation distance
`delta_altitude`	m	Difference between aircraft altitudes
`delta_speed`	kt	Difference between aircraft speeds
`missile_range`	-	Missile range multiplicative factor
`rcs`	m^2	Radar cross section
`firerange`	%	Percentage of WEZ
`angle_uni_to_tgt`	°	Off-boresight angle between aircraft
`delta_heading`	°	Difference between aircraft headings
`concept`	-	Which blue aircraft concept type was employed (1 or 2)
`kill`	-	Whether the target aircraft was killed by the missile

10.4.4 Hypeparameters Tuning

We created all the models using the Scikit-learn library, except the XGBoost model that has its library (CHEN; GUESTRIN, 2016), and, unless otherwise stated, with its default settings for their hyperparameters (PEDREGOSA *et al.*, 2011a). We perform the grid search algorithm as the technique for tunning all models since we want to use our experience from previous works to find the best set of values, even though it is well-known that random search may avoid the drawbacks of regular intervals (BERGSTRA; BENGIO, 2012).

The LR model is using the default Scikit-learn hyperparameters except for `C` = 100, which is the inverse of regularization strength, after performing a grid search exploring in

$\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$.

Regarding the KNN model, we chose 12 for its hyperparameter `n_neighbors`, which is the number of neighbors, after checking different values (from 1 to 50) and keeping track of the error rate for each of these models to find the minimum value.

Concerning the SVM model, we searched for `C`, which defines the amount of violation of the margin allowed, in $\{0.1, 1, 10, 100, 1000\}$ and `gamma`, a parameter that must be specified to the learning algorithm, in $\{1, 0.1, 0.01, 0.001, 0.0001\}$, resulting in `C` $= 10$ and `gamma` $= 0.1$ as the best set.

In the ANN model, we explored `learning_rate_init`, the initial learning rate, in $\{0.001, 0.01, 0.1\}$, `activation`, representing the activation function for the hidden layer, in $\{$`logistic`, `tanh`, `relu`$\}$, `solver`, which is the solver for weight optimization, in $\{$`sgd`, `adam`$\}$, and `alpha`, the strength of the L2 regularization term, in $\{0.0001, 0.001\}$. After that, we found `learning_rate_init` $= 0.001$, `activation` $=$ `relu`, `solver` $=$ `adam` and `alpha` $=$ `0.001` as the best set.

The NB model is operating the default except for `var_smoothing` $= 0.002$, which represents the part of the most significant variance of all features that is added to variances for calculation stability, found after running the grid search trying 100 different values spaced evenly on a log scale from 10^{0} to 10^{-9}.

In the RF model, we searched for `max_features`, the number of features to consider when looking for the best split in the model, in $\{2, 3, 4, 5, 6, 7, 8, 9, 10\}$, `min_samples_leaf`, the minimum number of samples required to be at a leaf node, in $\{3, 5, 8\}$, and `min_samples_split`, the minimum number of samples required to split an internal node, in $\{4, 8, 12\}$. The best hyperparameters set found was `max_features` $= 5$, `min_samples_leaf` $= 8$, and `min_samples_split` $= 4$.

In the XGBoost model, we inspect the best hyperparameters between `gamma`, the minimum loss reduction required to make a further partition on a leaf node, in $\{0.5, 1, 1.5\}$, `subsample`, the subsample ratio of the training instances, in $\{0.6, 0.8\}$, `colsample_bytree`, the subsample ratio of columns when constructing each tree, in $\{0.6, 0.8, 1.0\}$, and `max_depth`, the maximum depth of a tree, in $\{3, 4, 5\}$. The best set was `gamma` $= 1.5$, `subsample` $= 0.8$, `colsample_bytree` $= 0.8$ and `max_depth` $= 5$.

After finding each model's best set of hyperparameters, we applied all of the resampling techniques in the dataset using the Imbalanced-learn library (LEMAÎTRE *et al.*, 2017) with its default hyperparameters.

10.4.5 Preprocessing

The main goal of standardizing features is to help the convergence of the technique used for optimization (LAURENT *et al.*, 2016), which we perform to use the algorithms SVM, ANN, KNN, and NB. Data scaling was employed to equally distribute the importance of each input in the learning process (PRIDDY; KELLER, 2005). Standardization is not required for LR, RF, and XGBoost models. Besides, we performed Exploratory Data Analysis (EDA) to identify the general data behaviors. The methods employed in this analysis were correlation and descriptive statistics analysis.

The effectiveness of the training phase of machine learning models can deteriorate if the training dataset is imbalanced, i.e., if there is not an adequate number of samples relating the values of the features for each of the classes (target values). Through the EDA, we verified this problem, as there were many more samples of the occurrence of shots belonging to class `NO KILL`, indicating the non-elimination of the target. Thus, we used resampling methods to increase the number of class `KILL` samples.

10.4.6 Models Training and Evaluation Process

Before performing any training process, we conducted a train-validate-test split, randomly separating the data, allocating 85% for training and validation using a 5-fold cross-validation technique and 15% for testing. The test dataset will allow the evaluation of the machine learning models later. We try every combination of values from the grid search proposed, calculating the performance metric f1-score using 5-fold cross-validation. The grid point that best fits the dataset is the optimal combination of the hyperparameters. After running all the algorithms with the imbalanced training-validate dataset and evaluating it, we perform resampling techniques to improve the number of samples of the minority class. We again employed all the supervised learning algorithms using their best set of hyperparameters with these new balanced datasets. As the model deals with a classification problem, we evaluated the model by observing the metrics accuracy, precision, recall, and f1-score (GERON, 2019). The metrics results were acquired by analyzing the models' predictions and actual values in the test dataset.

10.5 Results and Analysis

This section examines the exploratory data analysis, the test dataset metrics, and time inference of the proposed supervised machine learning algorithms with and without resampling techniques.

10.5.1 Exploratory Data Analysis

Table 10.4 shows an overview of the descriptive statistics of these model's inputs numerical variables. All the model's input variables follow a uniform distribution since these variables were sampled using the LHS.

TABLE 10.4 – Descriptive statistics of the model's input numerical variables.

Variables	Mean	Std	Min	25%	Median	75%	Max
`radar_track_range`	217,375.7	40,219.6	150,289.2	182,961.3	214,907.9	251,175.8	285,788.2
`distance`	40,841.9	16,529.9	3,023.6	29,319.6	38,078.3	50,553.9	100,717.4
`missile_act_dist`	22,442.6	4,353.8	15,197.3	18,742.0	22,735.8	26,224.3	29,828.1
`delta_altitude`	768.6	2,357.6	-14,168.3	-655.3	641.8	2,284.2	13,573.3
`delta_speed`	12.2	71.7	-232.4	-38.0	16.6	63.1	263.3
`missile_range`	1.5	0.3	1.0	1.2	1.5	1.8	2.0
`rcs`	-0.1	5.8	-9.9	-5.3	-0.1	5.0	9.7
`firerange`	60.2	5.7	50.2	55.4	60.6	65.2	69.9
`angle_uni_to_tgt`	-0.8	21.8	-44.9	-9.0	0.0	4.6	44.9
`delta_heading`	181.9	77.1	-42.8	140.5	180.0	220.0	401.5

Remember that one of the features (`concept`) and the target variable (`kill`) are categorical. The distribution of the input variable `concept` is almost equal, presenting 67,238 (43.78%) samples as `Type 1` and 86,368 (56.22%) samples as `Type 2`. On the other hand, the output variable `kill` is very imbalanced, presenting only 18,397 (11.98%) samples as `KILL` and 135,209 (88.02%) samples as `NO KILL`. This imbalance is due to the fact that many of the 240 simulation cases have generated challenging shooting conditions, which degraded the missile envelope. These conditions could be aircraft with large altitude and speed differences or low aircraft RCS coupled with large missile activation distances, among others.

Pearson's correlation analysis of the variables can be seen in the correlation matrix represented in Figure 10.2. Notice that none of the model's features has a strong correlation with each other, with the largest absolute value being only 0.33 between `delta_heading` and `angle_uni_to_tgt`, and 0.31 between `missile_range` and `distance`. The algorithm's performance may deteriorate if two or more variables are tightly related, called multicollinearity (DANTAS *et al.*, 2021b). We may also be interested in the correlation between input variables with the output variable (`kill`) to provide insight into which variables may or may not be relevant as input for developing a model. Only the variable `distance` has a slight correlation (-0.22) with the target variable. The missile launch analysis is a complex non-linear problem that depends on multiple variables with a low correlation that corroborates the application of supervised machine learning algorithms since they may relate to all these variables.

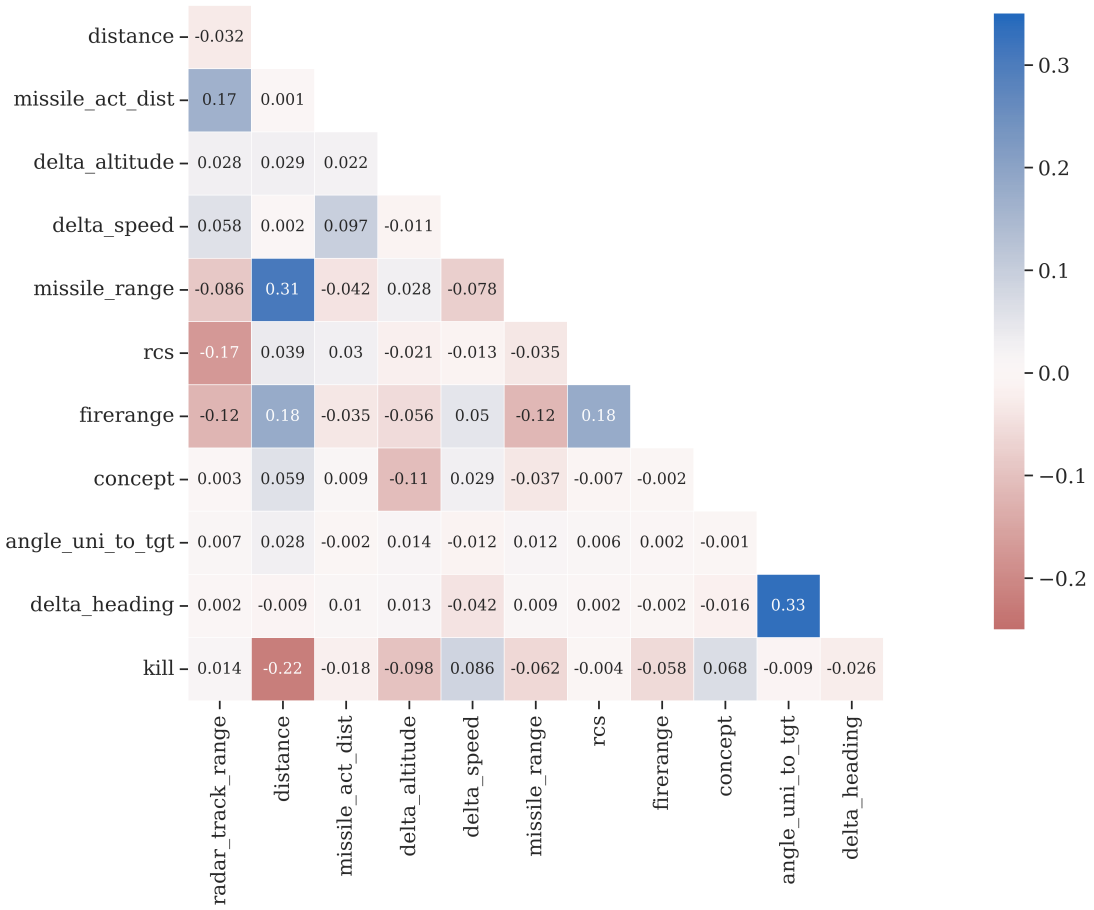


FIGURE 10.2 – Pearson correlation matrix of all model variables.

10.5.2 Models Metrics

Table 10.5 shows the metrics, accuracy (ACC), precision (PREC), recall (REC) and F1-score (F1), and the inference time (IT) in milliseconds (ms) obtained after evaluating the test dataset on the supervised machine learning models using or not resampling techniques. Regarding the accuracy, we expect at least metrics better than 0.880, which is the proportion of the majority class in the imbalanced dataset. If the model consistently predicted the majority class, the accuracy would be 0.880. Thus, accuracy may not be the most appropriate metric for evaluating the models (MENARDI; TORELLI, 2014). However, we notice a slight improvement in all models without resampling, except the LR model, with the RF models showing the best results with a low inference time regarding real-time applications (160.8 ms to predict the test dataset), as expected since this technique is quite suitable for imbalanced data (KHOSHGOFTAAR *et al.*, 2007). Precision, recall, and f1-score can better assess this problem (JENI *et al.*, 2013).

In general, the resampling techniques such as oversampling methods or hybrid approaches, using oversampling and undersampling techniques together, increased recall and f1-score with a slight decline in the accuracy and precision. Thus, considering it is

an imbalanced dataset, each supervised machine learning model and the corresponding resampling technique must be applied in different situations, depending on the interest of having a greater precision or recall, which depends on the actual application of that model.

Low precision refers to cases where models are erroneously reporting that the missile would succeed if launched, but that does not happen in reality. Regarding costs associated with modern air combat, models with high precision avoid unnecessary missile launches and help maintain the aircraft's capability since there is a limit to the number of missiles to be loaded on the aircraft in a regular air combat operation.

Low recall refers to cases in which the model suggests that the missile would not hit the target aircraft if launched in a given condition, recommending not fire at that moment, though, in reality, that missile would have hit the target aircraft successfully if launched. As target neutralization is a critical factor in BVR air combat, not launching a missile that would probably hit a particular target is a massive issue that good pilots try to avoid, which shows the importance of having a high recall. The F1-score, the harmonic mean of precision and sensitivity, synthesizes the model's performance more broadly. The model using RF and SMOTE obtained the best f1-score overall with low inference time as well (108.3 ms), compared to the worst inference time from the model using KNN and SMOTE (5194.4 ms).

Therefore, we observe that the SVM model without resampling techniques presents the best precision results (0.716), even though it has one of the worst recall metrics. XGBoost and RF models also offer good precision, 0.648 and 0.686 respectively, and low recall, both around 0.250. Considering the analysis of the recall metric itself, in all the models, ADASYN and SMOTE-ENN brought the best increase in this metric. In ANN models, which showed the highest recall value, the ADASYN technique allowed a 224.30% increase in the recall, going from 0.251 to 0.814. In the LR models, the increase was from almost null recall values to values that reached 0.711 using SMOTE-ENN. Thus, resampling techniques demonstrated the capacity to increase the recall metric, even with a slight decrease in accuracy and precision.

10.6 Outcomes

This work compares the application of supervised machine learning methods, using resampling techniques to improve the predictive model, on data from air combat constructive simulations to improve the effectiveness of missile launching, and demonstrates the differences in the performance of these several different models.

Without resampling, XGboost and RF brought the most consistent results considering

TABLE 10.5 – Supervised learning classification models metrics and inference time.

MODEL	ACC	PREC	REC	F1	IT[ms]	MODEL	ACC	PREC	REC	F1	IT[ms]
LR	0.877	0.421	0.003	0.006	1.4	ANN + SMOTE-TL	0.757	0.300	0.744	0.428	157.0
LR + SMOTE	0.655	0.215	0.685	0.327	1.1	ANN + SMOTE-ENN	0.730	0.283	0.784	0.416	35.7
LR + ADASYN	0.646	0.212	0.694	0.325	1.3	NB	0.884	0.672	0.096	0.168	4.3
LR + SMOTE-TL	0.661	0.217	0.676	0.328	1.3	NB + SMOTE	0.641	0.208	0.690	0.320	5.0
LR + SMOTE-ENN	0.625	0.204	0.711	0.317	1.1	NB + ADASYN	0.599	0.196	0.736	0.310	3.3
KNN	0.889	0.639	0.208	0.314	3903.1	NB + SMOTE-TL	0.641	0.208	0.691	0.320	3.3
KNN + SMOTE	0.763	0.296	0.678	0.412	5194.4	NB + SMOTE-ENN	0.590	0.194	0.743	0.307	3.2
KNN + ADASYN	0.730	0.272	0.721	0.395	4970.0	RF	0.895	0.686	0.262	**0.379**	**160.8**
KNN + SMOTE-TL	0.763	0.297	0.685	0.414	5090.1	RF + SMOTE	0.851	0.415	0.528	**0.465**	**108.3**
KNN + SMOTE-ENN	0.725	0.273	0.750	0.400	4632.8	RF + ADASYN	0.844	0.401	0.551	0.464	175.8
SVM	0.891	0.716	0.185	0.294	266.5	RF + SMOTE-TL	0.848	0.408	0.537	0.463	350.0
SVM + SMOTE	0.766	0.308	0.729	0.433	654.7	RF + SMOTE-ENN	0.809	0.351	0.664	0.459	35.2
SVM + ADASYN	0.722	0.276	0.785	0.409	752.8	XGBoost	0.892	0.648	0.255	0.366	17.1
SVM + SMOTE-TL	0.766	0.307	0.727	0.432	614.5	XGBoost + SMOTE	0.826	0.368	0.593	0.454	6.6
SVM + SMOTE-ENN	0.737	0.287	0.775	0.419	321.345	XGBoost + ADASYN	0.811	0.347	0.615	0.444	6.4
ANN	0.890	0.638	0.227	0.335	116.6	XGBoost + SMOTE-TL	0.825	0.369	0.609	0.459	6.2
ANN + SMOTE	0.765	0.308	0.735	0.434	94.4	XGBoost + SMOTE-ENN	0.791	0.332	0.699	0.450	6.8
ANN + ADASYN	0.703	0.267	0.814	0.402	163.1						

the f1-score. Concerning all oversampling methods or hybrid techniques, it is possible to indicate that these techniques increase recall and f1-score with a slight decline in accuracy and precision. The model with the best performance, considering the f1-score, without using any resampling techniques was RF which brought 0.379, with an inference time of 160.8 milliseconds regarding the time to predict the test dataset. After employing SMOTE, the RF model got 0.465, the best overall f1-score, increasing 22.69%.

Therefore, since obtaining real air combat data is quite problematic, we show through data obtained through constructive simulations that it is possible to develop decision support tools that may improve flight quality in BVR air combat since they are trying to support effective missile launches. We can use these models in the attempt to enhance the missile launching process by unmanned combat aerial vehicles or aid pilots in real air combat scenarios, increasing the effectiveness of offensive missions.

For future work, we suggest analyzing different scenarios with multiple aircraft equipped with other missiles, generating less deterministic simulations. Also, we recommend studying not just the missile launching moment but a sequence of several timeframes to understand better the coordination of the future events of a specific scenario. Besides, since the hyperparameter space for the classifiers is not complete due to computational costs, we propose expanding the grid search for all supervised machine learning models. Finally, we recommend developing more suitable sampling strategies that allow the choice of other input variables to improve the dataset to be analyzed, which can bring more reliability and generalization to the results to enhance the pilot's situational awareness.

10.7 Source Code

The source code associated with this chapter can be found at the following link: `https://github.com/jpadantas/effective-missile-launch`.

Part V

Aerospace Data Analytics

11 Simulation Analysis Library

This chapter highlights the role of data analysis in improving decisions and outcomes in air combat operations. It introduces AsaPy, a custom Python library designed to handle complex aerospace simulation data. AsaPy helps users explore large, dynamic datasets and extract insights. Through simple case studies, the chapter shows how such a tool can enhance mission performance and safety. As indicated in Figure 1.2, this chapter primarily contributes to the "Aerospace Data Analytics" area within the proposed research framework.

The library, AsaPy, is the result of a collaborative effort among experts in the field, as documented in the following reference:

> **DANTAS, J. P. A.**; SILVA, S. R.; GOMES, V. C. F.; COSTA, A. N.; SAMERSLA, A. R.; GERALDO, D.; MAXIMO, M. R. O. A.; YONEYAMA, T. AsaPy: A Python Library for Aerospace Simulation Analysis. In: Proceedings of the 38th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation. Proceedings [...]. New York, NY, USA: Association for Computing Machinery, 2024. (SIGSIM-PADS '24), p. 15–24. ISBN 9798400703638. Available at: `https://doi.org/10.1145/3615979.3656063`.

11.1 Summary

AsaPy is a custom-made Python library designed to simplify and optimize the analysis of aerospace simulation data. Instead of introducing new methodologies, it excels in combining various established techniques, creating a unified, specialized platform. It offers a range of features, including the design of experiment methods, statistical analysis techniques, machine learning algorithms, and data visualization tools. AsaPy's flexibility and customizability make it a viable solution for engineers and researchers who need to quickly gain insights into aerospace simulations. AsaPy is built on top of popular scientific computing libraries, ensuring high performance and scalability. In this work, we provide an overview of the key features and capabilities of AsaPy, followed by an exposition of

its architecture and demonstrations of its effectiveness through some use cases applied in military operational simulations. We also evaluate how other simulation tools deal with data science, highlighting AsaPy's strengths and advantages. Finally, we discuss potential use cases and applications of AsaPy and outline future directions for the development and improvement of the library.

11.2 Introduction

The application of simulation technologies in aerospace has significantly expanded, notably in commercial aviation, space exploration, and particularly in the military domain (COSTA *et al.*, 2022). This shift from live exercises to simulation is due to multiple reasons, including cost reduction and increased safety (DAVIS; HENNINGER, 2007). Simulation may be used for designing, testing, and optimizing complex systems such as aircraft, radars, and weapons (DANTAS *et al.*, 2021b). However, the vast amount of simulation data generated can be overwhelming, making the analysis process time-consuming and challenging, which may require more sophisticated algorithms and tools (DANTAS *et al.*, 2022b; DANTAS *et al.*, 2022). In response, AsaPy, a custom-made Python library, was designed in the context of the Aerospace Simulation Environment (*Ambiente de Simulação Aeroespacial – ASA* in Portuguese) (DANTAS *et al.*, 2022a; DANTAS *et al.*, 2023a) to simplify and expedite the analysis of military simulation data to support the decision-making process.

Rather than introducing new methods, AsaPy specializes in integrating a range of established techniques into a cohesive and specialized toolkit, adept at meeting the complex needs of aerospace data analysis. AsaPy offers a comprehensive pipeline of routines that typically would be performed step by step by researchers, including pre-checks before employing a specific analysis method, for example. Integrating processes into a single workflow makes AsaPy accessible even to those not proficient in programming, enabling them to apply robust analysis to aerospace data. The library includes features such as experimental design methods, statistical analysis, machine learning algorithms, and data visualization tools. This array of tools allows engineers and researchers to extract valuable insights from aerospace simulations, applicable not just in military scenarios but also in civilian and commercial aerospace sectors. Initially developed to operate alongside ASA, AsaPy's adaptable architecture also supports its use with other simulation frameworks, as will be demonstrated in the use cases section, evidenced by its integration of recognized scientific computing libraries like NumPy (HARRIS *et al.*, 2020), SciPy (VIRTANEN *et al.*, 2020), and Scikit-learn (PEDREGOSA *et al.*, 2011a), ensuring both high performance and scalability.

The main contribution of this work is to provide an overview of the key features and

capabilities of AsaPy, including its structure, effectiveness, and potential applications, mainly for analyzing aerospace and military simulation data. We also review some of the available simulation software, focusing on what data science capabilities they provide. Additionally, we bring some use cases to the AsaPy library applied to the defense context. Finally, we outline future directions for the development and improvement of the library. We have provided a link to AsaPy and hope that this library proves beneficial for other analysis projects.

11.3 Related Work

Being conceived as a part of the ASA suite, AsaPy provides an integrated solution for data science activities within a Computer-Generated Forces (CGF) package. In this context, we focused on evaluating existing CGF tools with respect to their data science features. This evaluation was made following a similar methodology as seen in Abdellaoui *et al.* (2009), Toubman *et al.* (2015). Additionally, we provide an overview of the research background on data farming and Knowledge Discovery in Simulation data (KDS), which are fundamental concepts for the development of AsaPy.

11.3.1 Existing Solutions

Abdellaoui *et al.* (2009) conducted a similar analysis and comparison of various modeling and simulation packages, with particular emphasis on their artificial intelligence (AI) capabilities. The evaluation was based on five crucial factors: architecture, autonomous operation, learning, organization, and realism. Despite the comprehensive nature of the study, notice that the authors only briefly touched on the role of data science in the context of these packages. Specifically, they made a passing reference to the existence of entity databases without delving into how data science principles might be applied to analyze simulation results.

Toubman *et al.* (2015) specifically examined the computer-generated forces (CGF) learning capabilities. Although they suggested using data for machine learning algorithms to extract behavior rules and apply them to new situations, their study did not address how commercial off-the-shelf (COTS) and government off-the-shelf (GOTS) products handle this approach. Moreover, they did not discuss how to analyze simulation data to derive general conclusions from scenario results.

We aim to expand these analyses by evaluating the status of data science capabilities within simulation packages, which would benefit researchers and practitioners in this field. Therefore, we surveyed publicly available product information for the same COTS

products (GOTS were excluded since they are not internationally available) as listed in Table 11.1, each briefly described as follows.

TABLE 11.1 – Mention of "Data Analysis" and "Design of Experiments" (DoE), on the websites of seven COTS CGF packages (in no particular order).

Product Name	Company Name	Mention of Data Analysis	Mention of DoE
STAGE	Presagis	No	No
VR-Forces	MAK Technologies	No	No
SWORD	MASA	No	No
VBS4	Bohemia Interactive	No	No
DirectCGF	Diginext	No	No
Steel Beasts Pro	eSim Games	No	No
FLAMES	Ternion	Yes	Yes

Scenario Toolkit and Generation Environment (STAGE) (PRESAGIS, 2023) is a standalone synthetic tactical simulation software that facilitates the development of models for complex war scenarios involving various platforms such as avionics, naval, and land systems. The software comes equipped with models of multiple sensors, including radar, sonar, and missile warning systems, as well as weapons such as missiles and guns (ANGHINOLFI *et al.*, 2013).

VR-Forces (MAK TECHNOLOGIES, 2023) is a simulation environment that enables the generation of multiple scenarios. The software is equipped with features required for use as a tactical leadership trainer, a threat generator, a behavior model test bed, or a Computer Generated Forces (CGF) application (PULLEN *et al.*, 2012).

SWORD (MASA, 2023) is a software suite comprising scenario creation applications, aggregated constructive simulation, and analysis tools specifically designed for staff training, education, classroom teaching, planning support, analysis, operational research, and C2 system stimulation. It is a comprehensive solution that allows simulation of operations ranging from battalion to division level and is the leading software provider in the market for training tactical level land staffs (DROZD *et al.*, 2023).

Virtual Battlespace 4 (VBS4) (BOHEMIA INTERACTIVE, 2023) is a virtual and constructive simulation platform that enables the creation and execution of military training scenarios. Its workflow and features allow for quick training initiation, simplified editing and updating of training scenarios and terrains, and collaborative training simulations across any location on its virtual Earth (EVENSEN, 2017).

DirectCGF (DIGINEXT, 2023) is a battlespace generation software by DIGINEXT, which utilizes the simulation engine DirectSim. It comes equipped with a collection of pre-built models such as platforms, sensors, weapons, electronic warfare, and communication, along with automatic and intelligent behavior. Its modular architecture facilitates reusability and enhances productivity gains, as users can integrate dedicated plug-ins into

the system (TREATY, 2021).

Steel Beasts (ESIM GAMES, 2023) is a simulation tool that models armored warfare scenarios. Military forces around the world use it to support training, mission rehearsal, and analysis of vehicle-centered scenarios featuring gunners, commanders, and drivers (MCKEOWN *et al.*, 2012).

FLAMES (TERNION, 2023) is a range of products that offer a framework for custom constructive simulations designed to meet the specific requirements of the aerospace, defense, and transportation industries. It includes customizable scenario creation, execution, visualization, and analysis and interfaces to constructive, virtual, and live systems (OEZTUERK, 2016). It is the only reviewed COTS package explicitly addressing the DoE and data analysis aspects, providing enhanced analysis options. Regarding the DoE, it accepts a manual setup in tables and imports experiment files defined by third-party tools. With respect to data analysis, it also mentions third-party programs that can deal with user-specified output files.

Besides these COTS products, we would like to mention a GOTS package that is particularly relevant to the context of this work. The Advanced Framework for Simulation, Integration and Modeling (AFSIM) is an objected-oriented C++ library used to create simulations in aerospace and defense contexts. It provides a range of features for simulating and analyzing complex operational scenarios, including air-to-air combat, air-to-ground strike, and reconnaissance missions (CLIVE *et al.*, 2015). With the focus on data science capabilities, we can point out the Visual Environment for Scenario Preparation and Analysis (VESPA), which supports creating scenario initial condition files that are compatible with AFSIM-based applications, enabling its usage as a DoE tool.

In summary, all the reviewed solutions lack an integrated approach with more comprehensive data science tools. FLAMES and AFSIM seem to be the closest to what AsaPy aims to provide within the context of ASA. However, they still rely on third-party packages and focus on data recording and visualization rather than the analysis itself.

11.3.2 Relevant Concepts

Originally, the concept of data farming emerged in the context of military simulations, providing decision-makers with the "Commander's Overview" for enhanced decision support (HORNE; SEICHTER, 2014). By encompassing a broad spectrum of parameter spaces and having the ability to explore both positive and negative effects, relationships, and potential options, data farming may unveil aspects not previously addressed in military simulation applications.

In other words, data farming has been employed to describe the intentional generation

of data from simulation models. Through extensive designed experiments, one can efficiently and effectively "cultivate" simulation output. This approach allows for exploring vast input spaces and discovering noteworthy features in complex simulation response surfaces. Embracing this innovative mindset enables significant advancements in the scope, depth, and timely acquisition of insights provided by simulation models (LUCAS *et al.*, 2015).

There are three primary goals in data farming (KLEIJNEN *et al.*, 2005): (i) developing a fundamental understanding of the simulation model and the emulated system; (ii) identifying robust policies and decisions; and (iii) comparing the merits of various policies or decisions. Well-designed experiments prove to be efficient and effective tools for achieving these objectives. Despite the exponential increase in processing capabilities, the strategic design of experiments remains essential for obtaining comprehensive insights through large-scale simulation studies.

The KDS combines data farming with visual analytics-based methods (FELDKAMP *et al.*, 2020). The idea is to start by defining experiments, focusing on the selection of factors. Factors can encompass a wide array, including structural, organizational, technical data, system load, and material flow data. The number of factors, along with the lower and upper-value limits for each factor, is chosen as expansively as possible unless they are evidently irrelevant or physically implausible. For data generation, the simulation model is treated as a black box that transforms a set of input factor values into output data. This output data is then stored in a simulation output database, where a row of parameter values represents each experiment. Notably, experiments can be easily distributed across parallel machines as they are independent of each other. Once all experiments are conducted, the data analysis phase can proceed. The initial analysis begins with the simulation output data, and subsequently, knowledge is derived, especially when exploring relationships between output data and input factors. Well-suited visualizations play a crucial role in establishing connections between corresponding input and output sets, enabling users to investigate and draw conclusions.

Both data farming and its combination with visual analytics (KDS) have been the basis for establishing AsaPy features, which aim to provide means for not only intentionally generating simulation data but also obtaining insights from this data.

11.4 Structure

From design, we aimed to develop a library that would help analyze simulation data, especially for military scenarios. In the ASA context, we noted that the analyst tends to follow a pattern that can be broken into four steps, shortly summarized:

(i) Design of Experiments, in which we define the input configuration for the executions;

(ii) Execution Control, in which we monitor the progress of a batch of executions;

(iii) Analysis, in which we conduct the actual data analysis; and

(iv) Prediction, in which we train a model to predict the outcome of new input configurations.

The literature concerning data analysis is vast, therefore it is important for our architecture not to limit the options available to the analyst. For this reason, we opted to package existing Python libraries that implement the desired methods, leveraging the benefits of this ecosystem and allowing easy interoperability.

Therefore, our structure consists of a curated set of third-party libraries wrapped in a standardized and extensible way. The code is divided into four modules: `analysis`, `models`, `doe`, and `utils`, which roughly resembles the steps mentioned above.

With this architecture, illustrated in Figure 11.1, we allow the analysts to use the techniques independently or in combination or even to extend the package with other desired methods. Furthermore, we can automatize the process for the analyst, choosing reasonably appropriate tools for specific tasks, thus guiding his work to conduct a proper analysis.

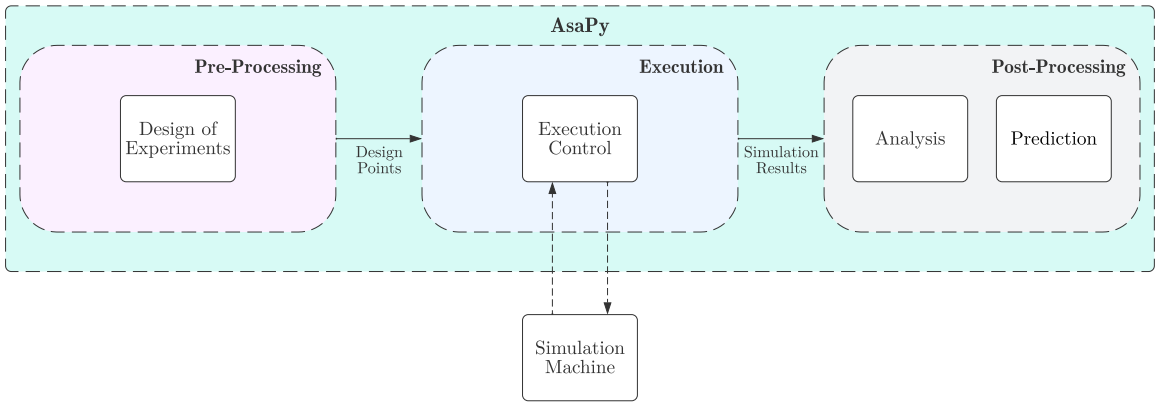


FIGURE 11.1 – The structure of the AsaPy library in the perspective of the analyst workflow.

Although designed to be used integrated with other ASA suite tools, AsaPy is generic with respect to the simulation machine chosen to actually execute the scenarios and generate the data, as discussed in Subsection 11.4.2. For illustration, the integration with ASA service for parallel batch execution is implemented in the `asa-client` package, which has features such as authentication on the ASA platform, access to available scenarios, submission of experiment executions, and retrieval of execution results, not the focus of this work though.

In the following subsections, we discuss the techniques available for each one of the mentioned steps, locating them in the library structure.

11.4.1 Design of Experiments

The Design of Experiments (DoE) step, enabled through the `doe` module, integrates a comprehensive array of tools and features for experimental design, as outlined in Montgomery (2017). Presently, the module offers the Latin Hypercube Sampling (LHS) technique (BOX *et al.*, 2005) as its primary method. Plans to augment this suite with advanced methods are underway, with particular emphasis on incorporating the Nearly Orthogonal Latin Hypercube (NOLH) technique, a more refined approach discussed in detail by Cioppa and Lucas (2007).

This module is adept at managing various data types, including numerical, categorical, and boolean. It is instrumental in generating input samples for diverse simulation executions. Typically, these simulations run in batch mode, subject to fluctuating input parameters. The module also includes strategies for orchestrating metrics to assess simulation performance. This aspect is crucial for making informed decisions about whether to prematurely halt batch execution, a topic further explored in Subsection 11.4.2.

To utilize this tool effectively, analysts must specify which input variables are subject to modification, either manually or via automation (as implemented in the `asa-client` using the ASA suite). After choosing a sampling technique, it generates the execution configurations and sends the design points to the next phase.

11.4.2 Execution Control

Once the scenarios are created and the input parameters assigned, the next step is to run such experiments. Therefore, in this subsection, we discuss the control of the executions, including the process of splitting the total amount of runs into chunks and evaluating metrics to determine whether to stop the batch execution early.

In the military context, the desired analysis is usually complex, requiring many executions to extract meaningful information. For this reason, the individual runs are usually dispatched as a batch, optimizing the usage of the computational resources available. However, often, not all executions planned are necessary for the analysis, which unfortunately can only be known during its execution. The large batch is broken into chunks of experiments to handle this limitation. Then, each chunk is sequentially executed until completion using all the available computational resources. After each chunk completion, the concerned metrics can be analyzed to decide whether to stop the batch execution or start the next chunk. To do so, one can observe the variation of the expected value of

a significant variable before and after running the last chunk. If it is below a certain threshold, we can assume that this statistic has converged, economizing on the number of individual executions.

Though naive, this heuristic, actually implemented by AsaPy, demonstrates how to apply a method to early stop a batch execution, saving time for the analyst and reducing the usage of the computational resources. Naturally, these evaluation metrics used for early stop criteria will depend on the objectives of the simulation.

For instance, consider a scenario in the defense context where simulations are conducted in batches to optimize the number of aircraft needed to neutralize all enemy aircraft. Evaluation metrics for each simulation might encompass the number of remaining enemy aircraft and the number of missiles expended by the conclusion of each simulation. The unique aspect here is the introduction of early stop criteria, which are assessed not for individual simulations but across the entire batch. Should a significant portion of simulations within the batch meet these criteria early on, the whole batch can be terminated beforehand. Results from the simulations completed up to that point are then analyzed. This method proves efficient, conserving both time and resources, especially when the criteria are satisfied early in the batch run. It is important to note that the end of an individual simulation is controlled by the parameters set in the simulation file and is not directly associated with AsaPy.

Describing its intended usage, the control of executions is carried out by the `ExecutionController` class, which is instantiated with two functions and one number, as exemplified in Listing 11.1. The first argument is a function that is responsible for effectively running the executions: receiving the collection of design points and returning the corresponding results. Moreover, the second function is the actual stop criteria, as already mentioned. Finally, the third argument is the size of the chunks into which the batch will be split.

Listing 11.1 – Usage example of the Execution Control module

```
1  def simulate(doe: pandas.DataFrame) -> pandas.DataFrame:
2      # 1. send execution requests using the Asa-client
3      # 2. retrieve executions results
4      return pandas.DataFrame.from_dict(asa_results)
5
6  def stop_check(result: pandas.DataFrame, last_result: pandas.DataFrame)
       -> bool:
7      # compare results using Asapy or custom functions
8      return compare_results(result, last_result)
9
10 ec = asapy.ExecutionController(simulate, stop_check, 100)
11 result = ec.run(doe)
```

Notice that this functional style allows the library client code to define how the simulations should be executed and what criteria should be used. This empowers analysts to use whichever simulation machine they may desire, just requiring the implementation of a function that interfaces with the chosen simulator and respects the expected signature. This interoperability easiness was one of the library design cardinal points and is now illustrated.

11.4.3 Analysis

The analysis step is supported by the `analysis` module and provides a range of tools for analyzing and exploring simulation data. Therefore, in this section, we discuss some of the available components and how the package automatically chooses an appropriate technique for the analyst.

One of the main features of the library is hypothesis testing, which determines whether a specific hypothesis about the data is true or false (KRUSCHKE, 2013). This component offers a collection of statistical tests from which to choose. Focusing on the needs of the typical analyst, the module utilizes the decision flow depicted in Figure 11.2 to automatically select an appropriate test for the data being analyzed. For instance, AsaPy can streamline the process of conducting an ANOVA (Analysis of Variance) test, not only by performing the test itself but also by including all requisite pre-test checks and data visualizations to aid in interpreting the results. Integrating a full analytical pipeline – from preliminary data checks to post-analysis interpretation tools – represents a methodological advancement. This ensures that users are not only able to execute the desired statistical tests but also do so with a comprehensive understanding of the prerequisites and implications of these tests.

Another functionality is the distribution fit technique (TAYLOR, 2017), used to adjust a particular statistical distribution to the data. This component gives information about the distribution that best fits the input data among the most common ones (SCHUSS, 2009), such as the normal, uniform, exponential, chi-squared, and beta distributions. The process of fitting distributions involves estimating their parameters, allowing us to extrapolate data beyond the range of the observed values.

This package also includes methods for determining feature scores (JOLLIFFE, 2002), which are used to rank the importance of different attributes in the dataset. These techniques are particularly helpful in filtering out non-informative or redundant variables from the input data (KUHN *et al.*, 2013).

In addition, the package includes tools for Pareto front analysis (DEB *et al.*, 2002), used to identify the optimal trade-off between two conflicting objectives. This is a useful tool

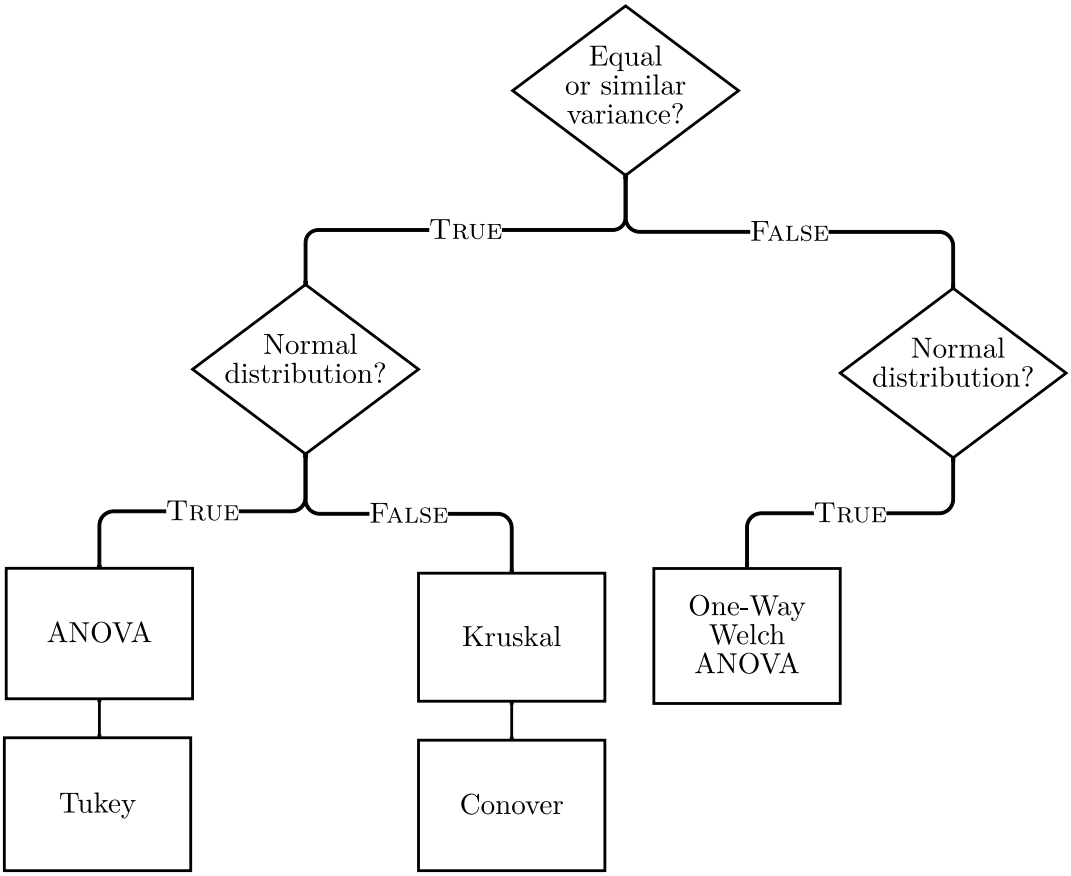


FIGURE 11.2 – Flow diagram for hypothesis testing using AsaPy.

for decision-making in complex systems.

Furthermore, the analysis package also provides methods for detecting and removing outliers in the data (ROUSSEEUW; DRIESSEN, 2019), which can significantly affect the accuracy of the analysis. This is achieved using various statistical techniques, such as the standard deviation and interquartile range methods.

Finally, the package provides exploratory data analysis (EDA) tools (TUKEY, 1977), which can be used to visualize and understand the data. This toolkit consists of a table of class balance for categorical variables, association, correlation, histograms, and boxplots with information on the number of outliers for numerical variables. These preliminary analyses can help identify patterns, trends, and anomalies in the data and guide the choice of statistical models and analysis methods.

11.4.4 Prediction

The prediction step, the last one, uses the `models` module and provides a comprehensive framework for building custom machine learning models, including but not limited to neural networks and random forests. This package covers the entire process of creating a model, including phases such as data preprocessing, hyperparameter tuning, cross-validation, evaluation, and prediction. The most significant advantage of using this module is that it allows us to obtain estimated results without performing new simulations, thus saving time and computational resources.

The package is built on top of popular machine learning libraries, such as TensorFlow (ABADI *et al.*, 2016) and Scikit-learn (PEDREGOSA *et al.*, 2011a). This allows for easy integration with other machine learning tools and workflows, aligned with the cardinal points conceived during the package design.

The `models` module provides various preprocessing methods to transform raw data into a format that can be used by machine learning algorithms. These methods include scaling, normalization, feature engineering, and more. It also provides options for handling missing values and categorical features, along with a wide range of available models. To help users effectively utilize these features, we provide some tutorials in our repository to guide the process of creating and evaluating models using the `models` package.

Hyperparameters are parameters that are not learned during the training process but are set before the training begins. These hyperparameters can have a significant impact on the performance of the model. The `models` package includes methods for hyperparameter tuning, such as random search (BERGSTRA *et al.*, 2011), which optimizes model performance by establishing the most effective hyperparameters before the training process.

Cross-validation is a technique used to evaluate the performance of a model. The `models` package includes various methods for performing cross-validation, including k-fold cross-validation (ARLOT; CELISSE, 2010).

The `models` package provides various evaluation metrics to measure the performance of a model to solve regression or classification problems. These metrics include accuracy, precision, recall, F1 score, mean squared error, and more. The package also provides options for visualizing the model's performance using plots.

11.4.5 Support module

The AsaPy library provides users with a primary support module, namely `utils`. This module supply additional tools and utilities to users for handling data.

The `utils` module is a helpful tool for performing mathematical calculations and conversions in various fields, including geodesy, physics, and engineering. This component contains an assortment of constants, methods, and functions for converting distance and angle measurement units and changing coordinate systems. One of the key features of the `utils` module is its ability to convert between different units of distance and angle measurement, such as meters, kilometers, feet, miles, radians, and degrees. The package includes methods for executing these conversions with ease, making it simple to switch between units as necessary.

Another essential aspect of the `utils` module is its support for various coordinate systems. This module provides methods for converting between different coordinate systems, such as geodetic, geocentric, and Cartesian coordinate systems. This can be especially useful for applications in geodesy and geolocation, where precise positioning is critical. For example, the Geod class in the `utils` module provides methods for converting between geodetic and Cartesian coordinates and calculating distances and bearings between points on the Earth's surface. The ECEF (Earth-centered, Earth-fixed) class, on the other hand, offers methods for converting between ECEF and geodetic coordinates and calculating the distance between points in ECEF space. In summary, the `utils` module provides users with essential tools and utilities for handling data, performing mathematical calculations and conversions, and changing coordinate systems.

11.5 Applications in BVR Air Combat Simulations

In this section, we demonstrate how AsaPy can be effectively used to analyze military operational scenarios, especially in the context of beyond visual range (BVR) air combat simulations. BVR air combat is a challenging and critical field in which engagements occur beyond the pilot's visual range (DANTAS, 2018). These engagements make use of advanced weaponry and sensor systems. AsaPy has been employed in various applications to enhance BVR air combat simulations that feature agents controlled by artificial intelligence models (DANTAS *et al.*, 2023), demonstrating its capabilities and versatility. This section will discuss three primary applications of AsaPy, as implemented in other works using ASA and other simulation software.

Toward the end of the section, we introduce a new example analysis featuring a BVR fighter aircraft navigation scenario using ASA. This scenario serves as a practical illustration of AsaPy's applicability in a specific aerospace context, further emphasizing its role as a versatile tool within the aerospace domain.

11.5.1 Engagement Decision Support

The study conducted by Dantas *et al.* (2021a) aimed to develop an engagement decision support tool for BVR air combat in the context of Defensive Counter Air (DCA) missions. In BVR air combat, engagement decision refers to the moment when the pilot decides to engage a target by executing corresponding offensive maneuvers.

To plan the execution of simulations, the authors used the `doe` module from AsaPy, selecting key variables, including categorical and numerical with different coverage ranges, to simulate a BVR air combat. The simulation data was pre-processed and explored using the `analysis` module to organize and better understand the data. These variables included the distance, angle between the longitudinal axis, and difference in altitude between the reference and the target. The authors ran 3,729 constructive simulations that lasted 12 minutes each, resulting in 10,316 engagements.

The authors evaluated the simulations using an operational metric called the DCA index, which represents the degree of success in this type of mission based on the expertise of subject matter experts. The DCA index is based on the distances between aircraft from both the same team and opposing teams, as well as the number of missiles deployed. The index indicates the likelihood of success in BVR air combat during DCA missions. The primary aim of these missions is to establish a Combat Air Patrol, which requires aircraft to fly in a specific pattern around a designated location.

The authors employed the `models` module from AsaPy to build a supervised machine learning model based on decision trees to determine the quality of a new engagement, using the engagement status right before it starts and the average of the DCA index throughout the engagement. Beyond model creation, the authors seamlessly integrated AsaPy for data preprocessing and hyperparameter tuning as well. Overall, the authors utilized various features of the AsaPy library to plan, execute, and analyze their simulations and to build and evaluate a model for engagement decision support in BVR air combat.

11.5.2 Weapon Engagement Zone Evaluation

Still in the BVR air combat context, Dantas *et al.* (2021b) used AsaPy to analyze simulation data generated by the ASA simulation environment, explicitly focusing on calculating an air-to-air missile's weapon engagement zone (WEZ). The WEZ allows the pilot to identify airspace where the available missile is more likely to successfully engage a particular target, i.e., a hypothetical area surrounding an aircraft where an adversary is vulnerable to a shot.

Designing experiments for missile launches in BVR air combat is a complex process

that involves considering various input variables. These variables help to simulate different scenarios and identify the best possible outcomes for missile launches. The seven input variables for the simulation runs include the shooter altitude, shooter speed, target altitude, target speed, target heading, the relative position of the target, and shooter pitch.

Each variable plays a crucial role in determining the WEZ maximum range. The authors used the `doe` module to perform the LHS method from AsaPy to plan and design the simulation experiments using these variables.

The simulations were executed in chunks to improve the computational efficiency and reduce the simulation time. After completing the simulation runs, the authors collected the output data and analyzed it using the `analysis` package in AsaPy, generating data visualization, feature engineering, and statistical tests to understand the data distribution and identify relationships between the input variables and the WEZ.

Using the data from the simulations and analysis, the authors built a supervised machine learning model using a Deep Neural Network (DNN) to predict the WEZ maximum launch range for a given scenario.

Finally, the authors used the `models` package in AsaPy to build and train the model to predict the WEZ maximum launch range for a given scenario. They evaluated its performance using metrics such as the mean absolute error and the coefficient of determination to ensure that the model accurately predicted the WEZ for different scenarios.

In a related study presented by Dantas *et al.* (2023b), the emphasis was on the WEZ of Surface-to-Air Missiles (SAMs). SAMs hold a crucial position in the landscape of modern air defense systems. The WEZ's significance is further accentuated as it is directly associated with a missile's maximum range, marking the farthest interception distance between a missile and its target.

Conventional simulation methods, in many instances, result in significant computational demands and extended processing times. To address these challenges, the study incorporated machine learning techniques, using AsaPy prediction methods synergized with specialized simulation tools to train supervised algorithms. Through AsaPy, researchers were able to streamline the simulation and analysis process, making it more efficient and data-driven.

By utilizing a comprehensive dataset from earlier SAM simulations, the model demonstrated remarkable accuracy in predicting the SAM WEZ based on new input parameters. This combination of machine learning and advanced simulation tools not only accelerated SAM WEZ simulations but also bolstered strategic planning in air defense, offering invaluable real-time insights that enhance the performance of SAM systems. The study, through AsaPy, provided an in-depth analysis of different machine learning algorithms,

elucidating their capabilities and performance metrics. It not only suggested avenues for future research but underscored the transformative potential of incorporating machine learning into SAM WEZ simulations.

11.5.3 Missile Hit Prediction

In Dantas *et al.* (2022), the authors analyzed both defensive and offensive scenarios in BVR air combat using AsaPy. The authors used AsaPy to generate constructive simulations of BVR air combat scenarios and extract various features related to situational awareness from the simulation data. They designed a multilayer perceptron neural network incorporating data from these simulations to enhance pilots' situational awareness during in-flight decision-making. By training their machine learning models based on neural networks on this data, they could accurately predict a pilot's situational awareness based on the missile's ability to hit the target. Therefore, Dantas *et al.* (2022) demonstrated the potential of machine learning in BVR air combat scenarios by generating fast and reliable responses concerning the tactical state to improve the pilot's situational awareness and, therefore, the in-flight decision-making process.

One of the key strengths of AsaPy is its ability to work with various simulation software, including commercial and open-source platforms (Figure 11.3). This versatility allows users to leverage the power of AsaPy regardless of the simulation software they are using, making it a valuable tool for military operational scenario analysis. For example, in Dantas *et al.* (2022b), the authors aim to develop a machine learning model that can predict the effectiveness of missile launches in BVR air combat scenarios. To generate the simulation data, the authors used the FLAMES simulation platform, a commercially available simulation software suite. The AsaPy library was used to organize and analyze the simulation data generated by FLAMES. The authors used AsaPy to build seven different supervised machine learning models that predict the effectiveness of missile launches in BVR air combat scenarios. To improve the performance of the machine learning models, the authors also used resampling techniques such as SMOTE (CHAWLA *et al.*, 2002) to generate more data on missile launches. This approach helped to address the class imbalance issue that commonly occurs in military operational scenarios, where successful missile launches are relatively rare compared to unsuccessful ones.

Overall, the successful implementation of AsaPy in those missile hit-prediction works demonstrates its versatility and utility in analyzing and modeling military operational scenarios. The AsaPy library proved a valuable asset in the data analysis and modeling process for the aforementioned works, providing efficient data preprocessing and analysis tools and aiding in developing models to solve complex problems in the BVR air combat context. AsaPy's compatibility with various simulation software highlights its versatility,

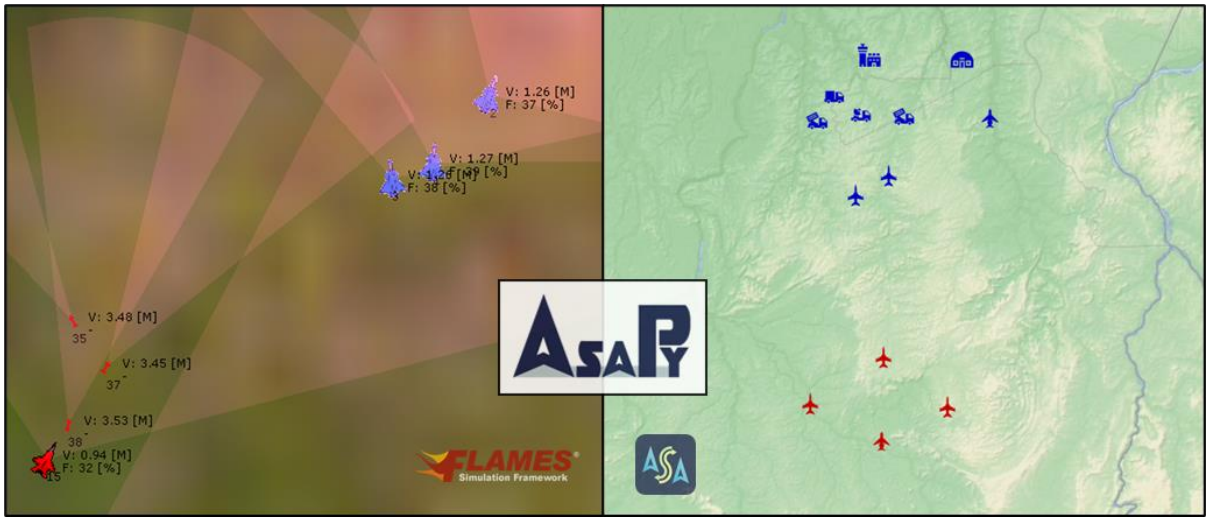


FIGURE 11.3 – Examples of simulation platforms in which AsaPy may be employed: FLAMES (left) and ASA (right).

making it an excellent choice for researchers and practitioners in the military and defense sectors who frequently work with multiple simulation platforms.

11.5.4 Fighter Aircraft Navigation

In this subsection, we explore the complexities of fighter aircraft navigation, examining the interconnections between various flight parameters and their impact on fuel efficiency. This analysis, supported by detailed experimental data, aims to deepen our understanding of efficient aircraft operation.

The scenario under examination involves a navigation flight executed by a fighter aircraft, characterized by diverse maneuvers at various altitudes and speeds. The aircraft navigates between 10,000 and 35,000 feet, adjusting its speed from 350 to 550 knots. Additionally, the flight includes a 10-minute holding maneuver at the third route point, where the aircraft follows a circular path in the air. This maneuver is typically used for traffic management or to delay the aircraft before landing.

This specific case study comprises two experiments aimed at elucidating the methodology of extracting and analyzing data from simulations to address pertinent questions in aerospace studies. For an in-depth understanding of this process, we invite you to examine the code associated with our analysis.

11.5.4.1 Experiment 1 – Analysis of the Relationship between Time of Flight and Fuel Consumption:

The first experiment investigates the link between time of flight, denoted as `time_of_flight`, and fuel consumption, referred to as `fuel_consumed`, in a flight simulation scenario. The main goal is determining if longer flight durations directly relate to increased fuel consumption. This involves analyzing data from 4,000 flight simulations, focusing on total flight duration in seconds and the amount of fuel consumed in pounds.

To accomplish this, various statistical methods, including linear regression and correlation analysis, are used. Furthermore, data visualization techniques, especially scatter plots, are utilized to interpret the relationship between these key variables (Figure 11.4).

The central theory suggests a direct, positive correlation between time of flight and fuel consumption, indicating that longer flights generally result in higher fuel usage. However, unexpectedly, the results show no clear linear relationship between these variables. This surprising finding is mainly due to variations in speed and altitude during the simulations, indicating that these factors significantly affect fuel consumption dynamics. This complexity, going beyond simple linear correlations, highlights the need for more research into how speed and altitude change influence fuel consumption.

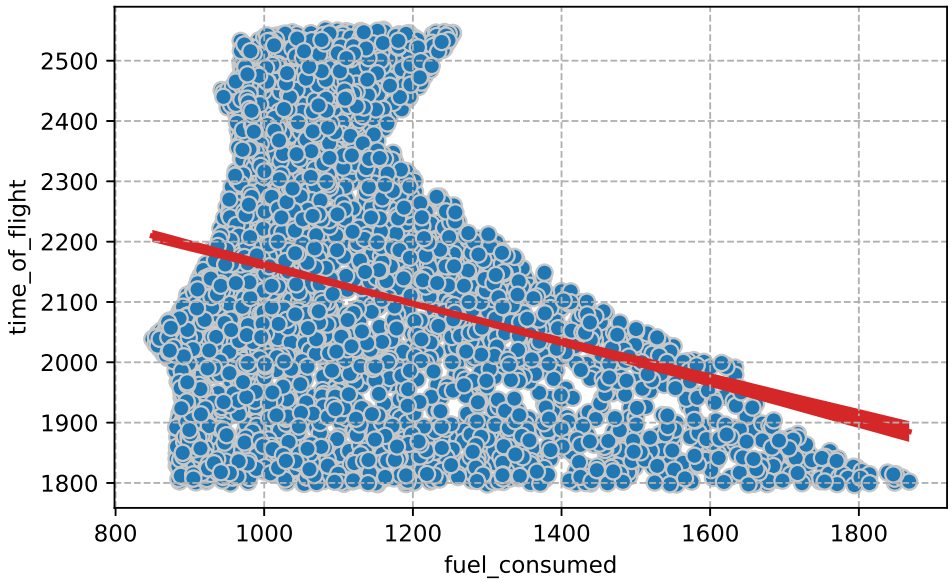


FIGURE 11.4 – Linear regression of Time of Flight vs. Fuel Consumed.

11.5.4.2 Experiment 2 – Analysis of the Relationship between Speed, Altitude, and Fuel Consumption:

Expanding on the initial experiment's results, the second experiment clarifies the relationship between flight speed, altitude, and fuel consumption in a simulation context.

The goal is to understand how these factors, both individually and together, impact an aircraft's fuel efficiency. This investigation involves analyzing extensive flight simulation data, focusing on the interplay between speed in knots, altitude in feet, and fuel consumption measured in pounds. The study reveals complex relationships and patterns using statistical analyses, two-dimensional charts, and surface plots (Figure 11.5).

The primary hypothesis of the experiment suggests that both speed and altitude significantly affect fuel consumption in a potentially complex and interactive manner. The research aims to determine if higher speeds or altitudes proportionally increase fuel consumption and to identify optimal efficiency points.

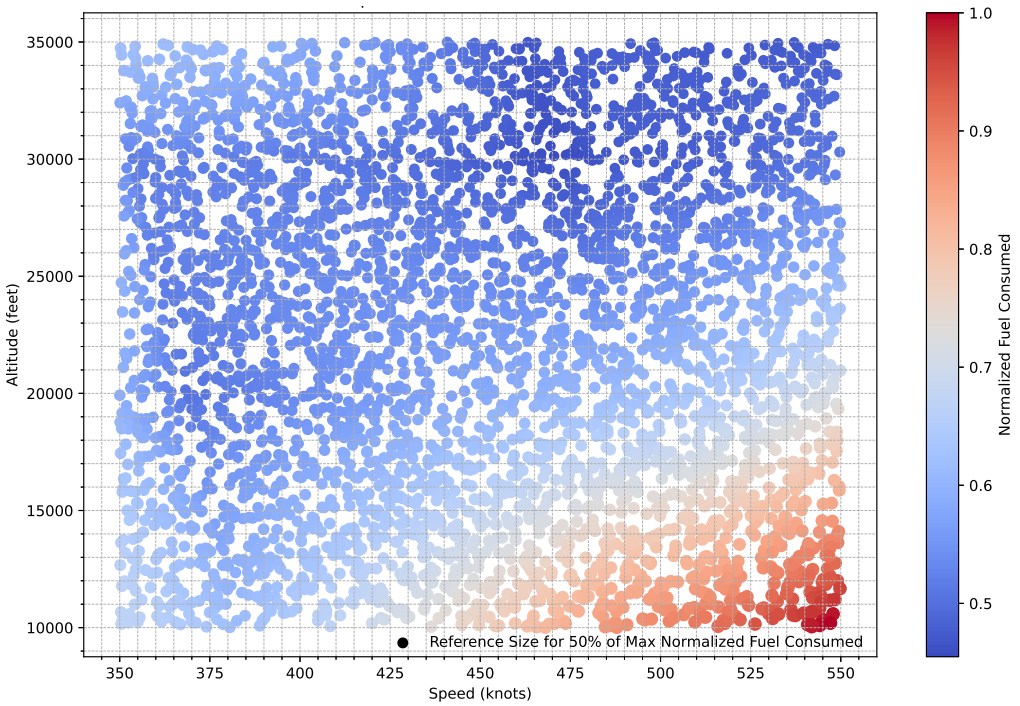
One of the key findings of the experiment is the discovery of fuel consumption peaks. These peaks are most prominent in higher and red regions of the chart, occurring at an altitude of approximately 10,000 feet and a speed of about 525 knots. At these parameters, the consumption reaches nearly 1800 pounds. This insight helps to understand the conditions under which fuel consumption is maximized, informing operational and design decisions for aircraft.

In contrast, the study also identifies areas of operational efficiency. These are indicated by blue areas on the chart, representing lower fuel consumption and suggesting more efficient operating ranges. The lowest consumption values, close to 1000 pounds, are observed at altitudes of around 25,000 to 30,000 feet and speeds between 400 to 450 knots. This finding is significant for optimizing flight paths and aircraft design for energy efficiency.

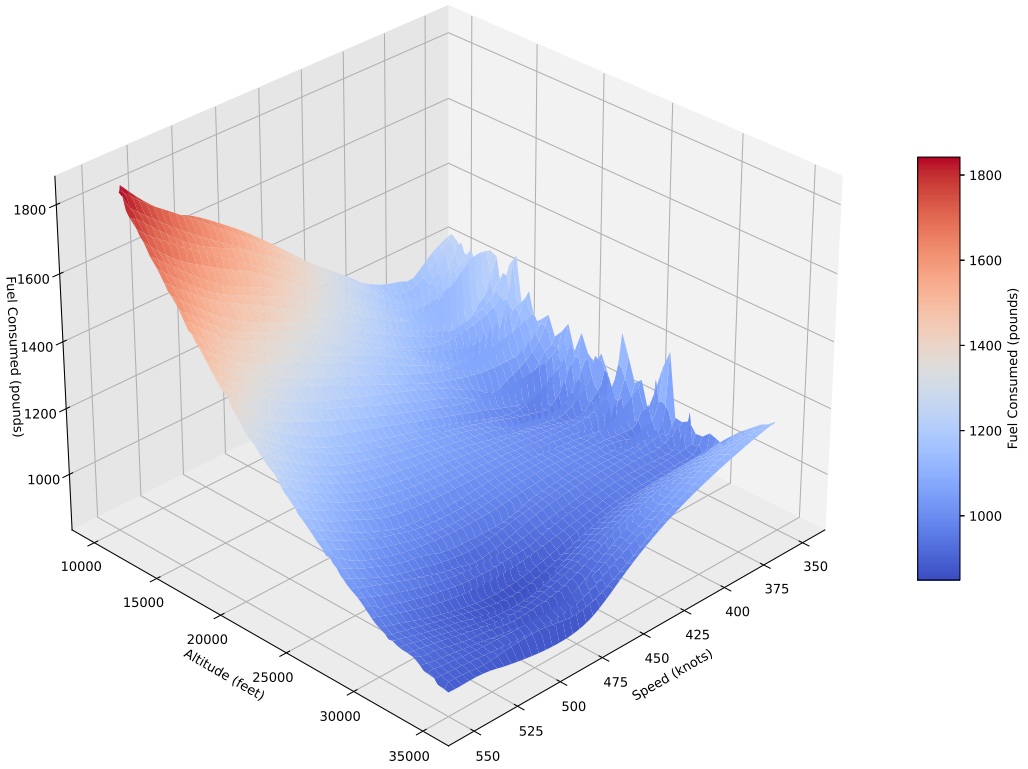
Additionally, the experiment reveals a complex variation in fuel consumption at certain intermediate speeds and altitudes. This observation indicates an operational efficiency point that does not follow a simple linear relationship with speed or altitude, adding a layer of complexity to the understanding of aircraft fuel efficiency.

Moreover, the insights gained from the chart are invaluable for planning routes that prioritize fuel efficiency. By avoiding altitude and speed ranges that result in excessive consumption, significant improvements in operational cost can be achieved.

Finally, the experiment's results reveal insights into aircraft performance. The data illustrate how the engine and aircraft perform under various operational conditions, aiding engineers in optimizing or developing more efficient propulsion systems. These findings are important for advancing the field of aeronautical engineering and contribute to the development of more efficient aircraft.



(a) 2D normalized surface plot showing the relationship between speed, altitude, and fuel consumption.



(b) 3D surface plot illustrating the dynamic interaction between speed, altitude, and fuel consumption.

FIGURE 11.5 – Comparative analysis of aircraft performance parameters. (a) presents a 2D perspective, while (b) offers a 3D visualization, providing a comprehensive overview of fuel efficiency dynamics.

11.6 Outcomes

In conclusion, AsaPy distinguishes itself as a versatile Python library that streamlines and accelerates the analysis of simulation data. Its features, including experiment design, statistical analysis, machine learning algorithms, and data visualization tools, make this library a good resource for engineers and researchers in simulation studies, particularly in the aerospace and military domains.

Future work on AsaPy is multifaceted, aiming at both enhancement and expansion. We plan to integrate additional DoE methods and machine learning algorithms to broaden its applicability. Another priority is optimizing the performance of AsaPy's algorithms, to enable faster processing and the handling of larger datasets. Enhancing interoperability through integration with other tools and platforms is also on our agenda, further improving usability. Continual refinement of the documentation and user interface will make AsaPy more user-friendly and accessible.

A key focus of our future work is the practical demonstration of AsaPy's effectiveness in real-world scenarios. We propose a comprehensive analysis of AsaPy's impact on data analysis efficiency. This would involve contrasting the processes of managing simulation output data from different systems, such as FLAMES or ASA, with and without AsaPy. The emphasis will be on assessing how AsaPy streamlines tasks like data reading, loading, cleaning, and preliminary analyses.

Additionally, we are planning to expand our suite of data analysis tools, particularly focusing on expanding supervised learning algorithms and introducing unsupervised learning methods, such as clustering and principal component analysis (PCA). This expansion aims to enhance AsaPy's ability to uncover patterns and relationships in data without the need for pre-labeled outcomes.

11.7 Source Code

AsaPy is available as an open-source library and can be downloaded from its GitHub repository at `https://github.com/ASA-Simulation/asapy`.

Part VI

Aerial Autonomous Agents

12 Deep Reinforcement Learning in Air Combat

This chapter explores designing and implementing a deep reinforcement learning-based autonomous agent for air combat simulations, leveraging the AsaGym Library as a high-fidelity simulation environment. The proposed agent is designed to emulate a high-performance fighter aircraft, learning to optimize its tactical decisions through a reward-driven approach aligned with operational performance metrics. As indicated in Figure 1.2, this chapter primarily contributes to the "Aerial Autonomous Agents" area within the proposed research framework.

The research presented in this chapter is based on the following works:

> **DANTAS, J. P. A**.; MAXIMO, M. R. O. A.; YONEYAMA, T. Autonomous Agent for Beyond Visual Range Air Combat: A Deep Reinforcement Learning Approach. In: Proceedings of the 2023 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation. Proceedings [...]. Orlando, FL, USA: Association for Computing Machinery, 2023. (SIGSIM-PADS'23). ISBN 979-8-4007-0030. Available at: `https://doi.org/10.1145/3573900.3593631`.
>
> **DANTAS, J. P. A.***; MEDEIROS, F. L. L.*; SAMERSLA, A. R.*; BOTELHO, P. L. R.; GOMES, V. C. F.; SILVA, S. R.; FERREIRA, Y. D.; ARANTES, A. O.; AQUINO, M. R. C.; MAXIMO, M. R. O. A. Deep Reinforcement Learning Agents with Collective Situational Awareness for Beyond Visual Range Air Combat. IEEE Access, 2025. (The manuscript is under review at the time of writing this thesis — third round.)

12.1 Summary

This chapter explores Beyond Visual Range (BVR) air combat simulations, focusing on two-versus-two scenarios involving autonomous agents. The engagement phase in BVR

*Equal contribution.

combat presents complex and unpredictable situations, as it is difficult to anticipate the behavior of opposing aircraft and the outcomes of tactical decisions, especially in multi-agent settings. A promising approach is the use of Deep Reinforcement Learning (DRL), which enables agents to learn from dynamic environments. According to fighter pilots, collective situational awareness, defined as understanding the spatial distribution and orientation of allies and opponents, is essential for executing coordinated tactical maneuvers. The main contribution of this chapter is AsaGym, a library for developing and training DRL-based fighter agents in BVR scenarios. A case study demonstrates its use, applying a reward function that promotes coordination based on collective situational awareness, and compares different DRL algorithms to assess their ability to foster cooperative behavior. The results highlight DRL's potential to address the complexities of modern air combat and support the development of more adaptive and effective tactics in multi-agent BVR scenarios.

12.2 Introduction

Air combat is a complex and dynamic scenario where skilled pilots make quick decisions to gain a tactical advantage over their opponents (YANG *et al.*, 2022). Beyond Visual Range (BVR) air combat, in particular, involves engagements taking place at distances where pilots cannot see the enemy aircraft (HIGBY; COL, 2005; DANTAS *et al.*, 2022). While some air combat still occurs within visual range (WVR), most engagements start in BVR. This phase is often the most important, as it can give advantages or create difficulties for the later stages of combat. The main challenge for pilots is the planning of maneuvers, which shows their ability to think tactically and decide the outcome of the fight (HU *et al.*, 2021).

Computer simulations of BVR air combat can recreate many different situations, helping to test new tactics, sensors, and weapons (DANTAS *et al.*, 2022b). One of the hardest parts of these simulations is mimicking the complex behaviors of pilots during all stages of combat. These decisions include adapting to new situations, coordinating with allies to execute strategies, and timing missile launches effectively.

This work explores the learning of BVR engagement maneuvers by autonomous agents. Engagement involves maneuvering the aircraft to gain an advantage over the opponent, that is, to position the enemy within the effective range of the aircraft's own missiles, known as the Weapon Engagement Zone (WEZ), while staying outside the opponent's WEZ (DANTAS *et al.*, 2025; DANTAS *et al.*, 2021b). The engagement stage becomes even more complex when there is more than one opponent.

A promising solution to this problem is the use of Reinforcement Learning (RL), which allows autonomous agents to learn from challenging experiences. RL is a machine learning

method in which an autonomous agent learns to make better decisions by interacting with its environment. The agent receives rewards or penalties for its actions and adjusts its strategy to maximize the rewards over time (SUTTON; BARTO, 2018). Deep Reinforcement Learning (DRL) is a more advanced form of RL that uses deep neural networks to manage complex environments, enabling agents to make decisions in dynamic and uncertain conditions, such as BVR air combat (LECUN *et al.*, 2015).

In this context, existing simulation environments for air combat often lack modularity, support for multi-agent DRL experimentation, or mechanisms to incorporate operational insights such as spatial coordination between allied and enemy aircraft.

Therefore, the main contribution of this work is:

- Development of AsaGym, a library for simulating and training DRL-based autonomous fighter agents in BVR air combat, with a case study demonstrating its use with a reward design that promotes coordination and situational awareness.

In addition, we provide the following specific contributions:

- Design of a task-oriented reward function that encourages cooperative agent behavior based on spatial relationships among allies and opponents.
- Incorporation of operational knowledge from Brazilian Air Force (FAB) fighter pilots, serving as Subject Matter Experts (SMEs), who emphasized the importance of spatial awareness for coordinated maneuvers in BVR combat.
- Comparative evaluation of four state-of-the-art DRL algorithms — Proximal Policy Optimization (PPO), Soft Actor-Critic (SAC), Twin Delayed Deep Deterministic policy gradient (TD3), and Advantage Actor-Critic (A2C) — applied to the engagement phase in a simulated BVR air combat scenario.

The remainder of this work is organized as follows. Section 12.3 presents an overview of related work, highlighting previous research on DRL applications in BVR air combat simulations. Section 12.4 details the proposed methodology, including the design of the DRL models used to represent the engagement phase of BVR combat, and the experimental setup used for training and evaluation. The results and analysis of the conducted experiments are discussed in Section 12.5, providing insights into the agent's performance across different scenarios. Finally, Section 12.6 summarizes the key findings and outlines potential directions for future work.

12.3 Related Work

RL has been widely used to model the engagement stage in constructive BVR air combat simulations. Most studies employ two types of rewards: intermediate and final. The intermediate reward is provided during training whenever the agent makes a decision, reflecting its current tactical advantage or disadvantage relative to opponents. The final reward is assigned at the end of the training episode, often based on the difference between the number of surviving friendly and opposing agents. A positive value indicates a favorable outcome, while a negative value indicates an unfavorable one. In this context, a group of multiple aircraft is referred to as a swarm.

A review of references that applied RL to model BVR air combat behaviors is summarized in Table 12.1. The second column indicates the BVR air combat setup, such as one-versus-one (1v1), two-versus-two (2v2), or n-versus-one (nv1). The third column lists the RL methods used in each study. The fourth and fifth columns indicate whether the works employed functions to compute intermediate and final rewards, respectively.

In Weilin *et al.* (2018), an intermediate reward function was used based on angles between the aircraft, the distance relative to the WEZ, radar range from one aircraft's perspective, and altitude differences.

In Qiu *et al.* (2020), an intermediate reward considered angles between the aircraft, altitude relative to a safe threshold, and speed in relation to the minimum required to avoid a stall. The final reward penalized the agent if it was eliminated by the opponent at the end of the episode.

Piao *et al.* (2020) proposed an intermediate reward that assigned positive or negative values based on events such as missile launches, stall situations, radar tracking (both active and passive), missile warnings, and evasion maneuvers. The final reward reflected outcomes like eliminating the opponent, being eliminated, collisions, and victory or defeat conditions.

In Hu *et al.* (2021), an intermediate reward function was employed, considering factors such as the distance from the simulation boundaries, angles and distances between the aircraft, and the likelihood of eliminating the opponent or being eliminated.

Soleyman *et al.* (2021) proposed a final reward function focused solely on the final outcome, calculated as the difference in the number of surviving aircraft between the opposing swarms.

The study elaborated in Zhang *et al.* (2024) used an intermediate reward function based on the angles and distances between the aircraft.

In Wang *et al.* (2021), the authors introduced a two-stage maneuver control system:

TABLE 12.1 – Review of references that applied RL to model BVR air combat behaviors.

Reference	Air Combat Setup	RL Methods	Intermediate Rewards	Final Rewards
(WEILIN *et al.*, 2018)	1v1	Improved Q-Network (IQN)	✓	
(QIU *et al.*, 2020)	1v1	TD3	✓	✓
(PIAO *et al.*, 2020)	1v1	Key Air Combat Event Reward Shaping (KAERS) and PPO	✓	✓
(HU *et al.*, 2021)	1v1	Long Short-Term Memory (LSTM) and Deep Q-Network (DQN)	✓	
(SOLEYMAN *et al.*, 2021)	6v6	Neuroevolution-based Simulation Optimization (NSO)		✓
(ZHANG *et al.*, 2024)	1v1	Double Deep Q-Network (DDQN)	✓	
(WANG *et al.*, 2021)	nv1	Deep Deterministic Policy Gradient (DDPG)	✓	
(HU *et al.*, 2022)	2v1	DQN	✓	✓
(WANG; WEI, 2022)	1v1	Generative Adversarial Imitation Learning (GAIL) and Multi-Agent Deep Deterministic Policy Gradients (MADDPG)	✓	
(LI *et al.*, 2022)	1v1	SAC and Parallel Self-Play (PSP)	✓	
(HAN *et al.*, 2022)	4v4	Deep Relationship Graph Reinforcement Learning (DRGRL)	✓	✓
(LIU *et al.*, 2022a)	2v1	Multi-Agent Proximal Policy Optimization (MAPPO)	✓	✓
(ZHANG *et al.*, 2022)	1v1	Monte Carlo Tree Search (MCTS) and Deep Neural Network (DNN)		✓
(FAN *et al.*, 2022)	1v1	Asynchronous Advantage Actor-Critic (A3C)	✓	✓
(ZHANG *et al.*, 2022)	1v1	PPO	✓	✓
(JIANG *et al.*, 2022)	1v1	Dueling Double Deep Q-Network (D3QN)	✓	✓
(LI *et al.*, 2023)	1v1	Expert-Soft Actor-Critic (E-SAC)	✓	
(SCUKINS *et al.*, 2023)	1v1	PPO		✓
(ZHOU *et al.*, 2024)	2v2	Advantage Highlight Multi-Agent Proximal Policy Optimization (AHMAPPO)	✓	
(QIAN *et al.*, 2024)	1v1	Three-level Hierarchical decision framework embedding Expert knowledge (H3E)	✓	✓
(ZHAO *et al.*, 2024)	1v1	Self-play DRL	✓	✓

the first stage focused on swarm control, using metrics such as separation distance (for dispersion) and grouping distance (for cohesion); while the second stage handled engagement using reinforcement learning. Although the scenario involved a swarm versus a single opponent, the intermediate reward function evaluated each agent individually based on distances and angles relative to the opponent.

In Hu *et al.* (2022), the intermediate reward was based on angles and distances between the agent and the opponent without considering information from allied agents. The final reward assigned a positive value if the agent eliminated the opponent and a negative value (penalty) if the agent was eliminated.

The intermediate reward proposed in Wang and Wei (2022) considered factors like angles, altitude differences, and speed variations between the agent and the opponent.

In Li *et al.* (2022), the intermediate reward function was based on angles and distances between the agent and the opponent.

The intermediate reward function proposed in Han *et al.* (2022) focused on individual agents and excluded information from allies. It considered events such as radar tracking (active and passive), missile firing, and missile evasion. The final reward was defined as the difference between the number of surviving allies and opponents at the end of the episode.

Similar to other swarm-based studies, the intermediate reward function proposed in Liu *et al.* (2022a) evaluated each agent individually, considering angles, distances, speed, and altitude differences relative to the opponent. The final reward reflected events such as successful tracking, missile engagements, and elimination outcomes.

In Zhang *et al.* (2022), the final reward function assigned a value of 1 if the agent eliminated the opponent without being eliminated, -1 if the agent was eliminated, and 0 in the case of no eliminations or mutual elimination.

In Fan *et al.* (2022), the authors developed an intermediate reward that considered altitude, speed, distance, and angles between the agent and the opponent. The final reward assigned positive values for successful eliminations and negative values for defeats or ties.

Both intermediate and final rewards were explored in Zhang *et al.* (2022). The intermediate reward considered missile launches, angle relations, distances, and altitude differences, while the final reward assigned positive values for victories and negative values for defeats.

Jiang *et al.* (2022) applied a reward function that combined an intermediate reward, focused on distances and angles between the agent and the opponent, and a final reward, with positive values for eliminating the opponent, negative values for being eliminated, and zero for ties.

The intermediate reward function used in Li *et al.* (2023) was based on angles and distances between the agent and the opponent.

DRL was specifically applied in Scukins *et al.* (2023) to model defensive maneuvers against incoming missiles in 1v1 BVR air combat. The final reward was based on the effectiveness of defensive actions, assigning positive values for successful evasion (based on the final distance from the missile) and zero if the agent was hit.

An intermediate reward function was employed in Zhou *et al.* (2024) based on events like radar tracking and missile launches by individual aircraft. However, it did not account for the spatial distribution or formation orientation of aircraft within the swarms.

In Qian *et al.* (2024), intermediate reward was used for actions such as missile firing,

radar tracking, evading enemy tracking, and missile avoidance. The final reward assigned a value of 1 for eliminating the opponent or surviving a specific duration and -1 if the agent was eliminated.

Zhao *et al.* (2024) employed an intermediate reward function based on missile usage, distances, angles, and altitude differences. The final reward considered missile effectiveness, collision events, boundary violations, and navigation performance.

12.4 Methodology

This section presents the methodology for developing and evaluating our RL environment and agents, focusing on the AsaGym library, which supports fighter agent training in air combat scenarios within the Aerospace Simulation Environment (ASA) framework (DANTAS *et al.*, 2022a; DANTAS *et al.*, 2023a). It covers the integration of key modules, the design of observation and action spaces, the implementation of the reward function, and episode termination criteria. Additionally, it presents the scenario configurations, initialization sampling methods, and custom wrappers that enhance training flexibility. The methodology also includes rendering methods for validation and debugging, the training setup with selected algorithms and hyperparameters, and the computational infrastructure. Finally, evaluation metrics are introduced to determine the most effective algorithm choices based on performance indicators such as episode rewards, episode lengths, and total training time.

12.4.1 AsaGym

The AsaGym computational library is essential to our methodology, providing a custom environment built on top of the Gymnasium framework specifically for training fighter agents using DRL. This subsection details the development and integration of AsaGym within ASA. AsaGym extends the capabilities of Gymnasium by introducing specialized modules that facilitate the training of Artificial Intelligence (AI) agents in complex air combat scenarios.

12.4.1.1 Library Development

The open-source Python module Gymnasium (TOWERS *et al.*, 2023) has emerged as a significant tool for RL researchers, replacing the widely used OpenAI Gym (BROCKMAN *et al.*, 2016). Gymnasium builds upon the foundation laid by OpenAI Gym, offering a robust and flexible framework for developing and comparing RL algorithms.

Gymnasium provides a comprehensive set of pre-built environments that are fully compatible with its API (TOWERS *et al.*, 2023). These environments cover a wide range of RL tasks, from classic control problems to more complex scenarios involving robotics, games, and simulations (BROCKMAN *et al.*, 2016). By standardizing the interface for these environments, Gymnasium ensures that researchers can seamlessly switch between different tasks without needing to modify their underlying RL algorithms (TOWERS *et al.*, 2023).

One of the key advantages of Gymnasium is its ability to streamline the development process of RL algorithms. Researchers can leverage the pre-built environments to quickly test and refine their algorithms, focusing on improving performance and exploring new techniques. This modularity and ease of use significantly reduce the overhead associated with setting up and managing different RL tasks (TOWERS *et al.*, 2023).

A Python package for the ASA framework, the AsaGym, was created to employ RL training for fighter agents. This package includes a subclass of the Gymnasium environment class to facilitate the training of new AI agents with RL. The library also offers unique wrappers to help in training setup configuration and methods for registering special environments to be recognized by the Gymnasium.

We utilize the Stable-Baselines3 Python module, a collection of RL algorithm implementations with a common interface, to test and verify the AsaGym library. These algorithms adhere to a standardized structure, using a unified interface that includes functions such as `train`, `save`, `load`, and `predict` (RAFFIN *et al.*, 2019). Each algorithm has its limitations, especially regarding support for discrete or continuous action spaces. Additionally, the Stable-Baselines3 package provides the capability to multiprocess multiple separate contexts into a single environment, accelerating the training process for supported algorithms.

12.4.1.2 Modules Integration

This work aims to use the ASA framework modules previously created in the background to construct a Python package that subclasses the Gymnasium main class. The modules utilized will be the Tacview recorder for rendering the environment, a specific AsaNode for processing the simulation steps and interacting with AsaGym, and the dynamically loaded models.

The set of BVR air combat behaviors of a fighter agent in the ASA framework is modeled through a Behavior Tree (BT). A simplified version of the fighter's BT is presented in Figure 12.1. BT is a formal language for the graphical and hierarchical representation of logical processes, i.e., behaviors (COLLEDANCHISE; ÖGREN, 2020). Fighter's BT is composed of six main branchs: five subtrees that handle collision situations, threat reac-

tions (break, crank, etc), situations to return to base (bingo fuel, damage, etc), combat air patrol, and navigation in formation; and a branch that handles three specific BVR phases: engagement; missile shooting; and missile supporting. The condition node of this branch checks whether the autonomous agent has a target, which is a requirement for the activation of one of the three mentioned phases. A detailed description of the engagement phase can be found in (DANTAS *et al.*, 2021a). It is important to mention that subtrees encapsulate complex behaviors composed of a combination of many other nodes. Due to the large size of the fighter's behavior tree, subtrees were used to make this tree easier to visualize in the article.

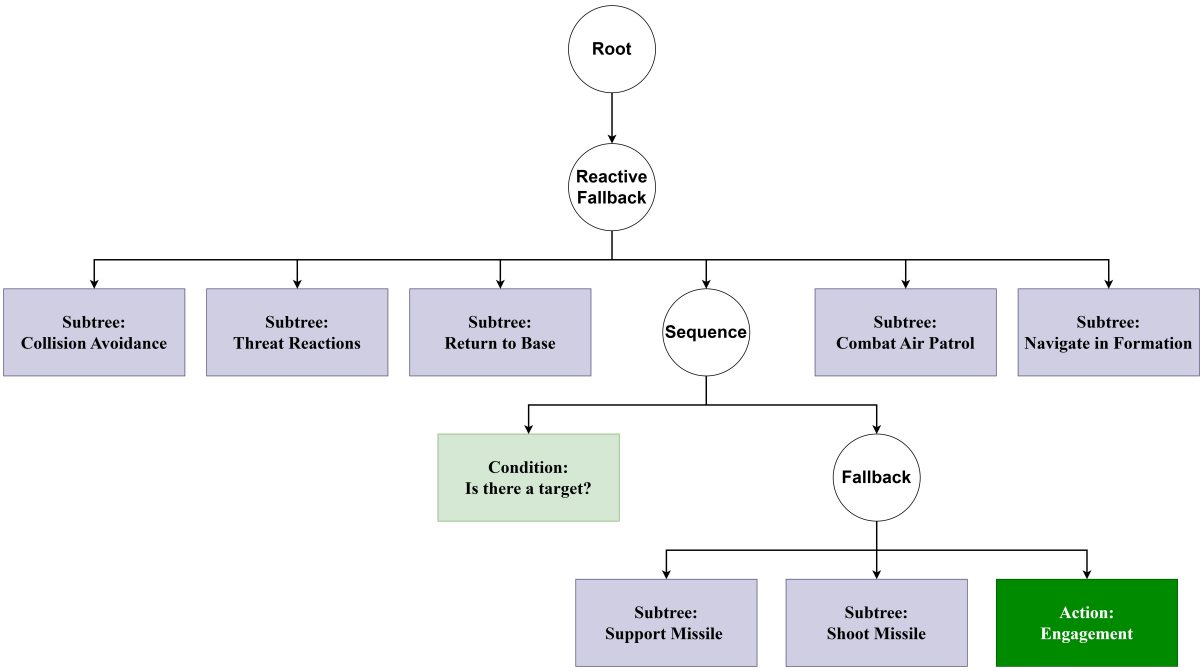


FIGURE 12.1 – Simplified BT of the fighter agent.

Initially, the engagement node of the fighter's BT was modified to respond to commands from an external software component rather than execute a predefined function. The goal is to train a specific behavior of the flying agent to locate a suitable spot to take a shot in a BVR conflict, as discussed in the following section. All other behaviors will still be dictated by the BT nodes already implemented. An ASA extension, the External Processor, was created to properly handle Input/Output (IO) messages. It initiates a high-performance asynchronous communication socket called ZeroMQ (ZMQ) and uses it to convey messages serialized using the Protobuf library (GOOGLE, 2023). We customized the AsaNode with an RL server capable of answering many queries required by the Gymnasium Application Programming Interface (API). This server established a second ZMQ connection, which AsaGym will connect to for message exchange with our package. Additionally, the RL server starts, resets, and ends the node's execution.

12.4.1.3 Communication and Sequence Diagram

The communication between the AsaGym library and the related objects is depicted in a Unified Modeling Language (UML) sequence diagram for the `step` method in Figure 12.2. This diagram visually represents the interactions and message flow between the various components involved in the RL training process. In the diagram, closed arrowheads imply synchronous messages, indicating that the sender waits for a response before proceeding. In contrast, open arrowheads denote asynchronous messages, where the sender continues its process without waiting for a response.

Dashed lines in the diagram represent responses, illustrating the return flow of information from the receiver back to the sender. Filled lines indicate method calls, showing the initiation of actions or data transfers between components. The specific methods used to write and read at the ZMQ connection are `zmq_send` and `zmq_recv`, respectively. These methods facilitate high-performance communication by efficiently managing the transmission and reception of serialized messages within the network.

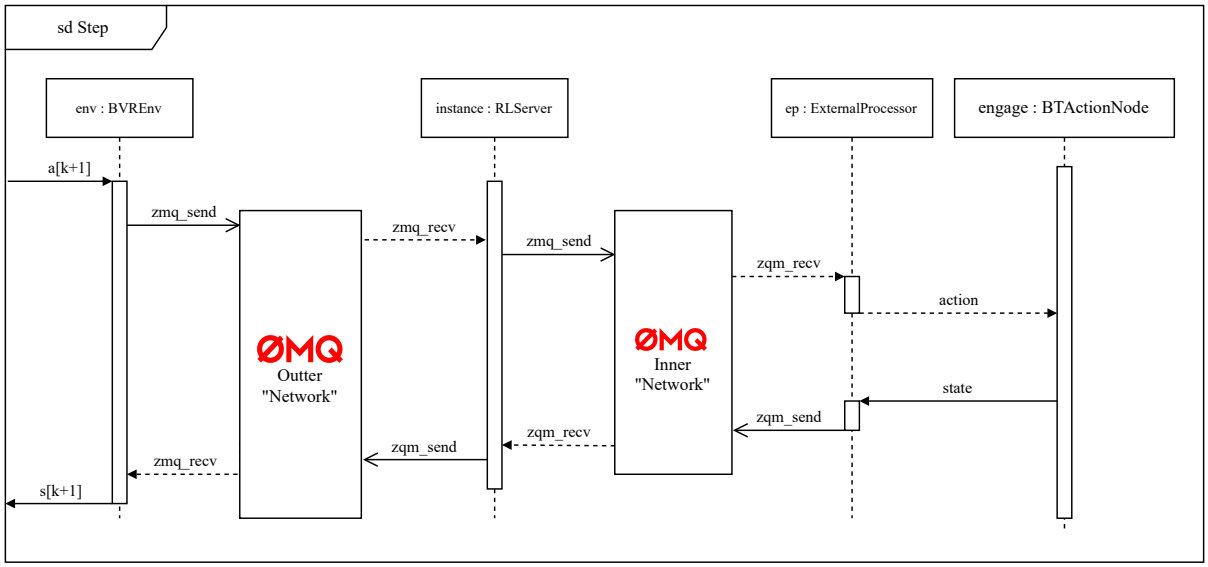


FIGURE 12.2 – UML sequence diagram illustrating the messaging interactions in the `step` method of the AsaGym library using ZMQ for high-performance communication.
Source: Gobi and Botelho (2023)

Using the same concept, message-exchanging mechanisms for setting up, clearing, and shutting down the environment were implemented. These mechanisms ensure seamless RL training session initialization, maintenance, and termination.

As presented in Figure 12.3, the engagement node of the BT communicates cyclically with a DRL model in AsaGym via the External Processor. The engagement node sends the agent's state to the DRL model, which processes this state to generate an observation. This observation serves as the input to the deep neural network of the DRL model, which

then estimates an action. The estimated action is sent back to the engagement node, which executes it through the fighter agent. After performing the action, the agent reaches a new state that is sent back to the DRL model, where it is processed into a new observation.

If the new state benefits the agent, a positive reward is assigned. If the state is unfavorable, a negative reward is given. During training, the RL model adjusts the weights of its neural network based on these rewards, refining its policy to maximize future rewards. A detailed explanation of state, observation, action, and reward is provided throughout this section.

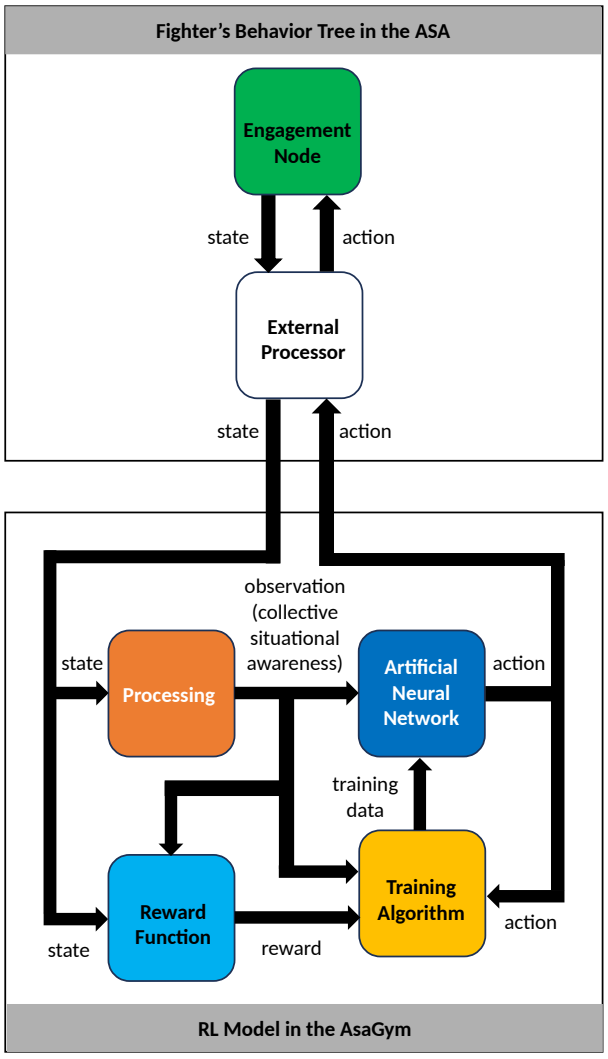


FIGURE 12.3 – RL model in AsaGym integrated with the fighter's BT in ASA, showing the interactions between situational awareness, the reward function, and decision-making processes.

12.4.2 Scenario Description

In this study, we consider simulations of three-dimensional BVR air combat scenarios. Each fighter agent is a computational model of the F-16 aircraft in the ASA framework, and is equipped with: datalink; six active BVR missiles with seekers; a radar; and a Radar

Warning Receiver (RWR) (COSTA *et al.*, 2025). Each fighter agent has a BT that emulates BVR air combat behaviors. The training of the DRL model involves engaging a team of two fighters against an opposing team of two aircraft in BVR air combat. In this type of combat, pilots must position themselves strategically to gain an offensive advantage while avoiding enemy tracking (DANTAS *et al.*, 2023).

This study explores a more complex multi-agent environment by introducing 2v2 BVR air combat. In this scenario, two allied aircraft engage two enemy aircraft, requiring coordination and teamwork to achieve tactical objectives.

Two distinct setups are considered, as illustrated in Figure 12.4. In **Setup 1**, the blue team consists of two agents: the Blue Leader, whose engagement node is controlled by a DRL model, and the Blue Wingman, whose engagement node follows engagement rules based on operational expert knowledge. They face an opposing red team composed of two enemies, both of which use engagement rules identical to those controlling the Blue Wingman. This setup challenges the Blue Leader to coordinate with its wingman to effectively engage a reactive opponent.

In **Setup 2**, both blue team agents, Blue Leader and Blue Wingman, have engagement nodes controlled by the same DRL model. Each agent has its own state, but receives a shared observation (collective situational awareness) that includes information about both allies and opponents. This setup allows the use of a single shared policy while providing each agent with a collective view of the scenario. It supports the learning of cooperative behavior and helps evaluate how a unified policy improves team coordination and performance against the red team. Each fighter agent of the red team follows the same engagement rules as in the first setup.

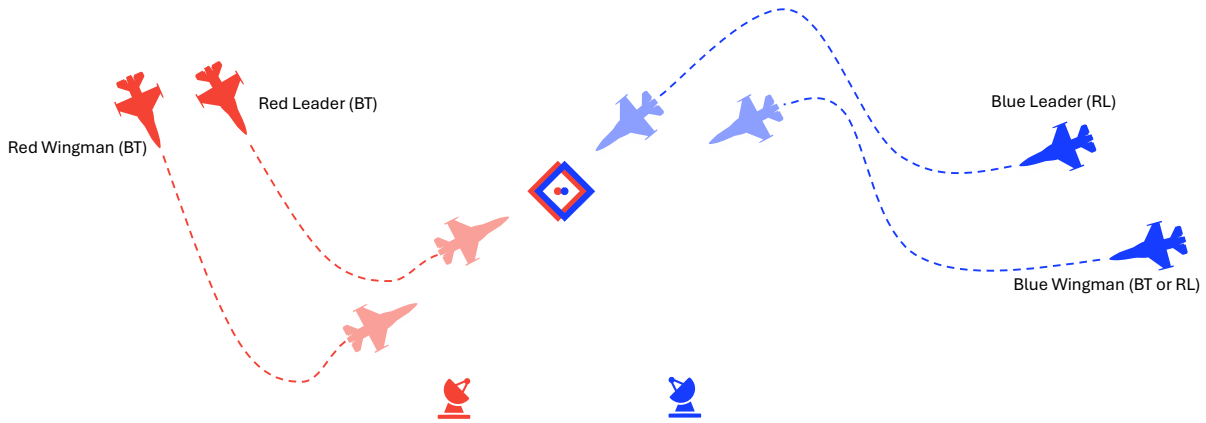


FIGURE 12.4 – Diagram illustrating the positions and roles of the Blue and Red teams in the exercise. In Setup 1, the Blue Leader has an engagement node controlled by a DRL model, while the Blue Wingman has an engagement node controlled by engagement rules. In Setup 2, both the Blue Leader and Blue Wingman have engagement nodes controlled by a single DRL model. Each team is responsible for defending a common point of interest and is equipped with ground radar to enhance enemy detection.

By expanding the scenario to include multi-agent interactions, this study enhances

the robustness and versatility of the trained models, preparing them for various combat situations. In both setups, each team is tasked with defending a common point of interest and is equipped with ground radar to improve enemy detection.

12.4.3 Learning Framework

This subsection describes the learning framework adopted in this study. It is organized into three parts: the observation space provided to the agents, the set of available actions, and the reward function designed to guide the learning process.

12.4.3.1 Observation

Observation is the input of the artificial neural network of the DRL model, as we can verify in Figure 12.3. In this work, observation is the collective situational awareness between the Blue and Red teams, as summarized in Table 12.2. This collective situational awareness is obtained through the processing of the estimated state of each agent.

TABLE 12.2 – State and observation components for blue and red teams.

Category	**Subcategory**	**Components**
State	Blue Leader	Latitude, Longitude, Altitude, Heading, and Airspeed
	Blue Wingman	
	Red Leader	
	Red Wingman	
Observation	(Blue Leader , Red Leader)	Magnitude Difference of Relative Azimuths, Slant Distance, Airspeed Difference
	(Blue Leader , Red Wingman)	
	(Blue Wingman , Red Leader)	
	(Blue Wingman , Red Wingman)	
	(Blue Leader , Blue Wingman)	

The use of such collective situational awareness, which is based on angular relations, slant distances and airspeed differences, plays an important role in training the DRL model. It makes the DRL model invariant to the specifications of positions and velocities. To enhance the learning process, all observations are normalized within predefined ranges, as this is a common practice in RL training that helps improve training efficiency and convergence.

12.4.3.2 Action

The action is designed to capture essential flight control elements, allowing the agent to maneuver effectively during BVR engagements. It includes key flight parameters that influence combat performance, such as **heading** and **airspeed**. Unlike real-world pilots,

who have full control over their aircraft, RL algorithms often benefit from a simplified action structure. In the ASA framework, the actions do not represent direct control commands for the aircraft but rather a desired attitude that the system aims to achieve.

The elements included in the action are:

- **Heading**: The direction in which the aircraft's nose is pointing, expressed in degrees, ranging from `-180` to `180`.
- **Airspeed**: The aircraft's speed along its flight path.

The action is structured as a dictionary, where each action parameter is represented as a constrained continuous value. The heading action allows for full directional control, while the airspeed action lets the agent adjust velocity to optimize engagement strategies. The decision to use a continuous action space improves maneuver precision compared to discrete alternatives. This setup enables the agent to explore a broader range of tactical options and adapt more effectively to different combat scenarios.

12.4.3.3 Reward

Before defining the reward function used in this study, we conducted preliminary tests with alternative formulations inspired by existing literature. These initial experiments focused on individualistic reward strategies, such as encouraging agents to minimize their distance to opponents or maximize angular advantage individually. However, such strategies often resulted in undesired behaviors, including agents pursuing conflicting goals, clustering in suboptimal areas, or failing to coordinate effectively with teammates. These observations motivated the adoption of a shared reward formulation that promotes collective behaviors and situational awareness across the team.

The proposed reward function is designed to encourage agents to adopt collectively advantageous tactical behaviors in BVR air combat. The definitions of advantageous and disadvantageous situations in BVR scenarios were informed by the experience of fighter pilots.

To achieve these behaviors, we propose a reward function that incorporates collective situational awareness among agents on both the Blue and Red teams. Specifically, the reward is composed of two primary components: **Relative Azimuth Reward** and **Velocity Reward**. Each of these components is designed to promote different aspects of engagement strategy, as detailed below.

- **Relative Azimuth Reward**: The idea behind this metric comes from a fundamental aspect of air combat: a pilot always tries to keep their aircraft pointed in a tactically

favorable direction relative to the enemy. In practice, this means constantly adjusting position and heading to face the opponent, making it easier to react, pursue, or defend. This reward is important for achieving favorable positioning, as maintaining alignment with the target enhances maneuverability and increases the effectiveness of weapon deployment.

It evaluates the angular alignment between the agents of both teams by considering their relative azimuth values. The relative azimuth of an agent is calculated as the difference between the bearing to the target and its own heading. Since each agent has its own frame of reference, the relative azimuth from the perspective of one agent may differ from that of the opposing agent. The reward function is designed to minimize differences in orientation between allied and enemy agents, encouraging the Blue team to maintain an optimal orientation toward the Red team. The calculated values are normalized using the maximum allowed azimuth difference to ensure they fall within the range $[-1.0, 1.0]$. This reward is important for achieving favorable positioning, as maintaining alignment with the target enhances maneuverability and increases the effectiveness of weapon deployment. The relative azimuth reward is defined by:

$$\text{reward}_{ra} = \sum_{i=1}^{n_B} \sum_{j=1}^{n_R} \frac{|ra(R_j, B_i)| - |ra(B_i, R_j)|}{n_B n_R \, ra_{\max}} \tag{12.1}$$

where n_B and n_R are the number of agents of the Blue and Red teams, respectively; B_1 represents the Blue Leader; B_2 represents the Blue Wingman; R_1 represents the Red Leader; R_2 represents the Red Wingman; $ra(A, B) \in [-180°, 180°]$ represents the relative azimuth (in degrees) of agent A to agent B, which is computed as:

$$ra(A, B) = \text{bearing}(B \to A) - \text{heading}(A) \tag{12.2}$$

where $ra_{\max} = 180°$ is the maximum value of the relative azimuth.

- **Velocity Reward**: This metric is inspired by how pilots control their speed in relation to an enemy during an engagement. In a real combat scenario, a pilot needs to approach the opponent at a speed that allows for effective action but also keeps a safe and manageable distance. If the agent stays too far, it risks losing the advantage; if it gets too close too quickly, it may compromise safety or tactical position. This component ensures that the agent does not remain too far from the target, which could compromise combat effectiveness.

  Let B_i be an agent of the Blue team and R'_i the nearest agent of the Red team in relation to B_i. Let $v(B_i)$ and $v(R'_i)$ denote the airspeeds of B_i and R'_i, respectively, where airspeed is a scalar representing the magnitude of the agent's velocity. The

velocity factor $F_v(B_i, R'_i)$ is computed based on the airspeeds of these agents, and $dv_{\max}$ is the maximum allowed velocity difference. The velocity reward encourages a single agent to close the distance with the nearest target while maintaining a tactically safe range. It is calculated based on the slant distance between the agent and the target, which is derived from their respective positions and altitudes. The reward is normalized to fall within the range $[0.0, 1.0]$, with higher values awarded for shorter distances within an acceptable operational envelope. This component ensures that the agent does not remain too far from the target, which could compromise combat effectiveness. The velocity reward is defined by:

$$R_v = \sum_{i=1}^{n_B} \frac{F_v(B_i, R'_i)}{n_B \, ra_{\max}} \left(|ra(R'_i, B_i)| - |ra(B_i, R'_i)| \right), \tag{12.3}$$

which applies only if:

$$\frac{|ra(R'_i, B_i)| - |ra(B_i, R'_i)|}{ra_{\max}} > 0.0, \tag{12.4}$$

otherwise:

$$R_v = 0. \tag{12.5}$$

The velocity factor $F_v(B_i, R'_i)$ is given by:

$$F_v(B_i, R'_i) = \begin{cases} 0, & \text{if } \Delta v < 0 \\ 1, & \text{if } \Delta v > 1 \\ \Delta v, & \text{otherwise} \end{cases} \tag{12.6}$$

where:

$$\Delta v = 1 - \frac{dv_{\max} - (v(B_i) - v(R'_i))}{dv_{\max}}. \tag{12.7}$$

The total reward is computed as the sum of the normalized Relative Azimuth Reward and Velocity Reward, ensuring a balanced evaluation of both positioning and engagement distance. The reward function dynamically adapts to the scenario by accounting for real-time changes in the relative positions and headings of the agents and targets.

12.4.4 Initialization Sampling

The scenario setup defines the initial conditions for both friendly and enemy agents, allowing for variability in positioning and airspeed. The initialization process is imple-

mented through a custom function, which sets the starting parameters for the agents based on a fixed reference point. The Red Leader serves as the reference for positioning the Blue Leader. The initial latitude and longitude of the Red Leader are fixed, with a constant heading of 180° and airspeed of 450 knots.

The Blue Leader is always initialized directly in front of the Red Leader, ensuring that the two agents start head-on. The distance between them is randomly selected within a range of 20 to 60 nautical miles. The Blue Leader's airspeed is assigned with a random value within [250,650] knots.

To determine the initial positions, the ground distance, which is the horizontal separation between agents measured along the Earth's surface, and the bearing, which is the direction from one agent to another measured in degrees clockwise from true north, are used to compute the Blue Leader's latitude and longitude. The bearing is set to match the Red Leader's heading, ensuring that both agents are aligned head-on, while the Blue Leader's heading is fixed at 0°. Both agents start the engagement at an altitude of 25,000 feet, simulating a realistic operational environment for BVR air combat.

Once the initial state is set, the formation pattern for the blue team is configured. In this formation pattern, the Blue Wingman is positioned 180 degrees relative to the Blue Leader at a distance of approximately 1.6 nautical miles. The Blue Wingman is also assigned the same heading as the Blue Leader. The Red Wingman is positioned alongside the Red Leader, following a similar formation strategy. The complete initialization sampling process is summarized in Algorithm 1. A visualization of this initialization process is presented in Figure 12.5.

Algorithm 1 Initialization sampling for 2v2 BVR air combat

1: Set fixed initial position, heading (180°), airspeed (450 knots), and altitude (25,000 feet) for Red Leader
2: Determine random distance from the Red Leader within [20, 60] NM to set the Blue Leader's position directly in front
3: Compute the Blue Leader's latitude and longitude using geodesic calculations with bearing equal to the Red Leader's heading
4: Set the Blue Leader's heading to a fixed value (0°)
5: Assign the Blue Leader's altitude as 25,000 feet
6: Assign the Blue Leader's airspeed with a random value within [250,650] knots.
7: Set the Blue Wingman 180 degrees relative to Blue Leader at a distance of 1.6 NM, maintaining the same heading
8: Set the Red Wingman alongside the Red Leader in a similar formation
9: Return the initial configuration with positions, headings, airspeeds, and formation details

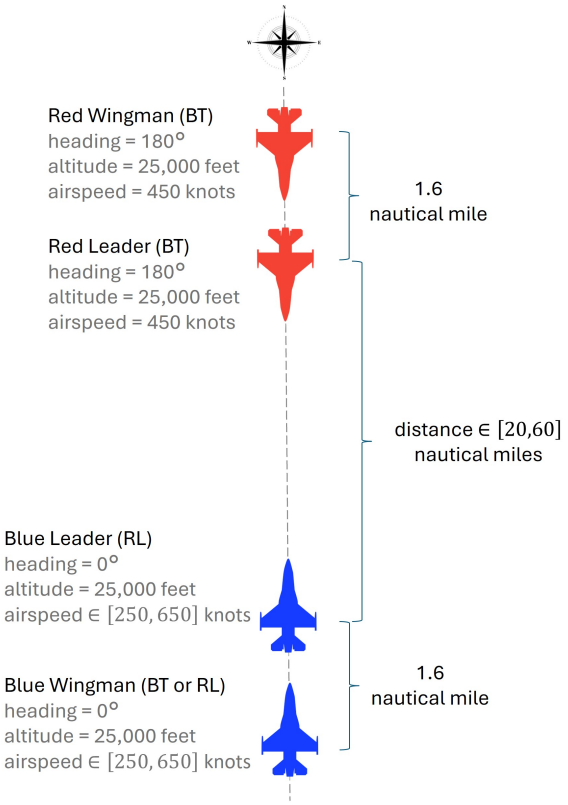


FIGURE 12.5 – Visualization of the initialization sampling process.

12.4.5 Episode Termination Criteria

The criteria for episode termination are important in training agents in the 2v2 BVR air combat environment, as they facilitate the training of specific aspects of combat, such as engagement scenarios, ensuring that engagements are tactically meaningful.

An episode ends when one of these conditions is met:

- **Aircraft Elimination**: The episode will conclude if at least one aircraft from either the blue or red team is eliminated. This condition ensures the integrity of the combat scenario, as the elimination of an aircraft would alter the dynamics of the engagement, effectively making it no longer a 2v2 scenario.
- **Priority-Based Termination Condition**: An episode will terminate if a BT node with a priority level greater than the priority of the engagement node is activated.

By defining structured termination criteria, the training environment ensures that each episode concludes with meaningful tactical outcomes. These criteria not only provide agents with well-defined objectives but also enable focused evaluation of specific behaviors and decision-making processes.

12.4.6 Rendering Methods

For validation and debugging purposes, visualizing the evolution of the agent's behavior is a key component of the development process. A practical approach to achieve this is through the use of the commercial software Tacview (RAIA SOFTWARE, 2024), which provides 3D visualizations of air combat scenarios. The ASA framework integrates seamlessly with Tacview by exposing all model attributes to a local port, enabling real-time rendering of the scenario. However, Tacview's streaming mode expects a continuous flow of temporal data, whereas AsaGym resets the environment's time at each episode. This episodic reset creates time discontinuities that Tacview cannot properly handle due to its serialized data collection, preventing its direct use for real-time visualization during training.

To address this limitation, each episode is saved as a separate `.acmi` file, Tacview's native format for recordings. This approach allows developers to review individual episodes offline, enabling a detailed post-training analysis of the agent's behavior and decision-making. By storing episodes in this manner, it becomes possible to thoroughly examine specific maneuvers and engagement strategies without interfering with the training process. The use of Tacview extends beyond basic debugging by providing a framework for performance evaluation. Its intuitive interface and robust visualization tools make it an invaluable resource for refining reinforcement learning models and ensuring that agents are prepared for realistic air combat scenarios.

12.4.7 Wrappers

The Gymnasium documentation (TOWERS *et al.*, 2023) states that wrappers provide a simple and flexible way to modify existing environments without changing the core code. In AsaGym, wrappers keep the main environment intact while allowing customization to meet the specific needs of air combat training. This section explains the custom wrappers developed for the AsaGym library, expanding the features of the Gymnasium framework.

12.4.7.1 Flatten Action

The `FlattenAction` wrapper simplifies the multi-dimensional action space of the environment by converting it into a flat, continuous vector representation. This transformation helps the agent by reducing complexity while preserving essential control information. The `FlattenAction` wrapper leverages the `ActionWrapper` from the Gymnasium framework to create a simplified interface for action processing. This allows RL algorithms to output a vector of continuous values, which the wrapper then maps to the appropriate control

inputs for the environment.

12.4.7.2 Skip Frame

A wrapper that maintains the same action for multiple frames has been implemented to train a more stable agent and emphasize the impact of each action. This technique was first introduced by DQN (MNIH *et al.*, 2015) to improve learning stability by reducing temporal aliasing and increasing the effective action duration. It is particularly effective in environments like AsaGym, where the ASA simulation processes commands every 100 milliseconds. By skipping frames, the wrapper ensures a stronger temporal influence on actions by accumulating all intermediate rewards and returning a cumulative reward at the end of the skipped frame block. The returned observation reflects the most recent state of the environment, ensuring accurate decision-making.

The `SkipFrameWrapper` takes a parameter defining the number of frames to be skipped. In this work, we are skipping 49 frames, which extends the interval between action decisions to approximately 5 seconds. This duration was chosen to prevent abrupt changes in actions at high frequencies, allowing sufficient time for the agent to execute a selected maneuver before making a new decision.

As previously described in Subsection 12.4.3.2, the actions in AsaGym do not represent direct control commands for the aircraft but rather a desired attitude that the system aims to achieve. Because of this, the low sampling rate introduced by the frame-skipping mechanism does not hinder control precision. Instead, it allows the agent to focus on meaningful tactical decisions rather than frequent low-level adjustments as the system smoothly transitions toward the commanded state.

12.4.8 Training Setup

For training the agents in the proposed scenarios, we employed four DRL algorithms: PPO (SCHULMAN *et al.*, 2017), A2C (MNIH *et al.*, 2016), SAC (HAARNOJA *et al.*, 2018), and TD3 (FUJIMOTO *et al.*, 2018). These algorithms were implemented using the Stable-Baselines3 framework (RAFFIN *et al.*, 2019).

The selection of these algorithms was based on balancing *on-policy* and *off-policy* learning methods, as well as their ability to handle continuous action spaces. Table 12.3 summarizes the main characteristics of each algorithm.

The default hyperparameter values from Stable-Baselines3 were used, with adjustments to the step-related parameters and `batch_size`. Specifically, for PPO and A2C, the `n_steps` parameter was set to 128, while for SAC and TD3, the `train_freq` parame-

TABLE 12.3 – Comparison of selected RL algorithms.

Algorithm	Policy Type	Exploration Strategy	Key Characteristics
PPO	On-policy	Stochastic (policy noise)	Uses a clipped objective to prevent large policy updates, improving training stability. Efficient for large-scale training with parallel environments.
A2C	On-policy	Stochastic (policy noise)	Actor-critic method where the policy is guided by value estimates. More sample efficient than vanilla policy gradient but lacks a mechanism to limit updates, making it more sensitive to hyperparameters.
SAC	Off-policy	Stochastic (entropy maximization)	Encourages exploration through entropy regularization, improving stability in continuous action spaces. Uses experience replay and separate Q-value and policy networks.
TD3	Off-policy	Deterministic (target policy smoothing)	Addresses overestimation bias in DDPG with twin Q-value networks, delayed policy updates, and target smoothing. Improves stability and consistency in continuous action spaces, making it effective for real-world applications.

ter was set to 128. These particular values were chosen through a random search process over traditional powers of 2 (e.g., 32, 64, 128, 256). This choice influences the trade-off between sample efficiency and update frequency. Higher values for step-related parameters allow the agent to accumulate more experience before updating, which helps stabilize policy learning by reducing gradient noise. However, it also delays policy updates, potentially slowing down adaptation to new situations. Smaller `batch_size` values lead to more frequent updates, while larger values help smooth the gradient updates at the cost of increased computational demand. Table 12.4 summarizes the selected hyperparameters and their roles.

TABLE 12.4 – Selected hyperparameters for each RL algorithm. The remaining parameters follow the default values from Stable-Baselines3.

Algorithm	train_freq (Off-Policy)	n_steps (On-Policy)	batch_size	Explanation
PPO	–	128	128	Collects 128 steps per environment before updating the policy and optimizes in minibatches to improve training stability.
A2C	–	128	–	Collects 128 steps per environment before updating the policy in a single batch, using a synchronous update mechanism.
SAC	128	–	128	Updates every 128 steps using experience replay and entropy regularization to improve exploration.
TD3	128	–	128	Updates every 128 steps using a replay buffer and twin critics to reduce overestimation bias in Q-values.

Each algorithm was trained five times with different random seeds in both setups, Setup 1 (RL + BT) and Setup 2 (RL + RL), to ensure statistical robustness while maintaining computational feasibility. This choice follows common practice in RL, where a small number of seeds, typically five, is widely adopted to report average performance with acceptable variance (HENDERSON *et al.*, 2018; ANDRYCHOWICZ *et al.*, 2020; HAARNOJA *et al.*, 2018).

Training was conducted for 500,000 steps per seed, using 16 parallel environments to accelerate learning. Model checkpoints were saved every 100,000 steps for evaluation and debugging, while performance metrics such as episode rewards, loss values, and training time were logged using TensorBoard. All experiments were run on a system with an AMD Ryzen 9 5900X processor (12 cores, 24 threads, 3.70 GHz) and 64 GB of RAM, enabling efficient parallel execution of training environments.

12.4.9 Evaluation Metrics

To assess the performance of the trained agents in the air combat scenarios, four primary evaluation metrics were considered: average episode reward; mean episode length; and total training time for each algorithm.

- **Average Episode Reward:** This metric measures the cumulative reward achieved by the agent over an episode, averaged across multiple runs. The moving average is computed based on the number of episodes completed within each policy update. Since PPO is trained with $n\_steps = 128$ and $num\_envs = 16$, each update aggregates data from $128 \times 16 = 2048$ environment steps. The number of completed episodes within this window determines the averaging scope. A higher average reward indicates improved decision-making and maneuver execution.
- **Mean Episode Length:** The mean episode length represents the average number of steps the agent takes before an episode terminates. This metric provides insight into the agent's survivability and engagement effectiveness. Shorter episodes may indicate either successful engagements or premature termination due to poor tactical positioning, while longer episodes may suggest prolonged engagements with effective maneuvering.
- **Total Training Time:** This metric captures the actual wall-clock time required to train the agents using each algorithm and is reported in hours. As all experiments were run on the same machine, differences in training time reflect algorithmic efficiency rather than hardware differences.

12.5 Results and Discussion

This section presents the results obtained from training the RL agents in the two proposed setups. The evaluation focuses on three key metrics: average episode reward, mean episode length, and total training time. For all reported results, both the mean and standard deviation were computed over the five independent training runs. In the plots

for average episode reward and mean episode length, the solid line represents the mean value, while the shaded region indicates the standard deviation. Additionally, qualitative analyses of the agent's behavior in both configurations are provided to illustrate the tactical maneuvers learned.

12.5.1 Reward and Episode Length Analysis

The following figures illustrate the learning progress of each algorithm over the training steps for both configurations proposed.

Figure 12.6 shows the evolution of the average episode reward in Setup 1. PPO demonstrates the highest reward convergence, followed by SAC, which stabilizes at a lower reward. TD3 exhibits instability, with a decline after an initial increase, which likely occurred because the agent made some random favorable choices early in training. A2C struggles to improve significantly, maintaining the lowest reward throughout training.

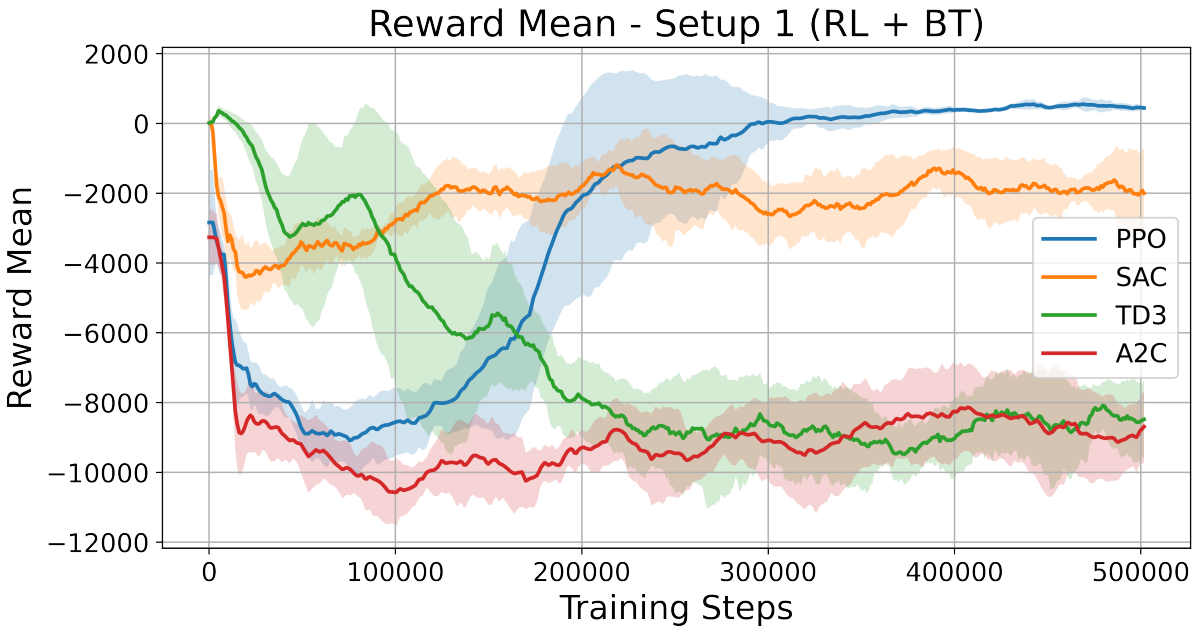


FIGURE 12.6 – Average episode reward for PPO, SAC, TD3, and A2C in Setup 1 (RL + BT).

Figure 12.7 presents the reward evolution in Setup 2. Interestingly, PPO reaches its stability region even faster than in Setup 1, with a slightly higher average reward, demonstrating strong adaptability to the more complex environment. In contrast, the other algorithms show worse performance compared to Setup 1, likely due to the increased difficulty of controlling two agents simultaneously. SAC performs better than TD3 and A2C but remains below PPO in terms of average reward and takes longer to stabilize. TD3, which already showed instability in Setup 1, performs even worse in Setup 2, exhibiting strong fluctuations and failing to improve. A2C remains the weakest, showing stable behavior but with consistently low performance and no meaningful progress.

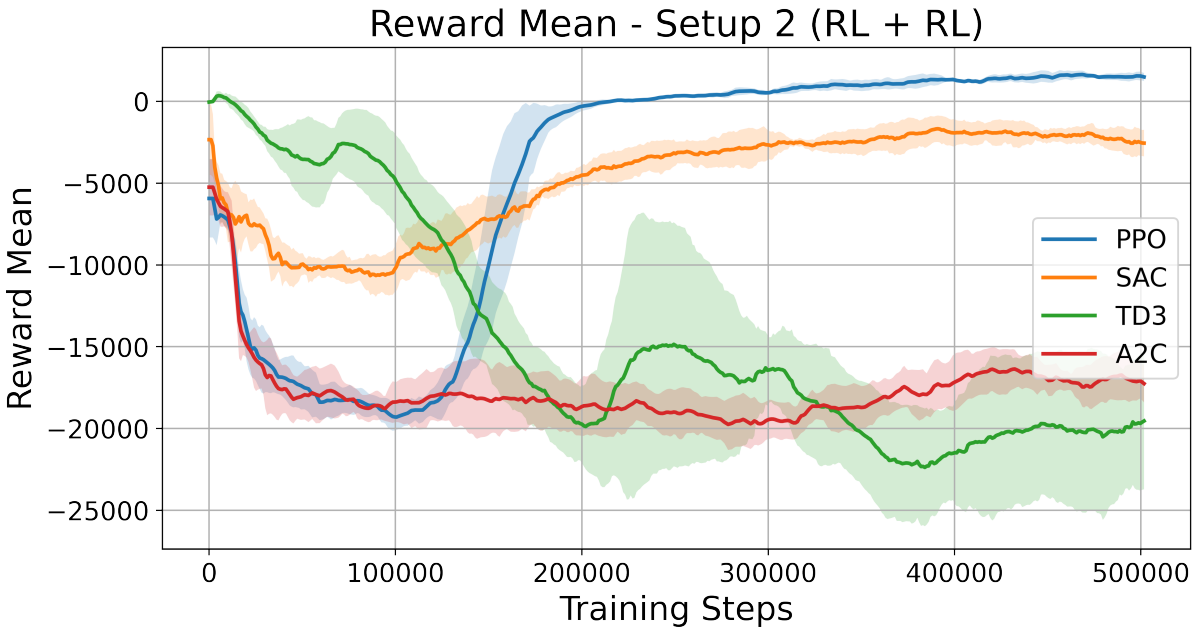


FIGURE 12.7 – Average episode reward for PPO, SAC, TD3, and A2C in Setup 2 (RL + RL).

Previous studies have shown that PPO and SAC frequently outperform other methods in continuous control tasks (SCHULMAN *et al.*, 2017; HAARNOJA *et al.*, 2018). Given this, it was expected that PPO and SAC would perform better than TD3 and A2C in this study.

Figure 12.8 presents the mean episode length over training for Setup 1. PPO and SAC stabilize with shorter episode durations, indicating faster and more decisive engagements. TD3 shows a slower improvement compared to A2C, but its episode length gradually increases, reaching values slightly below A2C. A2C maintains high episode lengths throughout training, reinforcing its weaker performance.

Figure 12.9 illustrates the episode length trends for Setup 2. Overall, the episode lengths increased compared to Setup 1, which was expected given the higher complexity of this scenario where two agents must be controlled simultaneously. PPO still stabilizes at shorter episode lengths, but the gap between PPO and SAC has decreased, possibly because the more dynamic environment reduces PPO's relative advantage in quickly reaching efficient strategies, allowing SAC to perform closer to PPO.

Notably, the difference between TD3 and A2C has increased, with TD3 now exhibiting longer episode durations than A2C. This shift may suggest that TD3 struggles more with policy convergence in the presence of another learning agent. On the other hand, A2C, despite its generally weak performance, maintains more consistent episode lengths, likely because its on-policy nature limits drastic fluctuations even if it fails to find optimal strategies.

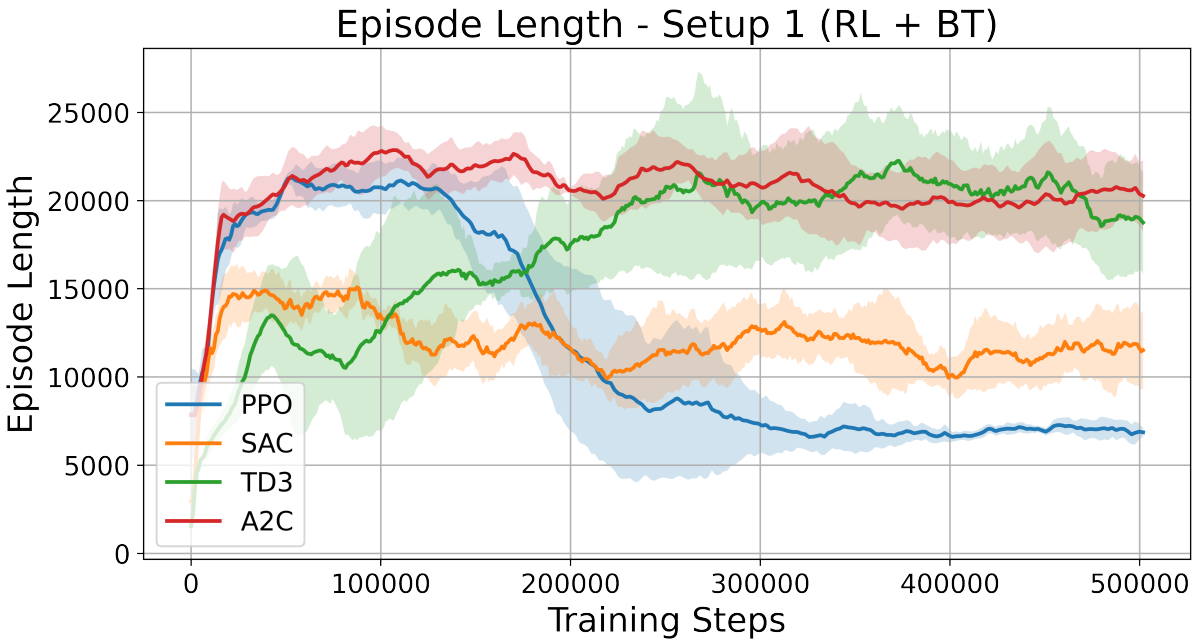


FIGURE 12.8 – Mean episode length for PPO, SAC, TD3, and A2C in Setup 1 (RL + BT).

12.5.2 Training Time Comparison

The total training time for each RL algorithm was measured in hours under both configurations. Table 12.5 summarizes the training duration required for PPO, SAC, TD3, and A2C algorithms. These values reflect the computational efficiency of each approach and provide a basis for assessing the trade-off between training speed and model performance.

TABLE 12.5 – Training time comparison for algorithms in different setups.

Setup	Algorithm	Training Time (hours)
Setup 1 (RL + BT)	PPO	6.77 ± 0.57
	SAC	5.72 ± 0.26
	TD3	4.74 ± 0.24
	A2C	4.10 ± 0.25
Setup 2 (RL + RL)	PPO	6.30 ± 0.16
	SAC	5.02 ± 0.31
	TD3	3.45 ± 0.10
	A2C	3.10 ± 0.14

In Setup 1 (RL + BT), the PPO algorithm required the longest training time, averaging 6.77 ± 0.57 hours, followed by SAC with 5.72 ± 0.26 hours. Despite its higher computational cost, PPO achieved the best performance, indicating that its longer training time is justified by superior results. SAC, while faster than PPO, did not reach the same performance level, suggesting that the slight reduction in training time comes with a trade-off in effectiveness. TD3 and A2C were the most time-efficient algorithms, with training times of 4.74 ± 0.24 and 4.10 ± 0.25 hours, respectively, but their performance

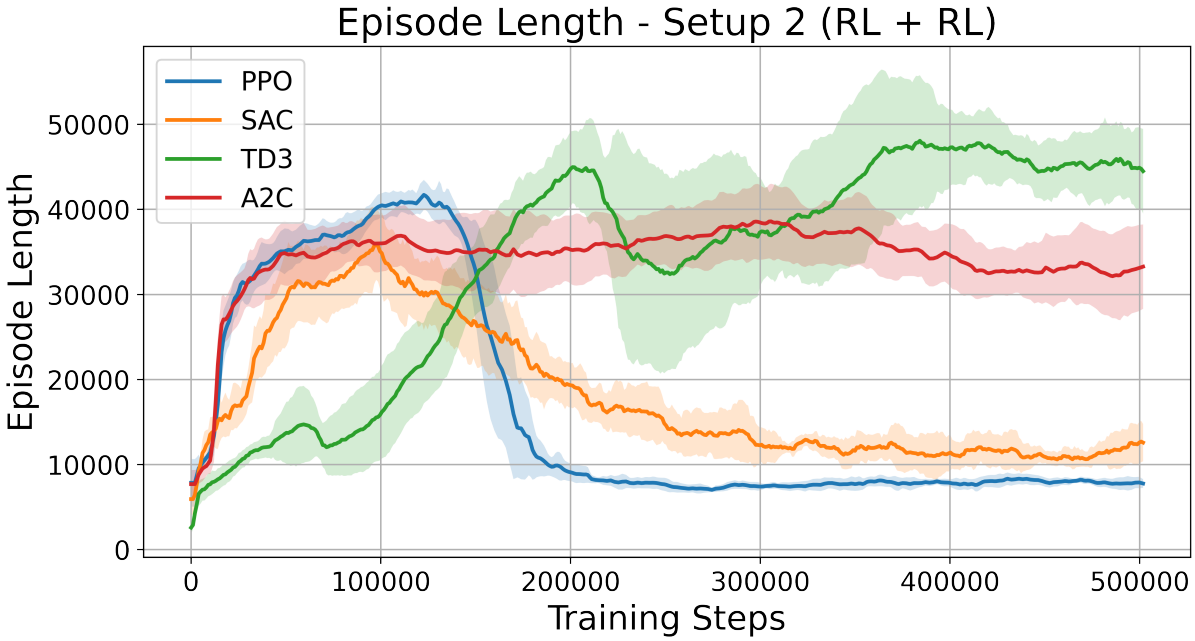


FIGURE 12.9 – Mean episode length for PPO, SAC, TD3, and A2C in Setup 2 (RL + RL).

was inferior to both PPO and SAC.

In Setup 2 (RL + RL), training times decreased across all algorithms. PPO's training time was reduced to 6.30 ± 0.16 hours, maintaining its performance advantage. SAC also saw an improvement in efficiency, with a training time of 5.02 ± 0.31 hours, though it remained less effective than PPO. TD3 and A2C continued to demonstrate the shortest training times, 3.45 ± 0.10 and 3.10 ± 0.14 hours, respectively, reinforcing their computational efficiency.

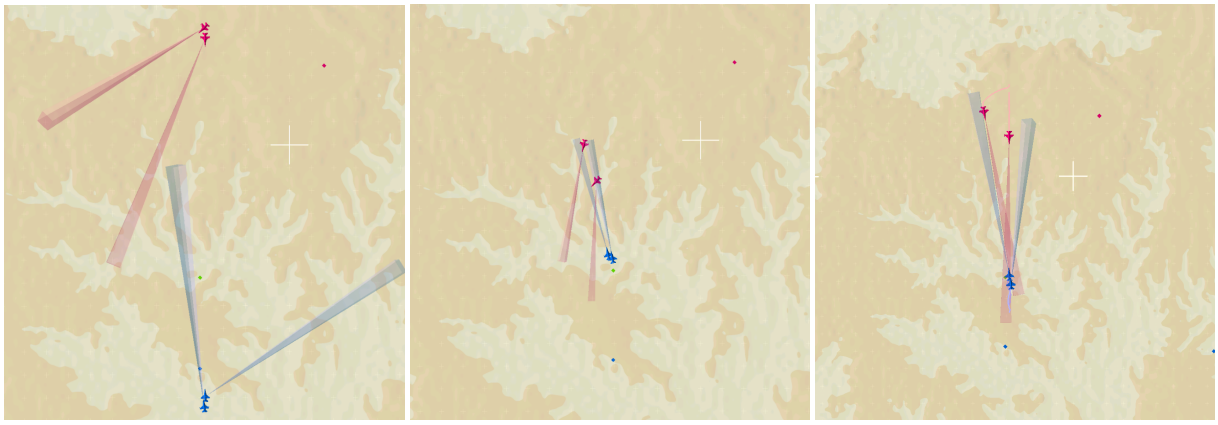
Despite Setup 2 being more complex, as it required the control of two agents through RL, the training times were consistently lower compared to Setup 1. This result may be attributed to the fact that training two agents simultaneously allowed the algorithms to collect twice as much experience per update, accelerating convergence. This suggests that the increased complexity in agent coordination did not necessarily lead to higher computational demands, possibly due to the shared learning process between the agents.

12.5.3 Agent Behavior Analysis

To better understand the learned behaviors, qualitative assessments were conducted by visualizing the agents' actions in different combat scenarios. The analysis focused on the PPO algorithm, which demonstrated the best performance among the evaluated methods. This evaluation was supported by SMEs, consisting of BVR fighter pilots from FAB, who provided insights into tactical maneuvers, positioning strategies, and decision patterns typically used in real-world BVR air combat. These insights guided the design

of the reward function and served as a reference for interpreting the agent's behavior.

Figure 12.10 presents a combined image made up of three subfigures: (a), (b), and (c), respectively illustrating the initial situation and agent behaviors in two distinct setups: Setup 1 (RL + BT) and Setup 2 (RL + RL). In these visualizations, blue and red aircraft represent the Blue and Red teams, respectively. The cones extending forward from each aircraft represent radar coverage. These cones indicate the area within which each aircraft is able to detect and track enemy targets, reflecting the sensor's range and field of view used for building situational awareness.



(a) (b) (c)

FIGURE 12.10 – Agent behavior in: (a) Initial engagement scenario, (b) Final engagement for Setup 1 (RL + BT), and (c) Final engagement for Setup 2 (RL + RL).

The initial situation at the start of the episode provides a common context for both setups. In Setup 1 (RL + BT), the PPO-controlled agent coordinates effectively with a BT-controlled wingman. Setup 2 (RL + RL) features two PPO-controlled agents exhibiting adaptive behaviors. Despite the different control schemes, both setups result in coordinated actions and successful target engagement, highlighting the effectiveness of the proposed reward-driven training. Agents in Setup 2 showed greater adaptability, dynamically adjusting their strategies in response to enemy maneuvers, while Setup 1 benefited from the predictable support provided by the BT-controlled wingman.

Additionally, to better illustrate the agent's behavior evolution during the engagement, Figure 12.11 presents a sequence of six frames sampled from a representative episode in Setup 2 (RL + RL). This sequence visually captures the agents' coordinated maneuvers and adaptive positioning as the engagement unfolds.

It is important to highlight that the use of an intermediate reward based on collective situational awareness allowed each agent to learn behaviors that led to situations of collective advantage. In other words, each agent made a combat decision that was favorable not only to the agent itself but also to the ally. These findings reflect common tactical principles in real 2v2 BVR air combat, such as mutual support, coordinated maneuvers,

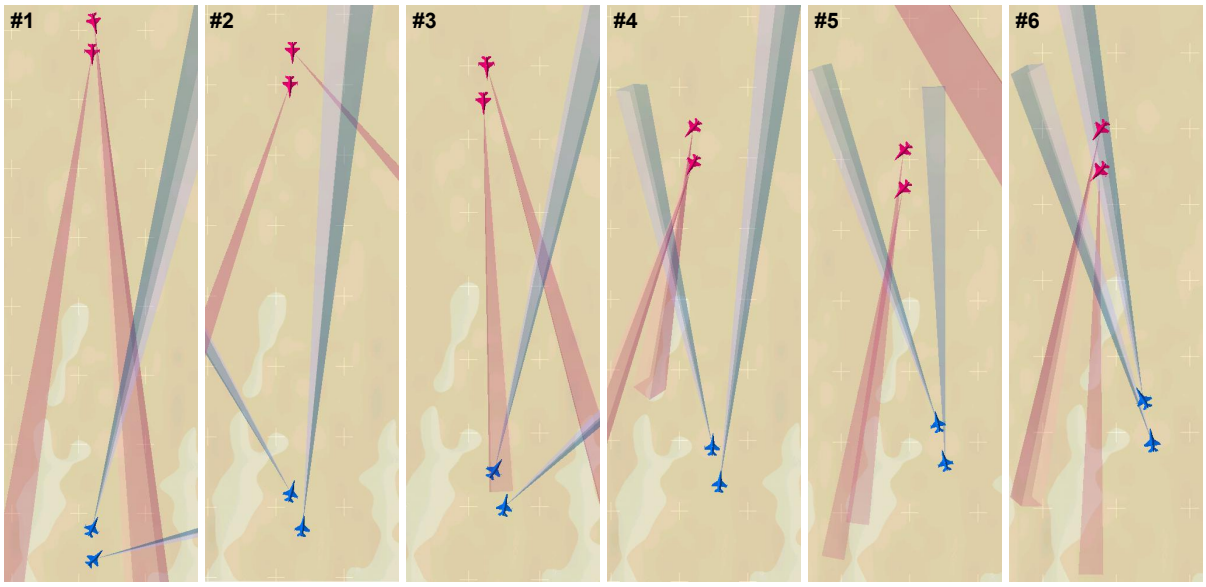


FIGURE 12.11 – Sequence of six frames from a typical engagement in Setup 2 (RL + RL). The images (#1 to #6) illustrate the evolution of the scenario, showing the adaptive maneuvers and coordination between the PPO-controlled agents as they react to enemy actions and work together to achieve positional advantage.

and adaptive targeting. According to feedback from SMEs, the ability of RL agents to create and maintain positional advantage while considering the ally's actions is similar to behaviors seen during real pilot training. This strengthens the case for using such agents in future decision support systems and human-autonomy teaming studies (DANTAS *et al.*, 2024).

12.5.4 Evaluation Summary

The evaluation highlights the superior performance of PPO across both setups, achieving the highest average rewards and shortest episode lengths, though at the cost of longer training times. SAC demonstrated stable learning but with slower convergence and lower rewards compared to PPO. TD3 and A2C exhibited inconsistent performance, with TD3 struggling in complex scenarios and A2C showing limited learning progress.

The qualitative behavior analysis was conducted specifically on PPO, given its superior performance, with the support of SMEs. This analysis confirmed that, in both training setups, the agents effectively engaged enemy aircraft by employing coordinated tactics and demonstrating clear signs of collective situational awareness throughout the engagements, including adaptive positioning, mutual coverage, and synchronized maneuvers.

12.6 Outcomes

This chapter addressed the use of DRL to model the engagement phase of BVR air combat, considering the collective situational awareness of both allies and opponents.

A key contribution was the development of the AsaGym computational library, which allows the development and training of DRL models for BVR air combat autonomous agents, i.e., fighter agents. Then, using AsaGym, the study analyzed the effectiveness of different DRL algorithms in a multi-agent air combat scenario and assessed their tactical capabilities based on operational knowledge from experts.

The results showed that PPO achieved the highest performance, demonstrating superior adaptability and decision-making, while SAC provided stable learning with slightly lower efficiency. TD3 and A2C faced more significant challenges in maintaining consistent performance. Training time analysis indicated that PPO required the most computational resources but delivered the best results, while TD3 and A2C trained faster but with limited effectiveness.

A qualitative assessment conducted by SMEs through visual analysis confirmed that agents with the engagement node controlled by a DRL model exhibited adaptive and cooperative tactics, such as synchronized offensive maneuvers and effective positioning strategies. In contrast, the setup, considering a leader with the engagement node controlled by a DRL model and a wingman with the engagement node controlled only by predetermined engagement rules, showed more rigid and predictable behaviors, reflecting the structured nature of BT-based decision-making.

Future work can address several limitations of this study. Introducing altitude variations would create more complex three-dimensional combat scenarios, enhancing the agents' maneuvering capabilities. Expanding to larger formations, such as n-versus-m, could evaluate the scalability of the approach and coordination among multiple agents. Decentralizing control by assigning independent policies to each agent, instead of a shared policy, may improve adaptability in dynamic situations. Additionally, relying solely on non-adaptive behavior tree-controlled opponents may cause overfitting; incorporating adaptive adversaries or leveraging self-play mechanisms would challenge the agents to develop more robust tactics. We also plan to explore adversarial reinforcement learning and self-play learning in larger formations such as 3v3 and 4v4. Involving human pilots in simulated missions could improve the agents' strategies by adding realistic feedback and enabling the study of real-time cooperation between humans and autonomous agents.

Furthermore, future research could explore replacing other behavior tree nodes beyond engagement, such as defensive and offensive maneuvers, allowing RL-based agents to handle a broader range of tactical decisions. Involving human pilots in simulated missions

could improve the agents' strategies by adding realistic feedback and enabling the study of real-time cooperation between humans and autonomous agents.

Another possible direction for future research is the evaluation and comparison of different reward functions for multi-agent air combat. While this study focused on one specific formulation designed to encourage collective situational awareness, a deeper analysis of alternative reward functions and their effects on coordination strategies may be explored in future work.

This thesis demonstrates different decision support systems, including those for weapon employment and engagement decisions, and shows how these can be integrated into training and simulation environments. One interesting suggestion for future research is to incorporate operational effectiveness metrics, such as probability of kill, directly into the reward function. Using such metrics would help ensure that the agents' behavior and training results are better matched to real mission needs and goals.

The findings highlight the potential of DRL to improve air combat simulations by allowing autonomous agents to learn and perform complex collective tactical maneuvers. The research showed that it is possible to use RL to train intelligent fighter agents that can adapt to dynamic scenarios and coordinate effectively in multi-agent settings. This work has the potential to develop autonomous agents that can operate with human pilots, supporting better teamwork between humans and machines in tactical missions.

12.7 Source Code

The source code used in this research is available in the AsaGym repository at `https://github.com/ASA-Simulation/asa-gym`. This repository provides an open-source version of the AsaGym environment, including its main modules, wrappers, and integration components, offering insights into the structure used for training and evaluating RL agents in air combat scenarios. The ASA framework itself, which runs the simulations, is not publicly available due to access restrictions. However, while researchers will not be able to run the environment, the released code can still serve as a reference for those working on RL applications in air combat.

13 Imitation and Generative Learning in Air Combat

The ability of autonomous agents to learn tasks by observing and mimicking the actions of humans or other agents is highly desirable in modern warfare. This research uses flight data from human fighter pilots to train models for autonomous air combat maneuvers, specifically pop-up attack maneuvers, which are widely employed in air-to-ground engagements. Additionally, generative learning techniques are employed to augment the dataset by creating additional data and enhancing the range and diversity of available human flight data. This chapter aims to develop systems replicating human decision-making and continuously improving through imitation learning, offering a promising method for advancing autonomous combat capabilities. As indicated in Figure 1.2, this chapter primarily contributes to the "Aerial Autonomous Agents" area within the proposed research framework.

The content of this chapter is derived from the following work:

> **DANTAS, J. P. A**. Autonomous Pop-Up Attack Maneuver Using Imitation Learning. In: Proceedings of the Winter Simulation Conference, PhD Colloquium. Proceedings [...]. Orlando, FL, USA: IEEE, 2024.
>
> **DANTAS, J. P. A.**; MAXIMO, M. R. O. A.; YONEYAMA, T. Autonomous Aircraft Tactical Pop-Up Attack Using Imitation and Generative Learning. IEEE Access, v. 13, p. 81204–81217, 2025.

13.1 Summary

This chapter presents a methodology for developing models that replicate the complex pop-up attack maneuver in air combat operations, using flight data from a Brazilian Air Force pilot in a 6-degree-of-freedom flight simulator. By applying imitation learning techniques and comparing three algorithms – Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) – the research trains models to

predict aircraft control inputs through sequences of state-action pairs. The performances of these models were evaluated in terms of Root Mean Squared Error (RMSE), coefficient of determination (R^2), training time, and inference time. To further enhance the training dataset with the aim of improving the robustness of the models, a Variational Autoencoder (VAE) was employed to generate synthetic data. These findings demonstrate the potential for deploying such models in fully autonomous aircraft, enhancing autonomous combat systems' reliability and operational effectiveness in real-world scenarios.

13.2 Introduction

The integration of autonomous systems into modern warfare is rapidly increasing, offering enhanced capabilities and operational efficiencies by taking over roles that traditionally required human involvement (SCHARRE, 2015). These systems are deployed across various military applications, from surveillance and reconnaissance to direct combat engagement, where they provide significant strategic benefits (RYAN; MITTAL, 2019). However, a key challenge in their development remains to enable them to execute complex tasks, such as air combat maneuvers, with the proficiency of human pilots (MCGREW, 2008).

A pop-up attack, for instance, is a maneuver where a fighter aircraft quickly ascends from a low altitude to engage a ground target, followed by a rapid descent to avoid counterattacks. This maneuver is a critical part of air combat operations and involves elements of surprise and precise execution under extreme conditions, making it particularly challenging to replicate autonomously (WANG *et al.*, 2009).

To address this challenge, imitation learning, specifically Behavior Cloning (BC), provides a promising approach for training autonomous systems. BC allows agents to learn by mimicking expert demonstrations, capturing human decision-making and execution skills from collected flight data (HUSSEIN *et al.*, 2017). This method is particularly effective for structured tasks where expert strategies are well-established, such as air combat maneuvers. In this work, BC is used to enable autonomous agents to replicate complex maneuvers like the pop-up attack with high fidelity (WANG *et al.*, 2022).

While reinforcement learning could also be applied to this problem, it introduces challenges such as the need for carefully designed reward functions and long training times for convergence (SUTTON; BARTO, 2018). Although reinforcement learning can discover novel strategies through trial and error, its learning process is often unstable and computationally expensive. Nonetheless, RL has already been applied in air combat scenarios, demonstrating its potential for maneuver optimization and tactical decision-making (GORTON *et al.*, 2024; DANTAS *et al.*, 2023). In contrast, BC follows a supervised learning approach,

making training more stable and data-efficient (ZARE *et al.*, 2023). Moreover, when expert demonstrations are available, imitation learning enables models to directly replicate successful strategies rather than searching for optimal actions independently. This advantage is particularly relevant in air combat, where decision-making is guided by established tactics and prior knowledge. Given these factors, BC was chosen as the primary method for training autonomous agents in this work, ensuring they can effectively learn from expert pilots and execute maneuvers with precision and reliability.

In addition to imitation learning, this work incorporates generative learning techniques (BOND-TAYLOR *et al.*, 2022) to generate synthetic flight data based on flight data from human pilots collected from a simulation environment. The synthetic data aims to improve the model's performance, enabling better generalization and enhancing the ability of autonomous systems to perform complex air combat maneuvers. This approach helps mitigate issues related to data limitations, which are common when working with complex military operations where collecting large amounts of real data may be impractical or costly (PARK *et al.*, 2021).

The main contribution of this chapter is the development of models for autonomous pop-up attack maneuvers using imitation learning, specifically BC. By training these models on flight data collected from human pilots in a simulation environment, the study aims to replicate human decision-making in air combat operations. Additionally, synthetic data generation using generative learning is introduced to expand the dataset, enhancing the robustness of the models. This work enhances our understanding of autonomous systems' ability to perform reliably in dynamic and unpredictable combat scenarios, emulating the proficiency of human pilots.

The remainder of this work begins with Section 13.3, which provides an operational background on the pop-up attack maneuver, a key technique in air-to-ground combat. Following this, in Section 13.4, we present an overview of related work, surveying key contributions in the fields of imitation and generative learning applied to autonomous systems, with a particular focus on the air domain. In Section 13.5, the study details the data collection process, emphasizing the role of a flight simulator model to capture the maneuver's complexities. This section also describes the development of multiple BC models using different techniques and explores generative learning methods to synthesize additional flight data. In Section 13.6, we present the results and provide a performance analysis of the models across various configurations. Finally, Section 13.7 concludes the work by discussing the research outcomes and suggesting future directions.

13.3 Operational Background

The pop-up attack maneuver is a key technique in air-to-ground combat, designed to enable a fighter aircraft to approach and engage a target while minimizing its exposure to enemy defenses (UNITED STATES AIR FORCE, 1996). This maneuver is initiated from a preplanned Pop-Up Point (PUP), which is strategically selected to optimize the aircraft's approach path and timing for the attack sequence (WANG *et al.*, 2009). As depicted in Figure 13.1, the offset pop-up maneuver typically involves an approach angle ranging from 15° to 90° relative to the final attack heading. This angular approach allows the pilot to visually acquire the target early and maintain visual contact until the weapon is released.

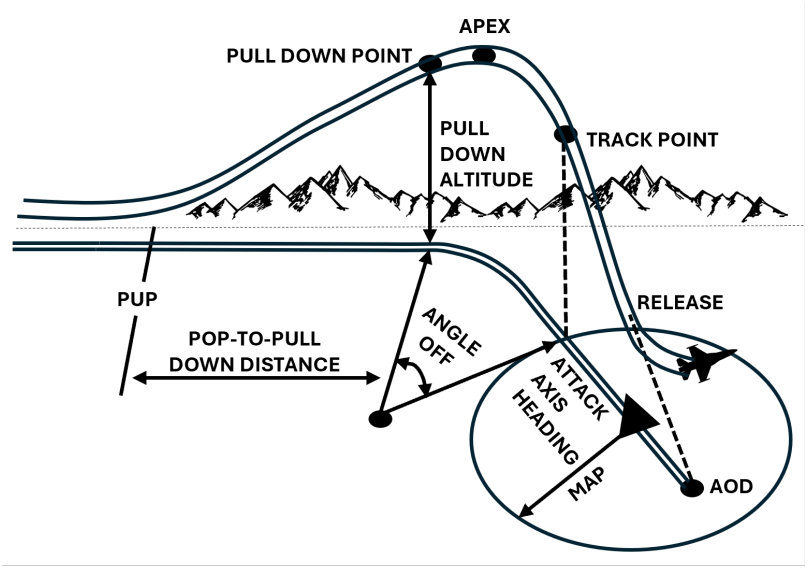


FIGURE 13.1 – Flight profile for offset pop-up delivery.

The maneuver begins as the aircraft reaches the PUP, with the pilot initiating a climb at a minimum speed of 450 knots calibrated speed. At this point, the pilot selects the appropriate power setting and executes a $3-4\,G$ wings-level pull-up to achieve the planned climb angle. Simultaneously, a chaff/flare countermeasure program is activated, which releases metallic strips (chaff) and infrared flares to confuse radar-guided and heat-seeking missiles, respectively, protecting the aircraft from potential surface-to-air threats. During the climb, the pilot must precisely maintain the planned climb angle while carefully monitoring the altitude gain (UNITED STATES AIR FORCE, 1996; FOO *et al.*, 2009).

As the aircraft approaches the predetermined pull-down altitude, the pilot executes a smooth roll towards the target, followed by a $3-5\,G$ pull-down maneuver to intercept the planned dive angle. During this phase, the pilot passes through the track point, where the aircraft's trajectory is adjusted to ensure alignment with the target (FOO *et al.*, 2009). Achieving this dive angle while aligning with the aim-off point (AOD), which is the ground distance from the target where the nose of the aircraft is pointed during tracking,

is essential for maintaining the preplanned delivery parameters. Minor deviations in the attack heading are generally acceptable and can be corrected during the final phase of the maneuver (UNITED STATES AIR FORCE, 1996).

Throughout the maneuver, the pilot must make real-time adjustments for any deviations in the pop-up point or unanticipated wind conditions encountered during the climb. The apex, the highest altitude reached in the pop-up delivery profile, is typically reached approximately halfway through the pull-down maneuver, providing the necessary altitude and positioning to execute the attack effectively. Additionally, the distance from the final maneuver point to the target (MAP) must be carefully managed as it combines bomb range and horizontal distance covered while tracking (UNITED STATES AIR FORCE, 1996).

Given the precision required at each stage, the pop-up attack maneuver is a highly complex and demanding technique. Successfully executing this maneuver presents significant challenges, especially for autonomous systems, which must replicate the decision-making and real-time adjustments that human pilots perform under dynamic conditions.

13.4 Related Work

Research on air combat operations has long recognized the complexity and dynamic nature of these military engagements. These operations require rapid decision-making, precise execution, and the ability to adapt to evolving threats in real-time (DANTAS *et al.*, 2021a). Historically, human pilots have been the base of air combat success, leveraging their skills and decision-making abilities to outmaneuver adversaries. However, with advancements in autonomous systems, there is a growing focus on replicating and enhancing these capabilities through automation (DOU *et al.*, 2023; WANG *et al.*, 2023). Integrating autonomous technologies into air combat offers potential benefits such as increased mission effectiveness, reduced cognitive load on operators, and improved survivability, though it also presents challenges in developing systems that can emulate human decision-making in critical environments (GUNSCH *et al.*, 1993).

In this context, recent research has increasingly focused on developing autonomous systems capable of executing complex air combat maneuvers. These systems are designed to handle a broad range of missions, including both close-range air-to-air engagements (dogfighting) and beyond-visual-range (BVR) combat, where detecting, tracking, and engaging adversaries occur at long distances (DANTAS *et al.*, 2022). Autonomous systems capable of excelling in air-to-air (DANTAS *et al.*, 2021b; DANTAS *et al.*, 2025) and ground-to-air combat (DANTAS *et al.*, 2023b), as well as in air-to-ground tactical operations, as presented in this study, represent a significant advancement in modern military aviation (MILLER *et al.*, 2007).

Imitation learning has emerged as a highly effective method for training autonomous systems by enabling them to mimic the decision-making processes of experienced pilots. For instance, deep feature representation has been used to map flight observations to continuous control actions in autonomous helicopters, showcasing the feasibility of transferring expert-level skills to autonomous systems (CHEN *et al.*, 2021). Similarly, intelligent autopilot systems that learn piloting skills through imitation have successfully replicated both low- and high-level flight skills, including complex maneuvers (BAOMAR; BENTLEY, 2016). Further work has explored the cloning of fighter pilot strategies through imitation learning, allowing autonomous systems to replicate intricate combat tactics, proving effective in replicating decision-making patterns under combat conditions (SANDSTRÖM *et al.*, 2022). Additionally, the DAgger algorithm has been applied to train neural network-based autopilots in unmanned aerial systems (UAS), showing significant improvements in generalization across diverse flight maneuvers and proving effective in maintaining flight stability and adaptability during complex aerial tasks (SHUKLA *et al.*, 2020).

Recent research has extended imitation learning to air combat scenarios. For instance, exploratory studies have highlighted its potential for modeling fighter pilot behavior, enabling autonomous agents to replicate combat tactics and decision-making strategies under dynamic and uncertain environments (GORTON *et al.*, 2023). This includes the development of methods to capture the complex behaviors of pilots and translate them into effective autonomous system strategies. Complementary work has focused on data-driven behavioral modeling for military applications, emphasizing the integration of imitation learning techniques to replicate the nuanced decision-making processes of expert operators in defense scenarios (SCHADD *et al.*, 2022). These studies demonstrate the critical role of imitation learning in enhancing the operational capabilities of autonomous systems, both in civilian and military aviation contexts, by improving their ability to adapt to complex and high-stakes situations.

Generative learning has been explored as a complementary approach to imitation learning, particularly for data augmentation in training models. For example, generative models have been employed to create motion control policies, enabling autonomous systems to generalize to new scenarios with synthesized data (YAO *et al.*, 2022). Additionally, Generative Adversarial Networks (GANs) have been applied to iteratively improve data efficiency in reinforcement learning, enhancing model performance (LIU *et al.*, 2019). Another approach uses a GAN-based training model to generate high-quality synthetic data for lightweight convolutional neural networks, addressing the shortage of training data and improving classifier accuracy (RATHER; KUMAR, 2023). Generative models have also been utilized to improve the robustness of object detection systems in low-visibility environments, such as in (PREMAKUMARA *et al.*, 2023), where synthetic data augmentation effectively counters natural perturbations like low light and blur, improving model

robustness in real-world scenarios.

Synthetic data has also been explored in military decision support systems, focusing on its fidelity and applicability in real-world scenarios. For instance, (WEYLAND *et al.*, 2024) evaluates the effectiveness of synthetic data in replicating operational conditions and enhancing decision-making processes in complex environments. This work highlights the importance of assessing synthetic data quality to ensure its reliability in augmenting training datasets and supporting robust system performance in military applications. Also, (DANTAS *et al.*, 2022b) introduced resampling techniques to generate synthetic data to address imbalanced data, which improved model reliability for supervised learning in autonomous aircraft systems in the context of aerial combat.

Previous work has examined the pop-up attack maneuver as an optimization-based approach to tactical mission planning, emphasizing weapon delivery precision, ballistic trajectory control, and detection avoidance, thus serving as a critical foundation for applications in both human-piloted and autonomous aircraft systems (WANG *et al.*, 2009). Additionally, intent inference models have been applied in air defense contexts, using flight profile analysis to predict potential weapon delivery points in pop-up scenarios, thereby enabling proactive threat assessment and enhancing situational awareness (FOO *et al.*, 2009).

13.5 Methodology

This section outlines the methodological framework employed in this work, focusing on developing both imitation and generative learning models. The following subsections detail the characteristics of the flight data and the approach used to develop and evaluate multiple imitation learning models. Additionally, this section introduces a generative learning technique to produce synthetic data, aiming to expand the dataset and enhance the robustness of the models.

13.5.1 Flight Data

The dataset for this research comprises 30 flight recordings of pop-up attack maneuvers executed by a Brazilian Air Force (FAB) fighter pilot. The flight data was collected using AEROGRAF (PETERSEN *et al.*, 2008), a 6DOF (Six Degrees of Freedom) flight simulator model based on the F-16 Fighting Falcon aircraft, which was developed by the FAB. AEROGRAF served as the predecessor to the Aerospace Simulation Environment (ASA), also developed by the FAB (DANTAS *et al.*, 2022a; DANTAS *et al.*, 2023a).

All flights in the dataset start from exactly the same initial point and heading, located

5.9 nautical miles from the target and with an altitude difference of about 146 meters, thus providing a standardized initial condition for each maneuver. These distances, altitudes, and headings were determined in collaboration with subject matter experts to represent a common operational scenario for this maneuver.

To ensure uniformity, each flight recording was trimmed to match the shortest sequence length across all samples. This process standardizes the data, enabling consistent inputs for model training and reducing variability in training data, which supports more reliable model performance. Figure 13.2 illustrates the flight patterns within this dataset, showing the trajectories for each of the 30 flights utilized in this study.

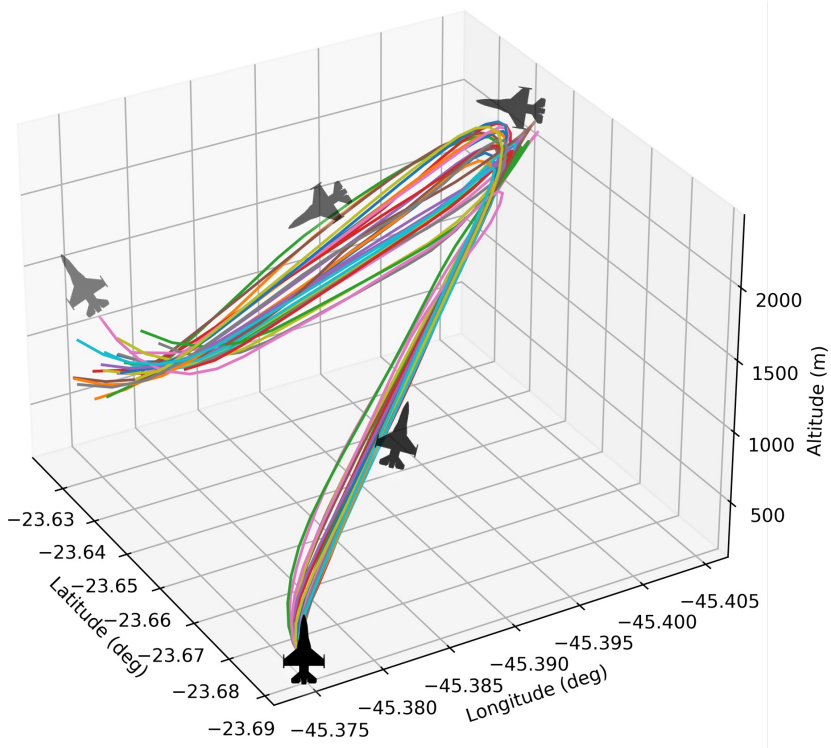


FIGURE 13.2 – Adjusted flight data for the 30 flights of the pop-up attack maneuver executed by the human pilot.

While this dataset provides valuable insights for training and evaluating autonomous agents, transitioning from simulated environments to real-world deployment presents several challenges. A key issue is the potential dynamic mismatch between the simulator and a real aircraft. Unless a high-fidelity, certified simulator is used, discrepancies in system dynamics, response times, and environmental factors could affect the transferability of trained models to real-world scenarios. Given these challenges, the deployment of autonomous agents should begin with extensive validation in simulation, including tests with variations in model parameters to assess robustness before considering real-world implementation. In this context, the flight data used in this study serves as an initial exploration for developing autonomous agents capable of performing air combat maneuvers, providing a first study for future advancements.

13.5.2 Imitation Learning Models

In this subsection, we will develop imitation learning models designed to replicate the actions of a human pilot during a pop-up attack maneuver. The models are trained to predict control inputs based on the aircraft's state, effectively learning to perform the maneuver autonomously. We will detail the process of constructing the imitation learning models, beginning with the preparation of state and action vectors from the recorded flight data, followed by presenting the architectures of the different models. Finally, we will clarify the model training process, how flight trajectory predictions are made, and the methods used for evaluation and analysis.

13.5.2.1 State and Action Vectors

To develop the imitation learning models, the flight data was segmented into sequences of state-action pairs. The state vectors, defined relative to the aircraft's body frame, included key variables: altitude, pitch, roll, and yaw angles, along with radial angle, distance to the target, and altitude difference between the aircraft and the target. The action vectors consisted of the control inputs commanded by the pilot: pitch, roll, and throttle. To ensure unbiased training, all data was normalized using the mean and standard deviation computed across the entire dataset.

All imitation learning models were trained in two configurations: a baseline version, which excluded derived variables such as linear velocities, angular velocities, and accelerations, and a full version, which included all available state variables. This approach was designed to emphasize the importance of these derived variables and to demonstrate their impact on model performance by comparing simpler models that lack temporal dependency handling with those capable of capturing it. The variables used in the state and action vectors are detailed in Table 13.1, which outlines the units and descriptions of each variable.

13.5.2.2 Multi-Layer Perceptron

The Multi-Layer Perceptron (MLP) model, a type of Feedforward Neural Network, was designed to efficiently learn the relationship between flight states and actions by utilizing fully connected layers that apply non-linear transformations to capture complex patterns between state variables and action predictions (GOODFELLOW *et al.*, 2016). We propose a model based on this MLP architecture, implemented using TensorFlow (ABADI *et al.*, 2016), whose final structure is illustrated in Figure 13.3, depicting the layers of the MLP-based imitation learning model. The architecture includes the following layers:

TABLE 13.1 – State and action variables used in the imitation learning model.

Variable	Units	Description	Type
ALT (m)	Meters	Altitude in meters	State
Phi (deg)	Degrees	Pitch angle (positive for nose-up)	State
Theta (deg)	Degrees	Roll angle (positive for left roll)	State
Psi (deg)	Degrees	Yaw angle	State
Vx (m/s)	Meters/second	Velocity in the pitch direction	State
Vy (m/s)	Meters/second	Velocity in the roll direction	State
Vz (m/s)	Meters/second	Velocity in the yaw direction	State
P (deg/s)	Degrees/second	Pitch angular velocity	State
Q (deg/s)	Degrees/second	Roll angular velocity	State
R (deg/s)	Degrees/second	Yaw angular velocity	State
Nx (m/s^2)	Meters/second^2	Lateral acceleration	State
Ny (m/s^2)	Meters/second^2	Longitudinal acceleration	State
Nz (m/s^2)	Meters/second^2	Vertical acceleration	State
Radial (deg)	Degrees	Angular position of the aircraft relative to the target	State
Distance (m)	Meters	Horizontal distance between the aircraft and the target (ground range)	State
DeltaAlt:Anv-Tgt (m)	Meters	Altitude difference between aircraft and target	State
JX	–	Positive for nose-up pitch	Action
JY	–	Positive for left roll	Action
Throttle	–	Throttle position	Action

- **Dense Layer 1:** A fully connected layer with 128 units, ReLU activation, serving as the first internal layer to capture high-dimensional relationships in the input data.
- **Dense Layer 2:** A fully connected layer with 64 units, ReLU activation, providing further non-linear transformations for improved model accuracy.
- **Dense Layer 3:** A fully connected layer with 32 units, ReLU activation, refining the learned representations.
- **Output Layer:** The final output layer has 3 units corresponding to the predicted actions (JX, JY, and Throttle), without any activation function, to output direct action values.

13.5.2.3 Long Short-Term Memory Network

The Long Short-Term Memory (LSTM) network, an advanced type of Recurrent Neural Network (RNN), was designed to capture complex temporal dependencies in sequential data. LSTMs are particularly effective in learning long-term dependencies, making them ideal for time-series data where each output relies on prior inputs (HOCHREITER; SCHMIDHUBER, 1997). We propose a model based on an LSTM architecture, also implemented using TensorFlow, with its final structure illustrated in Figure 13.4 and consisting of the following layers:

- **LSTM Layer:** The first layer of the network is an LSTM (Long Short-Term Memory) layer with 128 units, which captures temporal dependencies across the entire

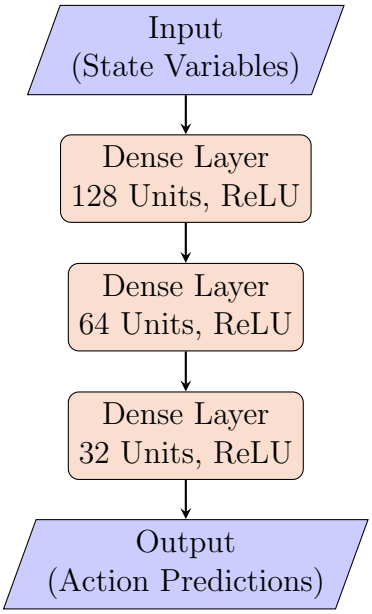


FIGURE 13.3 – Architecture of the MLP-based imitation learning model.

sequence. It processes each step of the input data while maintaining a sequential nature, allowing the model to learn from past steps and make predictions based on long-term dependencies in the data.

- **TimeDistributed Dense Layer 1:** The second layer is a TimeDistributed Dense layer with 64 units and ReLU activation. The TimeDistributed wrapper applies the Dense layer to each time step independently, allowing the network to learn complex relationships between the input state and the predicted action at each time step in the sequence.

- **TimeDistributed Dense Layer 2:** Another TimeDistributed Dense layer with 32 units and ReLU activation further refines the predictions. Like the previous layer, it is applied independently to each time step, adding non-linear transformations that enhance the network's ability to handle complex sequential patterns.

- **Output Layer:** The final layer is a TimeDistributed Dense layer with 3 units (corresponding to the actions: JX, JY, and Throttle) and no activation function. This layer produces the directly predicted values for each action at each time step, allowing the network to output continuous values, which is typical for regression tasks where real-valued outputs are predicted.

13.5.2.4 Gated Recurrent Unit

The Gated Recurrent Unit (GRU) model provides a simplified alternative within the RNN family. Compared to Long Short-Term Memory (LSTM) networks, GRUs utilize

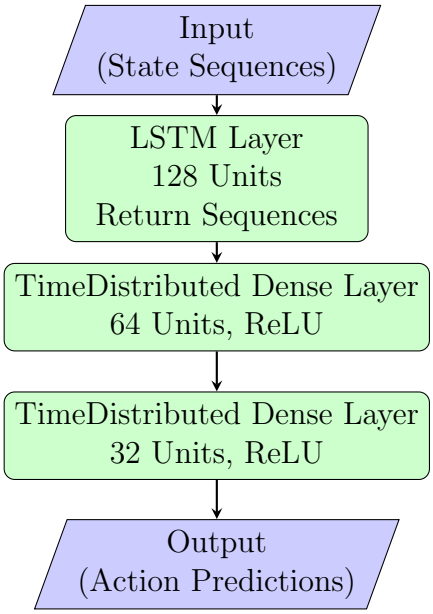


FIGURE 13.4 – Architecture of the LSTM-based imitation learning model.

a simpler structure with fewer parameters by merging the forget and input gates into a single update gate and removing the cell state (CHO *et al.*, 2014). This efficient design allows GRUs to achieve performance similar to LSTMs while lowering computational costs, making them effective for tasks like trajectory prediction. Despite its simplicity, the GRU can maintain essential temporal relationships in the data, making it a practical choice for this work. We propose a model based on this GRU architecture, with the final design illustrated in Figure 13.5, which depicts the layers of the GRU-based imitation learning model. The architecture includes the following layers:

- **GRU Layer:** The first layer consists of a GRU with 128 units, configured to return sequences. Like the LSTM, the GRU captures temporal dependencies across the entire sequence, processing each time step while preserving the sequence's sequential nature. The GRU layer is similar to the LSTM but is more computationally efficient, as it uses fewer parameters while still learning long-term dependencies in the data.

- **TimeDistributed Dense Layer 1:** A TimeDistributed Dense layer with 64 units and ReLU activation is applied to each time step independently. This allows the model to learn complex relationships between the input states and the corresponding action predictions at each time step in the sequence.

- **TimeDistributed Dense Layer 2:** A second TimeDistributed Dense layer with 32 units and ReLU activation further refines the predictions by introducing additional non-linear transformations. Just like the first TimeDistributed layer, this layer is applied independently to each time step, enhancing the model's capacity to capture intricate patterns in sequential data.

- **Output Layer:** The final TimeDistributed Dense layer has 3 units (corresponding to the actions: JX, JY, and Throttle), producing the directly predicted values for each action at each time step in the sequence. The absence of an activation function ensures that the network can output continuous values, typical of regression tasks.

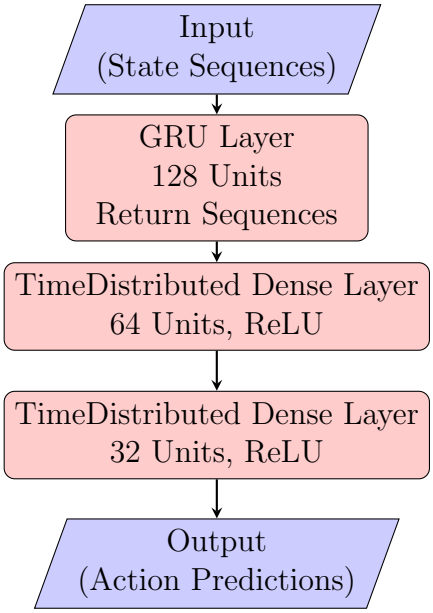


FIGURE 13.5 – Architecture of the GRU-based imitation learning model.

13.5.2.5 Model Training

The training process involved 5-fold cross-validation to ensure robust evaluation of the models. The dataset was split, with 30% reserved for testing and the remaining 70% used for training and validation. During cross-validation, the models were trained and evaluated across each fold, using the Adam optimizer (KINGMA; BA, 2014) with a learning rate of 10^{-5} and the Mean Squared Error (MSE) loss function, which is suitable for continuous regression problems. A batch size of 32 was employed to maintain computational efficiency while balancing memory usage and training stability.

Within each fold, early stopping was implemented with a patience of 20 epochs, monitoring the validation loss to prevent overfitting. This allowed the training process to halt if no improvement was detected in validation loss over 20 consecutive epochs, thus reducing the risk of overfitting to a particular fold. Hyperparameter tuning was carried out using random search (BERGSTRA; BENGIO, 2012), covering a range of configurations for activation functions, the number of units per layer, the depth of the model (number of layers), learning rate, and batch size. This randomized search process allowed for the exploration of diverse model architectures, optimizing the choice of hyperparameters to enhance the models' performance while maintaining generalization capabilities. In this

study, only the best-performing architectures for each model, as determined through tuning, are presented, focusing on configurations that maximize performance and stability.

Once cross-validation was complete, the final models were trained on the entire dataset, leveraging all available data to enhance model performance. In this phase, early stopping was not employed since no dedicated validation group was available. Instead, the number of epochs for this training was determined based on the average number of epochs achieved during cross-validation, providing an empirically informed stopping criterion to ensure that the models were sufficiently trained without excessive epochs. By incorporating all data, the models gained the advantage of a more comprehensive training set, which could contribute to improved generalization on unseen samples.

To address the high variability observed in the results with the addition of synthetic data, the final model was trained five times using different seeds. This approach aimed to enhance the robustness of the results by averaging the performance across multiple runs, thereby mitigating the effects of random initialization and stochastic processes during training. This methodology ensured more stable and reliable results, particularly given the variability introduced by the synthetic data.

The entire training process was conducted on a system with 20 cores of the Intel Xeon Gold 6230R CPU, running at 2.10 GHz, and 40 GB of RAM, providing sufficient computational resources to handle the training workload efficiently.

13.5.2.6 Prediction of Flight Trajectories

Different methods were employed to predict flight trajectories based on the model type. For the MLP model, a single-step prediction approach was used, where each state vector was treated independently. The MLP model predicted the corresponding action for each time step without considering a sequence of states. This was implemented using a function that normalizes the input state sequence and uses the trained model to predict actions across the entire sequence. The actual and predicted actions were then collected for comparison, with the results plotted to show the mean and standard deviation of the trajectories.

In contrast, the LSTM and GRU models, designed to capture temporal dependencies, used a sliding window approach (Figure 13.6). In this method, the input data was segmented into overlapping state sequences of fixed length, allowing the models to learn patterns across time effectively. The sequence length was set to 5, chosen based on experimentation with values of 3, 5, 10, 15, and 20. This choice proved effective, as it balances capturing temporal dependencies efficiently and aligns well with the maneuver length of 39 frames. Each window contained a sequence of consecutive time steps, with a fixed overlap (stride = 1), ensuring that the model maintained continuity across predictions (YU

et al., 2014; HOTA *et al.*, 2017). These sequences were provided to the model to generate action predictions, where each window produced a prediction for the last time step, and these predictions were accumulated sequentially to reconstruct the entire trajectory. Any remaining time steps not covered by the sliding windows were filled by repeating the first predicted action, ensuring consistency with the original sequence length. The function for the LSTM and GRU models included normalization of each window, model prediction, and denormalization of the output actions. The complete process for this sliding window approach is outlined in Algorithm 2.

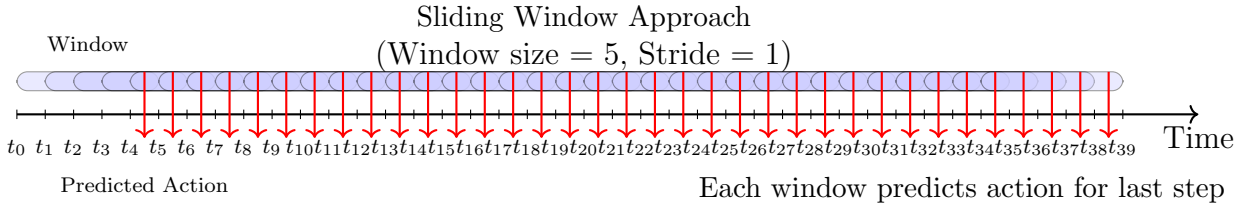


FIGURE 13.6 – Sliding window approach with overlapping sequences to predict flight trajectory across 39 timesteps. Only the first window and predicted action are labeled to illustrate the pattern followed by all subsequent windows.

Algorithm 2 Sliding window approach for predicting full flight trajectories (LSTM and GRU)

Input: Full sequence of state vectors
Output: Predicted full trajectory
Initialize an empty list for storing predicted actions
Divide the full sequence into overlapping windows of fixed length L (where $L = 5$) with a stride of 1
for each window **do**
    Reshape the window to match the input shape expected by the model
    Feed the window into the model to predict the action for the last time step
    Append the predicted action to the list of predicted actions
end for
if any remaining time steps were not covered by the windows **then**
    Fill the first uncovered time steps with the first predicted action, repeating it
end if
Return the complete predicted trajectory

13.5.2.7 Evaluation and Analysis

The predicted trajectories were compared with the actual recorded maneuvers to assess the model's performance in replicating realistic flight actions. The evaluation metrics, including Root Mean Squared Error (RMSE) and the Coefficient of Determination (R^2), were calculated on the denormalized output values to ensure that predictive accuracy reflects the original scales of the actions. Additionally, a comparative analysis of the mean and standard deviation of the predicted actions was conducted against the actual actions, providing insights into the precision and consistency of the model's predictions. This analysis highlighted any areas where variability diverged from the real maneuvers, allowing for a deeper understanding of the model's stability across different time steps and scenarios and identifying specific areas for further refinement.

13.5.3 Generative Learning Model

This subsection details the generative learning model employed in this work. A Variational Autoencoder (VAE), implemented using TensorFlow, is used to generate synthetic flight data that replicates the dynamics of pop-up attack maneuvers. The VAE architecture allows the model to learn a low-dimensional latent representation of the high-dimensional flight data, which is then sampled to generate synthetic sequences. These sequences enhance the training dataset and improve model robustness.

13.5.3.1 Data Preprocessing

To prepare the flight data for VAE training, all available recordings of the pop-up attack maneuver were processed and normalized. The data was loaded from text files and stripped of irrelevant columns. Next, the data was normalized by subtracting the mean and dividing by the standard deviation computed across all flight records, ensuring a consistent scale across variables.

13.5.3.2 Architecture

The VAE model consists of three main components: the encoder, the latent space sampling, and the decoder. The encoder compresses the input flight data into a latent representation, the latent space sampling introduces variability into the model, and the decoder reconstructs the data from the latent space.

- **Encoder:** The encoder takes an input sequence of specified length and dimensionality, passing it through a bidirectional LSTM layer with 256 units and L2 regularization to prevent overfitting (NG, 2004). This is followed by a Dense layer with 128 units and ReLU activation. A dropout layer with a rate of 0.4 is applied after the Dense layer to further enhance regularization (SRIVASTAVA *et al.*, 2014). The encoder outputs two vectors representing the mean and variance of the latent space distribution.

- **Latent Space Sampling:** To introduce variability in the synthetic data, a sampling function generates latent vectors by combining the mean and variance vectors with Gaussian noise. The latent dimension was set to 50, enabling the VAE to capture the underlying structure of the flight data while maintaining computational efficiency.

- **Decoder:** The decoder reconstructs the input data from the sampled latent vector. Since the latent vector represents a single compressed representation of the input sequence, it is first passed through a `RepeatVector` layer to match the sequence length,

enabling the decoder to process each timestep independently. The `RepeatVector` effectively "repeats" the latent vector across the number of timesteps, preparing it for the next layers. Following this, a Dense layer with 128 units and ReLU activation is applied, with a dropout layer (rate 0.4) to provide additional regularization. A TimeDistributed LSTM layer with 256 units processes the repeated latent vectors over time, followed by an output layer with linear activation that produces the final reconstructed sequence.

13.5.3.3 Loss Function

The VAE's loss function consists of two terms:

- **Reconstruction Loss:** This term, based on the mean squared error, measures the difference between the original input sequence and its reconstruction, summed over all timesteps. It ensures that the generated sequences closely resemble the original data.
- **KL Divergence Loss:** The Kullback-Leibler (KL) divergence term regularizes the latent space, encouraging it to follow a standard Gaussian distribution (KINGMA; WELLING, 2013). This term is weighted by a factor β to balance between generating realistic data and maintaining a smooth latent space.

13.5.3.4 Training Procedure

The VAE was trained using an Adam optimizer with a learning rate scheduler that decays exponentially over training epochs, starting with an initial learning rate of 1×10^{-3}, decay steps of $10,000$, and a decay rate of 0.9. This approach facilitates efficient convergence by reducing the learning rate as training progresses. Early stopping with the patience of 20 epochs was employed to stop training once the validation loss plateaued, ensuring efficient learning without overfitting. The training was conducted with a large maximum number of epochs (1,000,000) to ensure convergence, though early stopping typically halted training much earlier. The training was conducted with a batch size of 32 and a validation split of 20%.

13.5.3.5 Synthetic Data Generation

After training, the decoder component of the VAE was extracted and used to generate synthetic flight sequences by sampling from the latent space. Synthetic samples were generated in quantities of 7, 15, 30, 45, 60, and 150 to assess the impact of varying amounts of synthetic data on model performance, with generation reaching up to 5 times

the size of the original dataset (30). These synthetic sequences were combined with real flight data to enrich the training dataset, ultimately enhancing the model's robustness in learning from a diverse set of trajectories.

13.5.3.6 Fine-Tuning of Synthetic Data

To enhance the realism of the synthetic flight data, a fine-tuning process was applied to the generated sequences. This process involved using a centered moving average to smooth the generated trajectories, reducing noise while preserving essential maneuver characteristics. Through experimentation, a moving average window of 10 frames was found to provide a suitable balance, ensuring that the synthetic data aligns more closely with the patterns observed in real flight data.

Additionally, subject matter experts (SMEs) from FAB reviewed and qualitatively validated the generated data through visual inspection, ensuring consistency with how a real pilot would execute the maneuver. Their expertise helped confirm that the synthetic sequences followed expected flight dynamics, reinforcing the reliability of the augmented dataset.

13.5.4 Evaluation of Synthetic Data Impact on Model Performance

To evaluate the impact of synthetic data on model performance, synthetic samples generated by the Variational Autoencoder (VAE) were added to the training and validation datasets. In each configuration, we maintained a consistent test set across all experiments to ensure that any observed performance differences were attributable exclusively to the additional synthetic data.

For each configuration (baseline and full), we trained models (MLP, LSTM, and GRU) on these augmented training sets and evaluated them using metrics such as R^2, RMSE, training time, and inference time. It is important to note that the reported inference time corresponds to predicting the entire test dataset, which remained the same across all models for consistency. This process enabled a thorough analysis of how incorporating synthetic data affects model training dynamics and performance potential, particularly in the context of replicating complex maneuvers. By gradually increasing the synthetic data volume, we aimed to assess its role in improving model robustness and generalization.

13.6 Results and Discussion

In this section, we present the outcomes of our experiments and analyze the impact of the proposed methods. In the first subsection, 13.6.1, we evaluate the performance of imitation learning techniques in replicating specific maneuvers based on actual flight data. The second subsection, 13.6.2, examines the role of synthetic data in enhancing model performance, particularly in scenarios with limited real-world data. Together, these analyses provide insights into the effectiveness of combining imitation learning with synthetic data generation for robust trajectory prediction and maneuver execution.

13.6.1 Imitation Learning for Maneuver Execution

The performance of the imitation learning models was evaluated on the test set, with models trained in both baseline (without derived variables) and full (with all state variables) configurations. Table 13.2 summarizes the performance metrics for each model configuration, presenting the final R^2 and RMSE values, as well as training and inference times.

TABLE 13.2 – Performance metrics for imitation learning models (baseline and full configurations). Values are presented as mean ± standard deviation, calculated over multiple executions with five different seeds. The best values in each column are highlighted in blue.

Model	Configuration	R^2	RMSE	Training Time (s)	Inference Time (ms)
MLP	Baseline	0.323 ± 0.009	8.850 ± 0.015	76.11 ± 1.12	267.52 ± 19.19
	Full	0.726 ± 0.070	1.977 ± 0.054	140.84 ± 0.40	$\mathbf{266.92 \pm 12.25}$
LSTM	Baseline	0.672 ± 0.003	8.708 ± 0.059	54.08 ± 17.33	1432.22 ± 122.63
	Full	0.970 ± 0.001	1.743 ± 0.088	74.12 ± 3.66	1283.86 ± 119.91
GRU	Baseline	0.670 ± 0.003	8.698 ± 0.055	$\mathbf{35.85 \pm 4.52}$	1716.36 ± 304.40
	Full	$\mathbf{0.971 \pm 0.002}$	$\mathbf{1.738 \pm 0.064}$	74.63 ± 2.47	1206.61 ± 65.94

The results indicate that including all state variables (full configuration) consistently improves model performance across all architectures, as reflected in higher R^2 and lower RMSE compared to the baseline. This effect is particularly beneficial for models that do not explicitly capture temporal dependencies, as they cannot infer hidden states over time. In contrast, recurrent architectures such as LSTM and GRU can leverage sequential patterns to partially compensate for missing state information.

For models designed to handle sequential data, such as LSTM and GRU, the full configuration also led to substantial improvements, reinforcing the importance of leveraging temporal dependencies for enhanced predictive accuracy. These models exhibited significant gains over the baseline, showing that access to complete state information contributes to better trajectory estimation.

Regarding computational efficiency, the GRU model had the shortest training time in the baseline configuration, while MLP maintained the fastest inference time across both

configurations. The inclusion of additional state variables slightly increased computational demand across all models, but the improvements in predictive performance justify the trade-off.

Among all tested models, GRU in the full configuration achieved the best balance of accuracy and computational efficiency, making it a strong candidate for real-time maneuver prediction tasks. However, in real-world scenarios, accessing all state variables may not always be feasible due to sensor limitations, communication delays, or operational constraints. These factors highlight the relevance of the baseline configuration, where synthetic data played a key role in compensating for missing information, demonstrating its potential to enhance model robustness under real-world constraints.

Figure 13.7 provides a detailed comparison between the actual and predicted actions for JX, JY, and Throttle during the pop-up attack maneuver, specifically illustrating the results from the best-performing model, the GRU in the full configuration. At the beginning of the maneuver, there is a slight delay of 5 steps due to the window size, as the full sequence is divided into overlapping windows of fixed length $L = 5$. Solid blue lines represent the mean of the actions taken by the pilot, while the dashed red lines illustrate the model's predictions. The shaded regions around the mean of the pilot's actions indicate the standard deviation, calculated from the dataset of recorded maneuvers, to visually represent variability in the pilot's actions. Similarly, the shaded regions around the predicted actions show the standard deviation of the model's predictions across multiple test sequences. This highlights the model's consistency in capturing the maneuver patterns, as well as its robustness in generating realistic action sequences across different test instances.

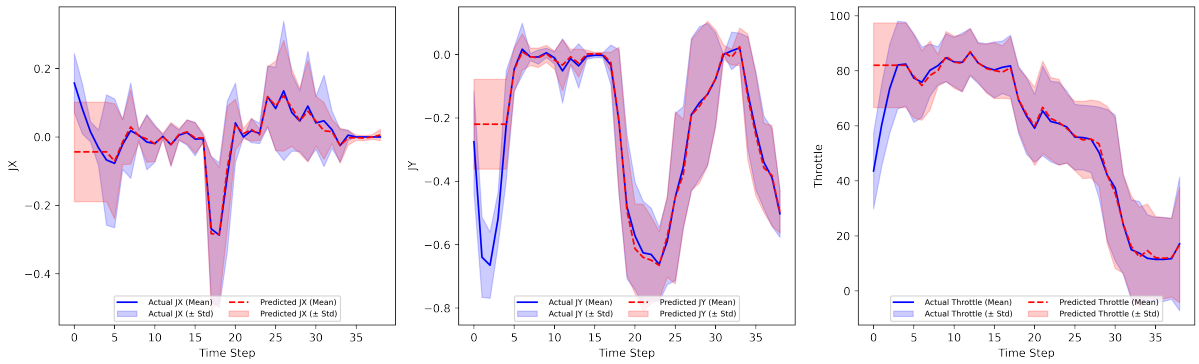


FIGURE 13.7 – Trajectory comparison – actual vs predicted actions with mean and standard deviation for pitch (JX), roll (JY), and Throttle during the pop-up attack maneuver.

Examining the plots in Figure 13.7, we observe the following key insights regarding the model's ability to replicate the pop-up maneuver:

- **JX:** The GRU-based model closely mirrors the actual pitch actions, maintaining alignment with the general trend throughout the maneuver. However, during seg-

ments with higher variability, the model slightly underestimates the amplitude. This underestimation is still within the standard deviation range, indicating an acceptable degree of accuracy.

- **JY:** The roll predictions are remarkably accurate, especially in capturing sharp transitions. The model's predictions align closely with the actual roll actions, deviating only slightly at certain peaks. These minor deviations fall within the shaded variability range, highlighting the model's effectiveness in tracking rapid changes in roll.

- **Throttle:** The model demonstrates a high level of accuracy in predicting throttle actions, especially during prolonged phases of consistent throttle application. Minor discrepancies appear during abrupt throttle transitions; however, these remain well within the standard deviation range, showcasing the model's robustness in handling throttle adjustments.

Examining the plots in Figure 13.7, we observe that the visual alignment between the predicted and actual actions across all three control inputs reinforces the quantitative performance metrics, such as R^2 and RMSE. This confirms the GRU model's ability to capture the intricate dynamics of the pop-up maneuver. The shaded regions around the actual actions, which represent pilot variability, demonstrate that the model's predictions consistently fall within the expected range. This further supports the model's potential applicability in both pilot training and autonomous maneuver systems.

13.6.2 Synthetic Data for Model Improvement

To assess the impact of synthetic data on model performance, we incrementally added synthetic flight samples to the training and validation datasets while keeping the test dataset consistent across all experiments. This approach enabled a systematic evaluation of how synthetic data affects model generalization and prediction accuracy for maneuver trajectories. Since the inference time of the models should not depend on the amount of training data and was already analyzed in previous sections, we only report the training time in this subsection. We present our findings separately for the **baseline configuration** (Section 13.6.2.1) and the **full configuration** (Section 13.6.2.2) to highlight how the inclusion of synthetic data influences model performance under different levels of available real-world features.

13.6.2.1 Baseline Configuration

Table 13.3 presents the results for the baseline configuration, bringing the number of synthetic samples, R^2, RMSE, and training time in seconds. Results are shown as mean ± standard deviation over five runs. The best R^2, RMSE, and training time for each model are highlighted in **blue**.

TABLE 13.3 – Performance metrics for imitation learning models in baseline configuration when synthetic samples are added to the training and validation datasets. Values are presented as mean ± standard deviation over five runs. The best R^2, RMSE, and Training Time for each model are highlighted in blue.

Model	Number of Synthetic Samples Added	R^2	RMSE	Training Time (s)
MLP	7	0.277 ± 0.069	$\mathbf{8.831 \pm 0.012}$	77.33 ± 0.82
	15	0.312 ± 0.066	8.895 ± 0.019	$\mathbf{74.41 \pm 4.82}$
	30	0.414 ± 0.074	9.141 ± 0.034	242.93 ± 3.55
	45	0.408 ± 0.121	9.158 ± 0.016	261.88 ± 1.48
	60	$\mathbf{0.455 \pm 0.053}$	9.313 ± 0.047	315.89 ± 4.34
	150	0.439 ± 0.043	9.324 ± 0.024	453.48 ± 3.58
LSTM	7	0.680 ± 0.003	8.638 ± 0.043	$\mathbf{43.85 \pm 0.89}$
	15	0.679 ± 0.005	8.644 ± 0.079	54.76 ± 1.54
	30	0.680 ± 0.004	8.663 ± 0.077	77.03 ± 4.85
	45	$\mathbf{0.681 \pm 0.003}$	$\mathbf{8.557 \pm 0.044}$	85.81 ± 7.15
	60	0.680 ± 0.005	8.618 ± 0.112	107.61 ± 3.00
	150	0.681 ± 0.004	8.581 ± 0.134	214.97 ± 27.72
GRU	7	$\mathbf{0.683 \pm 0.002}$	8.660 ± 0.076	$\mathbf{35.59 \pm 0.81}$
	15	0.683 ± 0.005	8.692 ± 0.119	64.91 ± 10.91
	30	0.681 ± 0.004	8.691 ± 0.131	72.95 ± 8.01
	45	0.682 ± 0.005	8.578 ± 0.092	79.99 ± 4.13
	60	0.680 ± 0.007	8.692 ± 0.111	113.06 ± 9.86
	150	0.681 ± 0.006	$\mathbf{8.520 \pm 0.143}$	142.02 ± 20.36

For this configuration, synthetic data had a notable impact on the performance of models, particularly for the MLP architecture. The inclusion of synthetic samples improved the MLP's R^2, with the highest value observed at 60 synthetic samples. However, RMSE slightly increased beyond this point, suggesting a reduced benefit as more synthetic data was added. This trend likely occurs because, while additional synthetic data expands the training set, it may also introduce redundancy or noise, limiting further improvements in predictive accuracy.

In contrast, recurrent architectures such as LSTM and GRU exhibited more stable performance across different numbers of synthetic samples, with only slight variations in R^2 and RMSE. This stability suggests that these models already extract sufficient temporal patterns from the available real data, making additional synthetic samples less impactful. Unlike MLP, which does not explicitly model temporal dependencies and can benefit from increased data diversity, LSTM and GRU inherently capture sequential patterns, reducing their reliance on dataset augmentation.

Training time increased consistently with the number of synthetic samples for all models. However, MLP showed the most significant increase, likely due to its fully connected structure requiring more updates per additional sample. In contrast, GRU remained the most computationally efficient model, benefiting from its simplified gating mechanism

compared to LSTM. The key reason for GRU's efficiency can be attributed to its architectural design. Unlike LSTM, which has separate forget, input, and output gates, GRU combines these operations into fewer gates, specifically the update and reset gates. This more compact structure reduces the number of parameters and matrix operations required per timestep, leading to faster training times (GOODFELLOW *et al.*, 2016).

13.6.2.2 Full Configuration

Table 13.4 presents the results for the full configuration. As in the baseline case, the table reports the number of synthetic samples, R^2, RMSE, and training time in seconds, with the best results for each model highlighted in **blue.**

TABLE 13.4 – Performance metrics for imitation learning models in full configuration when synthetic samples are added to the training and validation datasets. Values are presented as mean ± standard deviation over five runs. The best R^2, RMSE, and Training Time for each model are highlighted in blue.

Model	Number of Synthetic Samples Added	R^2	RMSE	Training Time (s)
MLP	7	**0.698 ± 0.085**	**2.230 ± 0.050**	186.20 ± 3.00
	15	0.616 ± 0.094	2.530 ± 0.099	**171.13 ± 2.38**
	30	0.598 ± 0.078	2.499 ± 0.089	235.22 ± 0.62
	45	0.527 ± 0.146	2.497 ± 0.091	280.90 ± 9.47
	60	0.472 ± 0.140	2.620 ± 0.057	328.74 ± 97.39
	150	0.397 ± 0.213	2.782 ± 0.146	544.65 ± 15.61
LSTM	7	**0.957 ± 0.002**	**2.099 ± 0.115**	**71.71 ± 7.53**
	15	0.951 ± 0.002	2.271 ± 0.093	80.02 ± 2.21
	30	0.950 ± 0.003	2.362 ± 0.076	77.50 ± 2.10
	45	0.947 ± 0.004	2.472 ± 0.093	107.83 ± 1.35
	60	0.944 ± 0.004	2.475 ± 0.128	119.31 ± 0.90
	150	0.939 ± 0.003	2.883 ± 0.172	144.32 ± 11.47
GRU	7	**0.958 ± 0.002**	**2.132 ± 0.073**	**51.38 ± 1.18**
	15	0.954 ± 0.004	2.244 ± 0.139	61.49 ± 9.20
	30	0.948 ± 0.005	2.478 ± 0.157	94.79 ± 13.24
	45	0.945 ± 0.004	2.537 ± 0.146	104.07 ± 12.96
	60	0.943 ± 0.005	2.672 ± 0.155	138.90 ± 6.39
	150	0.937 ± 0.007	2.805 ± 0.201	164.40 ± 13.79

In this configuration, synthetic data was not only less useful but, in many cases, made performance worse for all models. Unlike in the baseline setup, where MLP still benefited from synthetic data, in the full configuration, adding synthetic samples consistently led to lower R^2 values and higher RMSE across MLP, LSTM, and GRU. The best results for all models occurred with the smallest number of synthetic samples (7 for LSTM, GRU, and MLP), and performance dropped as more synthetic data was added.

A possible reason for this is that, in the full configuration, models already had access to a richer set of real-world features that provided enough information for learning. The synthetic data, instead of adding useful diversity, may have introduced inconsistencies or patterns that did not fully match the real data distribution. This could have confused the models, leading to worse generalization.

Training time also increased as more synthetic samples were added, which was expected. However, since the extra data did not improve performance, this additional

computational cost brought no real benefit in this configuration.

13.6.2.3 Summary of Findings

The evaluation of synthetic data inclusion across different configurations highlights key insights:

- **MLP improved in the baseline configuration:** MLP showed the most gains with synthetic data, reaching the highest R^2 at 60 synthetic samples. Beyond this, RMSE increased, likely due to overfitting or redundant patterns.
- **LSTM and GRU remained stable in the baseline configuration:** Both models showed minimal variation across different amounts of synthetic data, indicating that they effectively captured temporal dependencies from real data without requiring augmentation.
- **Synthetic data reduced performance in the full configuration:** Unlike in the baseline, synthetic samples consistently lowered R^2 and increased RMSE in the full configuration for all models, likely introducing inconsistencies instead of useful variability.
- **Training time increased:** As expected, adding synthetic data raised training time for all models. MLP saw the highest increase, while GRU remained the most efficient.

Notably, the baseline configuration provides a more realistic approximation of real-world operational constraints, as pilots often rely on a partial state space with a limited subset of available variables for decision-making. In this context, synthetic data proved to be effective, expanding the training set and introducing additional variability that contributed to improved performance. These findings highlight the potential of synthetic data to mitigate data scarcity and enhance model robustness in environments where real-world features are limited.

13.7 Outcomes

This study successfully developed models to replicate the complex pop-up attack maneuver in air combat, utilizing simulated flight data from an FAB fighter pilot. Through imitation learning techniques with MLP, LSTM, and GRU networks, the research demonstrated that these models could effectively predict aircraft control inputs, closely mimicking the execution patterns of experienced pilots. Validated through cross-validation

and test group evaluations, the models showed strong potential for integration into autonomous combat systems.

The BC approach achieved satisfactory replication of the pop-up maneuver, especially when using state variables that reflect temporal aspects, such as angular and linear velocities and accelerations. Among the tested models, the GRU network achieved the highest performance, effectively capturing temporal dependencies while maintaining computational efficiency, making it particularly suitable for autonomous maneuver execution.

Additionally, we explored generative learning techniques to produce synthetic data that closely mirrors the collected flight data. This approach has the potential to enhance model performance, particularly in scenarios where the agent's state space is not fully known or when non-temporal-dependent algorithms are used. Initial findings suggest that integrated synthetic data can improve model performance and enhance generalization.

While these results are promising, there is room for improvement. Future work will focus on expanding the dataset by collecting additional flight recordings from pilots with different profiles and capturing a more diverse range of strategies and variations in maneuver execution. Expanding data collection to include a wider variety of maneuvers relevant to air combat operations, such as missile evasion and engagement with enemy aircraft, will further improve model generalization. Additionally, evaluations with real flight data will be considered, enabling model updates and reducing the gap between simulation-based training and real-world applicability.

Despite the challenges associated with reinforcement learning, as discussed at the beginning of this work, it remains a valuable research direction for this domain. Future work could also investigate how reinforcement learning can be used to discover innovative approaches for executing ground attack maneuvers, potentially identifying strategies that differ from traditional expert-driven tactics. Moreover, future studies could explore testing alternative imitation and generative learning approaches, such as adversarial imitation learning (AIL) and transformer-based sequence models, which have shown promise in sequential decision-making tasks (HO; ERMON, 2016; CHEN *et al.*, 2021). Combining reinforcement learning with imitation learning could also be explored as a way to balance stability with adaptability, allowing autonomous agents to refine their skills while maintaining the benefits of expert guidance (NAIR *et al.*, 2018).

Furthermore, future studies should explore how well the models handle sudden and unforeseen operational states, such as extreme maneuvers, unexpected inputs, and variations in mission parameters such as different aircraft configurations and threat levels, ensuring that autonomous agents can adapt beyond the conditions seen during training. Another important direction is addressing the risk of excessive dependence on synthetic data by improving the balance between real and synthetic samples during training, vali-

dating the models with real flight data, and refining synthetic data generation to better match a large variety of real-world conditions.

Source Code

The source code for this research is publicly available at `https://github.com/jpadantas/pop-up_attack_generative`. This repository contains the scripts and models used in this work. The flight data, however, is not available due to its classified nature.

14 Social Navigation in Air Combat

This chapter examines the dynamics of social navigation in human-autonomous collaboration, focusing on simulated air combat scenarios. It explores methodologies and strategies to improve coordination between human pilots and AI-controlled wingmen. Key contributions include the definition of social navigation metrics and the design of a user study to assess human–AI interaction.

As the final technical chapter of this thesis, it proposes a user study experiment to evaluate how pilots interact with autonomous agents in shared airspace during air combat. The goal is to assess the effectiveness, naturalness, and safety of these interactions. Building on the metrics introduced, the study offers a practical step toward validating human-AI teaming in operational settings. As indicated in Figure 1.2, this chapter primarily contributes to the "Aerial Autonomous Agents" area within the proposed research framework.

The research presented in this chapter is based on the following work:

> **DANTAS, J. P. A.**; MAXIMO, M. R. O. A.; YONEYAMA, T. Loyal Wingman Assessment: Social Navigation for Human-Autonomous Collaboration in Simulated Air Combat. In: Proceedings of the 38th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation. Proceedings [...]. New York, NY, USA: Association for Computing Machinery, 2024. (SIGSIM-PADS '24), p. 61–62. ISBN 9798400703638. Available at: `https://doi.org/10.1145/3615979.3662149`.

14.1 Summary

This study proposes social navigation metrics for autonomous agents in air combat, aiming to facilitate their smooth integration into pilot formations. The absence of such metrics poses challenges to safety and effectiveness in mixed human-autonomous teams. The proposed metrics prioritize naturalness and comfort. We suggest validating them through a user study involving military pilots and autonomous loyal wingmen in simulated air combat scenarios. The experiment will include setting up simulations, designing

scenarios, and evaluating performance using feedback from questionnaires and data analysis. These metrics aim to enhance the operational performance of autonomous loyal wingmen, thereby contributing to safer and more strategic air combat.

14.2 Introduction

Autonomous aircraft in shared airspace must navigate safely and efficiently while adhering to social norms expected in human-centric environments (GAO; HUANG, 2021). These norms include respecting personal space (ALTHAUS *et al.*, 2004), maintaining comfortable velocities and accelerations (KATO *et al.*, 2015), and keeping a safe distance from other aircraft (GLOZMAN *et al.*, 2021). Research into socially aware navigation aims to improve interactions between autonomous agents and humans (KRUSE *et al.*, 2013), but there is still a need for new metrics to evaluate these methods more effectively (WANG *et al.*, 2022).

The air combat domain introduces additional complexity layers to social navigation (BIRKELAND, 2018). Integrating a loyal wingman with human pilots requires safety and efficiency and a deep understanding of tactics and formation dynamics, demanding a sophisticated mix of social and combat skills (DANTAS *et al.*, 2023). This paper adapts socially aware navigation for air combat by introducing tailored social navigation metrics for autonomous wingmen.

Our main contribution is developing these metrics and proposing a validation process through a user study experiment with military pilots in high-fidelity simulations. This research addresses a gap in the existing literature and sets the stage for future integration of autonomous systems in manned military operations, enhancing both effectiveness and social compatibility.

14.3 Proposed Metrics

We propose key metrics focused on naturalness and comfort to assess autonomous agents in air combat. Table 14.1 outlines these metrics, detailing the aspects evaluated and the rationale for each.

Naturalness: This aspect evaluates the similarity of the wingman's motion to human movements and the smoothness of its path (KRUSE *et al.*, 2013). It involves analyzing the agent's velocity, acceleration, and jerk to assess movement smoothness and human-likeness, crucial for human trajectory prediction research (RUDENKO *et al.*, 2020). Humans typically exhibit trajectories with compatible velocities, accelerations, and min-

TABLE 14.1 – Summary of the proposed social navigation metrics.

No.	Aspect	Metric	Description
M_1	Naturalness	Velocity	Computes the mean of the squared velocities over the time period, highlighting significant speed variations from typical human norms
M_2	Naturalness	Acceleration	Calculates the average of squared accelerations to assess how naturally the acceleration changes compare to human-like movements
M_3	Naturalness	Jerk	Evaluates the mean squared jerk to identify abrupt changes in acceleration, aiming for smoother, more human-like trajectories
M_4	Comfort	Minimum Distance	Calculates the smallest distance between two agents by iteratively comparing their positions over a given time period and updating the minimum found
M_5	Comfort	Collision Risk	Assesses the collision risk by determining how often two agents come within a critical distance or have a closing velocity that predicts a potential collision

imal jerk (KRUSE *et al.*, 2013; WANG *et al.*, 2022). To measure the wingman's trajectory smoothness, we calculate its average velocity, acceleration, and jerk to determine if these averages meet predefined thresholds that approximate human pilot levels, which depend on the type of aircraft. The squared derivatives ensure non-negativity, highlight significant variations, and smooth noise for easier mathematical handling

Refer to Equation 14.1 for the naturalness metrics calculation. In the equation, p denotes position, with w and h representing wingman and human, respectively. Superscripts w or h indicate affiliation, subscript t denotes the current time, and T is the total episode duration. The symbol n indicates the derivative order, where $n = 1, 2, 3$ for velocity, acceleration, and jerk, respectively.

$$M_n = \frac{1}{T}\sum_{t=0}^{T}\left(\frac{d^n p(t)}{dt^n}\right)^2, \quad \text{where } n = \begin{cases} 1 & \text{for velocity,} \\ 2 & \text{for acceleration,} \\ 3 & \text{for jerk.} \end{cases} \tag{14.1}$$

Comfort: Quantifying the comfort level of human pilots interacting with an autonomous loyal wingman is a complex task that cannot be captured by a single metric (RIOS-MARTINEZ *et al.*, 2012; RIOS-MARTINEZ *et al.*, 2015; KOTHARI *et al.*, 2021). Research indicates that the psychological comfort of humans is significantly influenced by the size of their personal space and how well others respect these boundaries (AIELLO, 1977; BALDASSARE, 1978; GREENBERG *et al.*, 1980; HALL, 1990). Therefore, we propose two metrics to evaluate personal space to measure the comfort of human pilots in shared

airspace operations with autonomous loyal wingmen.

The first comfort metric measures the smallest distance maintained between the human and the wingman throughout the air combat simulation. For specific calculation details, see Algorithm 3.

Algorithm 3 Calculate M_4: Minimum Distance Comfort Metric

Initialize minimum distance comfort metric: $M_4 \leftarrow +\infty$
for $t = 0$ to T **do**
    Extract human position $\mathbf{p}_t^h$ and wingman position $\mathbf{p}_t^w$ for frame t
    Calculate distance for frame t: $d_t \leftarrow \left\|\mathbf{p}_t^w - \mathbf{p}_t^h\right\|$
    if $d_t < M_4$ **then**
        Update minimum distance: $M_4 \leftarrow d_t$ ▷ Record new minimum across all frames
    end if
end for

The second comfort metric assesses the safety of autonomous aircraft operations by calculating the risk of collisions based on the principles of the Time to Closest Point of Approach (TCPA) (GLOZMAN *et al.*, 2021). The collision risk comfort metric, M_5, increments in situations where the distance d between an autonomous aircraft and a wingman is less than a critical threshold ε, recommended to be set at 0.5 nautical miles for initial trials (DOSHI *et al.*, 2000), or when the closing velocity $v_{\text{close},t}$ indicates a decreasing distance that could lead to a collision within a critical time frame t_{critical}. This metric effectively integrates both proximity and Time to Reach (TTR), which calculates the time until a potential collision by dividing the distance by the closing velocity, providing a comprehensive evaluation of collision risks. Refer to Figure 14.1 and Algorithm 4 for implementation details.

Algorithm 4 Calculate M_5: Collision Risk Comfort Metric

Initialize collision risk comfort metric: $M_5 \leftarrow 0$
for $t = 1$ to T **do**
    Calculate relative position vector: $\mathbf{r}_t \leftarrow \mathbf{p}_t^h - \mathbf{p}_t^w$
    Compute distance: $d_t \leftarrow \|\mathbf{r}_t\|$
    Compute relative velocity vector: $\mathbf{v}_t \leftarrow \frac{d\mathbf{p}_t^h}{dt} - \frac{d\mathbf{p}_t^w}{dt}$
    Calculate closing velocity: $v_{\text{close},t} \leftarrow \frac{\mathbf{r}_t \cdot \mathbf{v}_t}{d_t}$
    if $d_t < \varepsilon$ **then** ▷ If within critical distance
        Increment collision risk comfort metric: $M_5 \leftarrow M_5 + 1$ ▷ Log alert
    else
        if $v_{\text{close},t} > 0$ **then** ▷ If distance decreasing
            Calculate Time to Reach (TTR): $TTR \leftarrow \frac{d_t}{v_{\text{close},t}}$
            if $TTR < t_{\text{critical}}$ **then** ▷ If below critical time
                Increment collision risk comfort metric: $M_5 \leftarrow M_5 + 1$ ▷ Log alert
            end if
        end if
    end if
end for

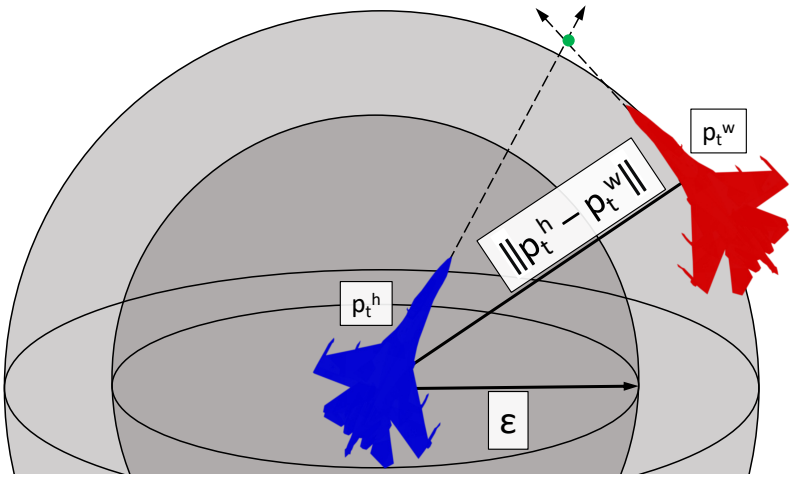


FIGURE 14.1 – Human-wingman personal space metric depiction.

14.4 User Study Experiment

This user study aims to validate social navigation metrics by comparing them against human pilot perceptions in simulated air combat scenarios, functioning as a form of Turing test for the autonomous wingman. The experiment will assess whether the wingman's behaviors, guided by social navigation metrics, are sufficiently natural to be indistinguishable from those of a human co-pilot. Military pilots with varied experience levels – selected based on flight hours, system proficiency, and simulation expertise – will participate, ensuring a broad range of insights.

14.4.1 Blinded Setup

To ensure unbiased perceptions, the study proposes a "blinded" design in which pilots will not be informed if they are flying alongside another human pilot or an AI agent. This approach is intended to isolate the effects of the social navigation metrics on pilot perceptions without introducing bias that could arise from knowing the wingman's nature.

Each pilot will complete a series of trials in a high-fidelity simulation framework, the Aerospace Simulation Environment (DANTAS *et al.*, 2022a; DANTAS *et al.*, 2023a). Although this study has not yet been conducted, this thesis proposes a methodology for future experimental evaluations.

For initial planning purposes, we propose conducting three trials per pilot to assess performance across different conditions. In each trial, pilots will perform Combat Air Patrol (CAP) missions, which are air defense operations focused on protecting a specific area from potential threats (DANTAS *et al.*, 2021a), alongside one of three versions of a wingman, each representing a different level of control and autonomy.

- **Human-Controlled Wingman**: Another human pilot controls the wingman, providing a baseline for human-human teaming perceptions.
- **AI Baseline Wingman**: An AI wingman with baseline navigation and engagement capabilities, representing standard autonomous behavior.
- **Advanced AI Wingman**: An AI wingman utilizing enhanced social navigation metrics designed to closely emulate human-like teamwork behaviors.

The pilots will remain unaware of their wingman's nature (human, AI baseline, or advanced AI) across trials, enhancing the study's objectivity in assessing the agent's performance based on behavior alone. This design allows for a robust comparison of perceptions across the different wingman types, helping validate the effectiveness of the social navigation metrics in approximating human-like behaviors in autonomous agents.

14.4.2 Design and Metrics Evaluation

The Defensive Counter Air (DCA) index (DANTAS *et al.*, 2021a) will be employed as a metric to evaluate the human-AI team's performance in achieving mission objectives. Objective performance metrics, such as reaction times, formation cohesion, and zone coverage efficiency, will be recorded during each trial.

Following each trial, pilots will complete a post-trial questionnaire to capture their subjective perceptions of the interaction with their wingman. These subjective assessments are essential for validating the social navigation metrics, as they reflect the pilots' perceptions of the agent's alignment with human behaviors and expectations.

14.4.3 Post-Trial Questionnaire

The post-trial questionnaire will use a 1 to 5 Likert scale format for each question. The response options for the main questions are as follows:

[1] **Strongly Disagree**

[2] **Disagree**

[3] **Neutral**

[4] **Agree**

[5] **Strongly Agree**

For the final question, which assesses the pilot's perception of the wingman's identity, the response options are:

[1] **Definitely Human**

[2] **Probably Human**

[3] **Unsure**

[4] **Probably AI**

[5] **Definitely AI**

Below are the specific questions aligned with the core aspects of social navigation metrics:

- **Naturalness:**
  - *The wingman's movements felt smooth and natural throughout the mission.*
  - *The wingman's speed and changes in direction were consistent with what I would expect from a human pilot.*
  - *The wingman's behavior made it easy for me to anticipate its actions.*
- **Comfort:**
  - *The wingman maintained a safe and comfortable distance from my aircraft.*
  - *I felt safe operating near the wingman, with minimal risk of near-collisions.*
  - *The wingman's presence did not cause distractions or additional challenges to my performance.*

In addition to naturalness and comfort, the following items assess overall trust, effectiveness, and the pilot's perception of the wingman's identity.

- **Trust:**
  - *I felt confident relying on the wingman in high-pressure situations.*
  - *I trusted the wingman's ability to effectively support the mission objectives.*
- **Effectiveness and Adaptability:**
  - *The wingman adapted well to changing tactical conditions.*

  - *The wingman's actions complemented my strategy without creating additional challenges.*

- **Situational Awareness and Coordination:**
  - *The wingman maintained good situational awareness and coordinated its actions with mine.*
  - *The wingman's behavior positively impacted my performance throughout the mission.*

- **Perception of Wingman's Identity:**
  - *Do you believe you were flying with a human pilot or an AI wingman during this mission?*

14.5 Sample Size

Determining the appropriate sample size is essential to ensure that the study has sufficient statistical power to detect meaningful differences between the behaviors of the wingman being evaluated. In this study, pilots will fly three times, corresponding to two AI wingman behaviors and one human-controlled wingman behavior, in a repeated-measures design. A power analysis was conducted following the guidelines outlined in Cohen (1988) and implemented using the `statsmodels` Python package (SEABOLD; PERKTOLD, 2010).

The following parameters were considered in the calculation:

- **Effect size (`f`)**: A medium effect size of 0.5 was assumed, representing the expected magnitude of differences between the three conditions.
- **Significance level (α)**: A threshold of 0.05 was used, representing a 5% probability of incorrectly rejecting the null hypothesis.
- **Statistical power ($1 - \beta$)**: A power level of 0.8 was chosen, ensuring an 80% probability of detecting a true effect if it exists.
- **Number of conditions (`k`)**: Three conditions were included, corresponding to two AI wingman behaviors and one human-controlled wingman behavior.
- **Repeated measures (`m`)**: Each pilot participated in all three conditions, providing data for repeated measurements.
- **Intra-class correlation (ρ)**: A correlation of 0.5 between repeated measures was assumed, reducing the total sample size required.

Using these parameters, the required sample size was calculated in two steps. First, the unadjusted sample size per condition for a general ANOVA was calculated using the `statsmodels` Python package, which incorporates refined computational methods for power analysis. This resulted in:

$$n_{\text{per condition}} = 41.07.$$

Next, adjustments were made for the repeated-measures design, which increases statistical efficiency. The adjusted total sample size accounts for the intra-class correlation (ρ) and the number of repeated measures (m) (SHROUT; FLEISS, 1979):

$$n_{\text{adjusted}} = \frac{n_{\text{per condition}}}{1 + (m-1) \cdot \rho} = \frac{41.07}{1 + (3-1) \cdot 0.5} = \frac{41.07}{2} = 20.53.$$

The relationship between the sample size and effect size for both general ANOVA and repeated measures is illustrated in Figure 14.2. This plot highlights how the adjusted sample size for repeated measures is significantly reduced due to the intra-class correlation and statistical efficiency of repeated designs.

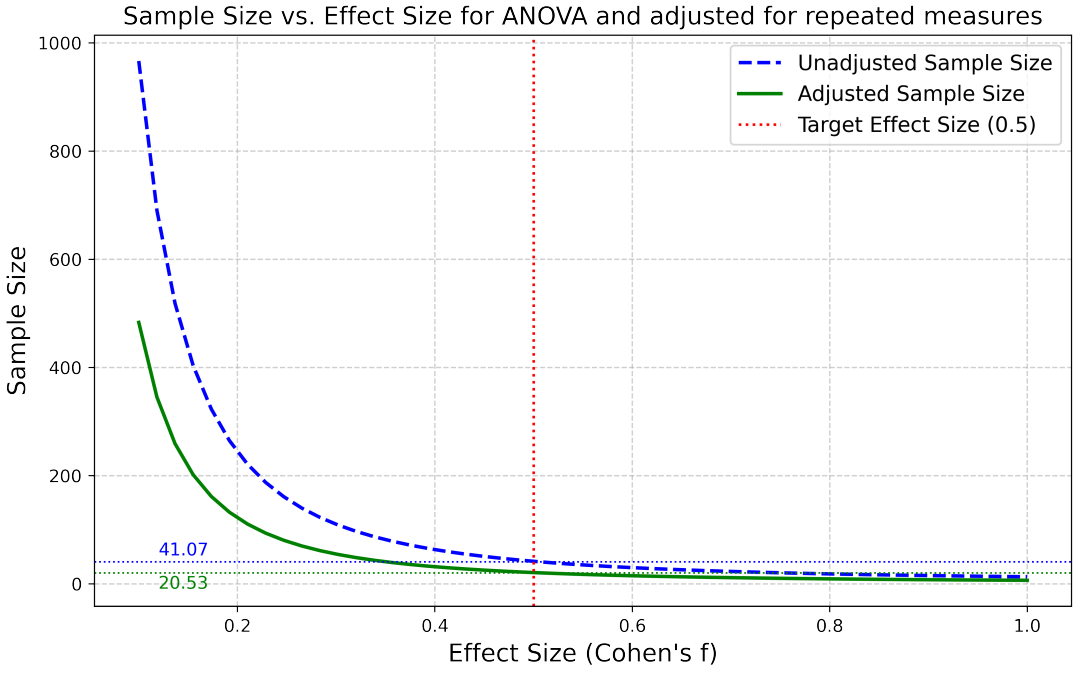


FIGURE 14.2 – Sample Size vs. Effect Size for General ANOVA and Repeated Measures. The blue dashed line represents the unadjusted sample size for general ANOVA, while the green solid line represents the adjusted sample size for repeated measures. The red vertical dotted line highlights the target effect size (`f` $= 0.5$).

In conclusion, approximately **21 unique pilots** are needed to achieve the desired power while accounting for repeated measurements. This ensures statistical robustness while considering practical constraints such as pilot availability and resource allocation.

14.5.1 Data Analysis

Collected data will be analyzed using the AsaPy Library (DANTAS *et al.*, 2024) to integrate and correlate both subjective questionnaire responses and objective performance metrics. This analysis aims to validate the social navigation metrics, demonstrating their applicability in human-autonomous air combat teaming. These insights are expected to refine the social navigation models, optimizing autonomous agents' algorithms for effective teamwork in CAP and other air combat missions.

This detailed user study design ensures that the social navigation metrics align with the operational expectations and preferences of human pilots, contributing to safer and more effective human-autonomous collaboration in defense scenarios.

14.6 Outcomes

This chapter's investigation into social navigation within human-autonomous collaboration in simulated air combat has produced several key insights and preliminary results. Firstly, the development of social navigation metrics has provided a framework for assessing the naturalness and comfort of interactions between human pilots and autonomous wingmen. These metrics, focusing on velocity, acceleration, jerk, minimum distance, and collision risk, offer a quantitative basis for evaluating the performance and safety of mixed human-autonomous teams.

Secondly, a user study experiment involving military pilots has been designed to validate these metrics through an approach inspired by the Turing Test. Although this experiment was not conducted within the scope of this thesis, it remains a proposed methodology for future research. In this Turing Test-inspired approach, the experiment would assess whether the behavior of the autonomous wingman, guided by the social navigation metrics, is indistinguishable from that of a human co-pilot. By gathering pilot feedback on naturalness, comfort, and perceived identity, this study would evaluate the degree to which the wingman's actions align closely with human behavior, promoting trust and effective collaboration. This planned setup thus offers a practical evaluation framework to further validate the metrics' effectiveness in bridging the perceptual gap between human and autonomous agents.

Although the research is still in progress, initial observations of autonomous agents suggest that the proposed metrics could effectively enhance the operational performance of autonomous loyal wingmen. The study, which collects feedback through questionnaires and data analysis, seeks to bridge the gap between theoretical models and practical applications. Positive pilot perceptions of the wingman's naturalness and comfort could

support the eventual integration of autonomous systems into manned military operations.

Overall, this work contributes to the main goal of integrating autonomous systems into crewed military operations, promoting safer and more strategic air combat scenarios. The ongoing nature of this research highlights the need for continuous refinement and validation of the proposed metrics to ensure their relevance and effectiveness in dynamic and demanding environments.

Currently, the proposed social navigation metrics primarily focus on the performance and behavior of the autonomous system within the mixed team. For future work, it is suggested to create additional metrics that enable a more detailed evaluation of the integrated human-autonomous team in the same environment, benchmarking their collective performance against scenarios involving only human pilots or only autonomous agents. Developing such comparative metrics would allow for a more realistic assessment of the true effectiveness of human-autonomous teams in simulated environments. The next steps include refining these metrics and conducting the proposed user study, with the suggested Turing Test-inspired design to guide future investigations.

One important aspect that remains as a proposal for future work is the practical adoption of the system by pilots. Although understanding how pilots would incorporate and rely on such human-autonomous collaboration systems is fundamental for real-world implementation, this thesis has primarily focused on the development and validation of the core metrics and evaluation framework. The investigation into actual pilot adoption, training implications, and integration into operational routines is acknowledged as essential, but was beyond the scope of the present research and is identified as a critical next step for future studies.

14.7 Source Code

The source code for the proposed social navigation metrics is available on GitHub. You can access it at `https://github.com/jpadantas/social-navigation-metrics` to explore and replicate the same metrics discussed in this chapter.

Part VII

Final Considerations

15 Conclusion and Future Research

In this final chapter, we synthesize the key findings of this research and outline future directions in the field of decision support and autonomy for aerial warfare. The integration of machine learning and simulation has been a central theme throughout this thesis, offering innovative solutions and advancements to air combat operations.

We begin by summarizing the main contributions made during this research journey. The findings highlight the development of simulation tools and services, the accurate modeling of weapon systems, the implementation of in-flight decision support systems, the creation of advanced aerospace data analytics tools, and the deployment of autonomous agents. Each of these elements plays an essential role in advancing the capabilities of modern air forces and contributes to a comprehensive research framework that integrates simulation and machine learning for testing and evaluation in realistic scenarios.

Following the summary of findings, we present recommendations for future research. These recommendations aim to address the existing limitations and explore new opportunities for further improving machine learning and simulation integration in aerial warfare. Areas such as enhanced validation with real-world data, human-autonomous collaboration, advanced machine learning algorithms, hardware integration, and the expansion of data analysis tools are identified as key focus areas for future studies.

Lastly, this chapter presents final remarks that reflect on the broader impact of this research. These reflections emphasize practical considerations, institutional challenges, and strategic perspectives on integrating simulation and machine learning in modern air combat operations.

15.1 Summary of Findings

The key findings and contributions of this research are summarized as follows:

- **Simulation Tools and Services**: Presenting a robust simulation environment and supporting services contributing to FAB's digital transformation efforts. These tools offer a comprehensive platform for testing and evaluating machine learning models

in realistic scenarios.

- **Weapon Systems**: Accurate modeling and analysis of air-to-air and ground-to-air missile performance were conducted. The machine learning models developed proved effective in predicting engagement zones and estimating the probability of kill.
- **In-Flight Tactical Systems**: Machine learning techniques were successfully applied to enhance situational awareness and support decision-making in aerial combat. Supervised learning approaches effectively optimized missile launch timing, improving engagement strategies and mission outcomes.
- **Aerospace Data Analytics**: The implementation of dedicated tools for aerospace simulation analysis enabled the efficient processing of large-scale data. These tools supported data visualization, statistical analysis, and the integration of machine learning models, helping analysts and operators make better decisions.
- **Aerial Autonomous Agents**: Deploying autonomous agents using deep reinforcement, imitation, and generative learning showed promising results in defense scenarios. These agents demonstrated the ability to perform complex combat tasks, enhancing mission success rates and operational efficiency.

15.2 Future Work and Recommendations

While this thesis has advanced the use of simulation and machine learning for decision support and autonomy in air combat operations, several key areas remain for future research to improve the effectiveness, robustness, and operational relevance of these approaches. The recommendations below address current limitations and suggest ways to build on the contributions of this study:

- **Validation with Real-World Data**: To build trust and improve the accuracy of decision support systems and autonomous agents, validating simulation and machine learning models with real-world operational data is essential. Future studies should focus on collecting and integrating high-quality flight and combat data from live exercises to calibrate and test these models. Incorporating real-world data will also help reduce the gap between simulated and real missions, making it easier for military personnel to understand, accept, and use these systems in practice.
- **Human-Autonomous Collaboration**: As autonomous agents become more common in aerial combat, it is important to ensure that they can work well and safely with human pilots. Future research should look into improving communication and shared decision-making between humans and autonomous agents. This includes developing

solutions that allow real-time interaction and coordination, supported by fast and reliable computing systems. Studies with users can also help us better understand how people feel when working with autonomous agents and guide the creation of simple measures to improve comfort and trust. These efforts will help humans and autonomous systems work better together in both simulations and real missions.

- **Advanced Machine Learning Algorithms**: Addressing the complexity and unpredictability of aerial warfare requires going beyond traditional supervised, reinforcement, and imitation learning methods. Future work should explore the use of generative AI not only for data augmentation but also for creating adaptive and interactive decision support systems. This includes investigating large-scale pre-trained models, multimodal learning approaches that combine different types of inputs (such as text, sensor data, and images), and techniques for continual learning that allow models to evolve with new information over time. These advancements can make autonomous systems more flexible, responsive, and effective in dynamic combat environments.

- **Expansion of Aerospace Data Analysis Tools**: Dedicated analytics tools have played an important role in supporting simulation and machine learning workflows. Future efforts should focus on expanding these tools with additional analysis pipelines to support the evaluation of complex scenarios. Enhancements in data visualization, statistical processing, and predictive capabilities will be essential for improving decision-making and meeting evolving operational requirements.

15.3 Final Remarks

This thesis contributes to the field of decision support and autonomy in air combat operations by advancing the integration of simulation and machine learning. The technologies and methodologies developed throughout this work offer valuable capabilities to enhance the operational effectiveness of modern air forces. In particular, we hope these contributions will support more effective mission planning, execution, and autonomy within FAB.

Beyond its immediate applications, this research encourages the use of simulation and machine learning to improve military operations and aims to stimulate further studies in the area. The five focus areas proposed in this thesis represent an initial research framework that can be expanded and refined in future work.

A key challenge is gaining the trust of military decision-makers in the use of simulation and machine learning for operational purposes. Although these technologies have shown promising results, some concerns may still exist regarding their role in real-world missions.

Continued research efforts like the one presented in this thesis are essential to demonstrate their reliability and effectiveness, helping to strengthen their use in modern air forces.

Operational expertise also plays a key role in shaping and applying these technologies. Collaboration between military professionals and technical experts is necessary to develop accurate and relevant models to meet real-world needs. This interdisciplinary approach ensures that decision support systems and autonomous agents align well with operational requirements. Moreover, there is a growing need to train skilled personnel capable of developing, operating, and interpreting these systems. Building teams with expertise in both the technological and operational domains is essential to ensure their successful deployment and practical use. These professionals will help translate data-driven insights into practical strategies, improving decision-making across all levels.

The importance of increasing research into human interaction with autonomous agents is evident, as this is already becoming a key element of modern military operations. Specialized units have been established in other countries to manage simulation and machine learning applications. In this context, a modern air force should anticipate this need by forming dedicated teams to support commanders with advanced technologies, enabling faster and more informed decisions.

The research framework presented in this thesis aims to reinforce the importance of simulation and machine learning in developing a modern air force. It is expected to support future advancements in this area and help consolidate the role of these technologies in national security. Such developments may also contribute to global peacekeeping efforts by adopting reliable and effective digital tools and intelligent systems.

I hope this thesis contributes to further progress in Brazil's technological defense sector and highlights the importance of simulation and machine learning in operational applications to strengthen our sovereignty.

Bibliography

ABADI, M.; BARHAM, P.; CHEN, J.; CHEN, Z.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; IRVING, G.; ISARD, M. *et al.* TensorFlow: A System for Large-Scale Machine Learning. *In*: **12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)**. **Proceedings** [...]. Savannah, GA, USA: USENIX Association, 2016. p. 265–283.

ABDELLAOUI, N.; TAYLOR, A.; PARKINSON, G. **Comparative Analysis of Computer Generated Forces' Artificial Intelligence**. Ottawa, ON, Canada, 2009.

AHA, D. W. Goal Reasoning: Foundations, Emerging Applications, and Prospects. **AI Magazine**, American Association for Artificial Intelligence, USA, v. 39, n. 2, p. 3–24, jun 2018. ISSN 0738-4602. Available at: https://doi.org/10.1609/aimag.v39i2.2800.

AIELLO, J. R. A further look at equilibrium theory: Visual interaction as a function of interpersonal distance. **Environmental Psychology and Nonverbal Behavior**, Springer, v. 1, n. 2, p. 122–140, 1977.

AIR LAND SEA APPLICATION CENTER. **Brevity: Multi-Service Tactics, Techniques, and Procedures for Multi-Service Brevity Codes**. Hampton, VA, USA, 2020. Available at: https://armypubs.army.mil/epubs/DR_pubs/DR_a/ARN37654-ATP_1-02.1-000-WEB-1.pdf.

AKABARI, S.; MENHAJ, M. B.; NIKRAVESH, S. K. Fuzzy modeling of offensive maneuvers in an air-to-air combat. *In*: REUSCH, B. (Ed.). **Computational Intelligence, Theory and Applications**. **Proceedings** [...]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 171–184. ISBN 978-3-540-31182-9.

AKHGAR, B.; SAATHOFF, G. B.; ARABNIA, H. R.; HILL, R.; STANIFORTH, A.; BAYERL, P. S. **Application of Big Data for National Security: A Practitioner's Guide to Emerging Technologies**. Waltham, MA, USA: Butterworth-Heinemann, 2015.

ALBARADO, K.; CODUTI, L.; ALOISIO, D.; ROBINSON, S.; DROWN, D.; JAVORSEK, D. AlphaMosaic: An Artificially Intelligent Battle Management Architecture. **Journal of Aerospace Information Systems**, v. 19, n. 3, p. 203–213, 2022. Available at: https://doi.org/10.2514/1.I010991.

ALBERTS, D. S.; GARSTKA, J.; STEIN, F. P. *et al.* **Network Centric Warfare: Developing and Leveraging Information Superiority**. Washington, USA: National Defense University Press, 1999.

ALFORD, R.; BORCK, H.; KARNEEB, J.; AHA, D. W. **Active Behavior Recognition in Beyond Visual Range Air Combat**. Washington, DC, 2015. Technical Report ADA626583. Available at: https://apps.dtic.mil/sti/citations/ADA626583.

ALIN, A. Multicollinearity. **WIREs Computational Statistics**, v. 2, n. 3, p. 370–374, 2010. Available at: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.84.

ALKAHER, D.; MOSHAIOV, A. Dynamic-Escape-Zone to Avoid Energy-Bleeding Coasting Missile. **Journal of Guidance, Control, and Dynamics**, American Institute of Aeronautics and Astronautics, v. 38, n. 10, p. 1908–1921, 2015.

ALKAHER, D.; MOSHAIOV, A. Game-Based Safe Aircraft Navigation in the Presence of Energy-Bleeding Coasting Missile. **Journal of Guidance, Control, and Dynamics**, v. 39, n. 7, p. 1539–1550, 2016. Available at: https://doi.org/10.2514/1.G001676.

ALTHAUS, P.; ISHIGURO, H.; KANDA, T.; MIYASHITA, T.; CHRISTENSEN, H. Navigation for Human-Robot Interaction Tasks. *In*: **IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004**. **Proceedings** [...]. New Orleans, LA: IEEE, 2004. v. 2, p. 1894–1900 Vol.2.

ANDRYCHOWICZ, M.; BAKER, B.; CHOCIEJ, M.; JOZEFOWICZ, R.; MCGREW, B.; PACHOCKI, J.; PETRON, A.; PLAPPERT, M.; POWELL, G.; RAY, A.; SCHNEIDER, J.; TOBIN, J.; WELINDER, P.; WENG, L.; ZAREMBA, W. Learning Dexterous In-Hand Manipulation. **The International Journal of Robotics Research**, SAGE Publications, v. 39, n. 1, p. 3–20, 2020.

ANGHINOLFI, D.; CAPOGROSSO, A.; PAOLUCCI, M.; PERRA, F. An agent-based simulator for the evaluation of the measurement of effectiveness in the military naval tasks. *In*: **2013 17th International Conference on System Theory, Control and Computing (ICSTCC)**. **Proceedings** [...]. Sinaia, Romania: IEEE, 2013. p. 733–738.

ARLOT, S.; CELISSE, A. A Survey of Cross Validation Procedures for Model Selection. **Statistics Surveys**, v. 4, p. 40–79, 2010.

ARONSSON, S.; ARTMAN, H.; LINDQUIST, S.; MITCHELL, M.; PERSSON, T.; RAMBERG, R.; ROMERO, M.; VEHN, P. T. Supporting after action review in simulator mission training: Co-creating visualization concepts for training of fast-jet fighter pilots. **The Journal of Defense Modeling and Simulation**, Sage Publications Sage UK: London, England, v. 16, n. 3, p. 219–231, 2019.

ARONSSON, S.; ARTMAN, H.; MITCHELL, M.; RAMBERG, R.; WOLTJER, R. A live mindset in Live Virtual Constructive simulations: a spin-up for future LVC air combat training. **The Journal of Defense Modeling and Simulation**, v. 20, n. 4, p. 447–465, publisher: SAGE Publications, October 2023. ISSN 1548-5129. Available at: https://doi.org/10.1177/15485129221106204.

BAE, J. W.; KIM, J. H.; MOON, I.; KIM, T. G. Accelerated simulation of hierarchical military operations with tabulation technique. **Journal of Simulation**, Taylor & Francis, v. 10, n. 1, p. 36–49, 2016.

BAKIRCI, M.; OZER, M. M. Adapting Swarm Intelligence to a Fixed Wing Unmanned Combat Aerial Vehicle Platform. *In*: RIVERA, G.; CRUZ-REYES, L.; DORRONSORO, B.; ROSETE, A. (Ed.). **Data Analytics and Computational Intelligence: Novel Models, Algorithms and Applications**. Cham: Springer Nature Switzerland, 2023, (Studies in Big Data). p. 433–479. ISBN 978-3-031-38325-0. Available at: https://doi.org/10.1007/978-3-031-38325-0_18.

BALCI, O. Verification, Validation, and Testing. **Handbook of Simulation**, John Wiley & Sons, v. 10, n. 8, p. 335–393, 1998.

BALDASSARE, M. Human Spatial Behavior. **Annual Review of Sociology**, Annual Reviews, v. 4, p. 29–56, 1978. ISSN 03600572, 15452115. Available at: http://www.jstor.org/stable/2945964.

BALL, R. E. **The Fundamentals of Aircraft Combat Survivability Analysis and Design**. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2003.

BAOMAR, H.; BENTLEY, P. J. An Intelligent Autopilot System that Learns Piloting Skills from Human Pilots by Imitation. *In*: **2016 International Conference on Unmanned Aircraft Systems (ICUAS)**. **Proceedings** [...]. Arlington, VA, USA: IEEE, 2016. p. 1023–1031.

BARNES, M. J.; CHEN, J. Y.; JENTSCH, F.; ORON-GILAD, T.; REDDEN, E.; ELLIOTT, L.; III, A. W. E. Designing for Humans in Autonomous Systems: Military Applications. **DTIC Document, January**, 2014.

BATISTA, G. E. A. P. A.; PRATI, R. C.; MONARD, M. C. A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. **SIGKDD Explorations Newsletter**, ACM, New York, NY, USA, v. 6, n. 1, p. 20–29, 2004. ISSN 1931-0145. Available at: https://doi.org/10.1145/1007730.1007735.

BATTITI, R. Using Mutual Information for Selecting Features in Supervised Neural Net Learning. **IEEE Transactions on Neural Networks**, v. 5, n. 4, p. 537–550, 1994.

BATTLESPACE SIMULATIONS. **MACE: Modern Air Combat Environment**. San Antonio, Texas, 2024. Available at: https://www.bssim.com/products/mace.

BENGIO, Y. Practical recommendations for gradient-based training of deep architectures. *In*: **Neural Networks: Tricks of the Trade**. Berlin, Germany: Springer, 2012. p. 437–478.

BENNETT, B.; HOLT, C.; ELLIS, C.; HEMMINGS, P.; HAMILTON, B. A. Global broadcast service DVB-T tactical broadcast architecture. *In*: **MILCOM 2005 - 2005 IEEE Military Communications Conference**. **Proceedings** [...]. Atlantic City, NJ, USA: IEEE, 2005. p. 120–127 Vol. 1.

BERGSTRA, J.; BARDENET, R.; BENGIO, Y.; KEGL, B. Algorithms for Hyper-Parameter Optimization. *In*: **Advances in Neural Information Processing Systems**. **Proceedings** [...]. Granada, Spain: Curran Associates, Inc., 2011. p. 2546–2554.

BERGSTRA, J.; BENGIO, Y. Random Search for Hyper-Parameter Optimization. **Journal of Machine Learning Research**, v. 13, n. 10, p. 281–305, 2012. Available at: http://jmlr.org/papers/v13/bergstra12a.html.

BHARGAVE, A.; AMBROSE, B.; LIN, F.; KAZANTZIDIS, M. Multi-sensor detection and fusion technique. *In*: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **Proceedings of SPIE**. **Proceedings** [...]. Bellingham, WA, USA: SPIE, 2007. p. 123–134. Available at: https://dx.doi.org/10.1117/12.719505.

BIDDLE, S. Victory Misunderstood: What the Gulf War Tells Us about the Future of Conflict. **International Security**, The MIT Press, v. 21, n. 2, p. 139–179, 1996.

BIDDLE, S. **Military Power: Explaining Victory and Defeat in Modern Battle**. Princeton: Princeton University Press, 2004. ISBN 9781400837823. Available at: https://doi.org/10.1515/9781400837823.

BIRKELAND, J. O. The Concept of Autonomy and the Changing Character of War. **Oslo Law Review**, v. 5, n. 2, p. 73–88, 2018.

BIRKMIRE, B.; GALLAGHER, J. Air-to-air missile maximum launch range modeling using a multilayer perceptron. *In*: **AIAA Modeling and Simulation Technologies Conference**. **Proceedings** [...]. Minneapolis, USA: American Institute of Aeronautics and Astronautics (AIAA), 2012. p. 4942.

BIRKMIRE, B. M. **Weapon Engagement Zone Maximum Launch Range Approximation using a Multilayer Perceptron**. Dissertation (Master's) — Wright State University, 2011.

BOCK, S.; WEIß, M. A Proof of Local Convergence for the Adam Optimizer. *In*: **2019 International Joint Conference on Neural Networks (IJCNN)**. **Proceedings** [...]. Budapest, Hungary: IEEE, 2019. p. 1–8.

BOHEMIA INTERACTIVE. **VBS4**. 2023. `https://vbs4.com/`. Last accessed 15 April 2023.

BONACCORSO, G. **Machine Learning Algorithms**. Birmingham, UK: Packt Publishing Ltd, 2017.

BOND-TAYLOR, S.; LEACH, A.; LONG, Y.; WILLCOCKS, C. G. Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE, v. 44, n. 11, p. 7327–7347, 2022.

BORCK, H.; KARNEEB, J.; ALFORD, R.; AHA, D. W. **Case-Based Behavior Recognition to Facilitate Planning in Unmanned Air Vehicles**. Springfield, VA, 2014. Technical Report ADA621410. Available at: https://apps.dtic.mil/sti/citations/ADA621410.

BORCK, H.; KARNEEB, J.; ALFORD, R.; AHA, D. W. Case-Based Behavior Recognition in Beyond Visual Range Air Combat. *In*: **Proceedings of the 28$^{\text{th}}$ International FLAIRS Conference**. **Proceedings** [...]. Hollywood, FL, USA, May 18$^{\text{th}}$-25$^{\text{th}}$: AAAI Press, 2015. p. 379–384.

BORCK, H.; KARNEEB, J.; FLOYD, M. W.; ALFORD, R.; AHA, D. W. Case-Based Policy and Goal Recognition. *In*: HÜLLERMEIER, E.; MINOR, M. (Ed.). **Case-Based Reasoning Research and Development**. **Proceedings** [...]. Cham: Springer International Publishing, 2015. (Lecture Notes in Computer Science), p. 30–43. ISBN 978-3-319-24586-7.

BOX, G.; HUNTER, J.; HUNTER, W. **Statistics for Experimenters: Design, Innovation, and Discovery**. Hoboken, NJ, USA: Wiley, 2005. (Wiley Series in Probability and Statistics). ISBN 9780471718130.

BRADLEY, E. Advanced Medium Range Air-to-Air Missile (AMRAAM) System Overview. *In*: AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS. **25th AIAA Aerospace Sciences Meeting**. **Proceedings** [...]. Reno, NV, USA: AIAA, 1987. p. 462.

BRAUNLINGER, T. K. **Network-Centric Warfare Implementation and Assessment**. Dissertation (Master's) — US Army Command and General Staff College, Fort Leavenworth, KS, USA, 2005.

BROCKMAN, G.; CHEUNG, V.; PETTERSSON, L.; SCHNEIDER, J.; SCHULMAN, J.; TANG, J.; ZAREMBA, W. **OpenAI Gym**. 2016. Available at: https://arxiv.org/abs/1606.01540.

BROWN, C. E. Coefficient of variation. *In*: **Applied Multivariate Statistics in Geohydrology and Related Sciences**. Berlin, Heidelberg: Springer, 1998. p. 155–157.

BRUZZONE, A. G.; MASSEI, M. Simulation-Based Military Training. *In*: **Guide to Simulation-Based Disciplines**. Cham, Switzerland: Springer, 2017. p. 315–361.

CAI, G.; CHEN, B. M.; LEE, T. H. Coordinate systems and transformations. *In*: **Unmanned Rotorcraft Systems**. Berlin, Heidelberg, Germany: Springer, 2011. p. 23–34.

CAO, Y.; KOU, Y.; XU, A.; XI, Z. Target Threat Assessment in Air Combat Based on Improved Glowworm Swarm Optimization and ELM Neural Network. **International Journal of Aerospace Engineering**, v. 2021, p. e4687167, publisher: Hindawi, October 2021. ISSN 1687-5966. Available at: https://www.hindawi.com/journals/ijae/2021/4687167/.

CHAO, J.; PIOTROWSKI, W.; MANZANARES, M.; LANGE, D. Top Gun: Cooperative Multi-Agent Planning. *In*: **Proceedings of the AAAI MAPF Workshop**. **Proceedings** [...]. Washington, DC: AAAI, 2023.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O.; KEGELMEYER, W. P. SMOTE: Synthetic Minority Over-sampling Technique. **Journal of Artificial Intelligence Research**, AI Access Foundation, El Segundo, CA, USA, v. 16, n. 1, p. 321–357, jun 2002. ISSN 1076-9757.

CHEN, C.; MO, L.; LV, M.; LIN, D.; SONG, T.; CAO, J. Enhanced Missile Hit Probability Actor-Critic Algorithm for Autonomous Decision-Making in Air-to-Air Confrontation. **Aerospace Science and Technology**, Elsevier, p. 109285, 2024.

CHEN, L.; LU, K.; RAJESWARAN, A.; LEE, K.; GROVER, A.; LASKIN, M.; ABBEEL, P.; SRINIVAS, A.; MORDATCH, I. Decision Transformer: Reinforcement Learning via Sequence Modeling. *In*: **Proceedings of the 35th International Conference on Neural Information Processing Systems**. **Proceedings** [...]. Red Hook, NY, USA: Curran Associates Inc., 2021. (NIPS '21). ISBN 9781713845393.

CHEN, R.; LI, H.; YAN, G.; WANG, Z.; PENG, H. Target Intent Recognition Method Based on Evidence Fusion in TimeSeries Networks. *In*: **2022 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)**. **Proceedings** [...]. Xi'an, Shaanxi, China: IEEE, 2022. p. 1–6. Available at: https://ieeexplore.ieee.org/document/9984289.

CHEN, S.; CAO, Y.; KANG, Y.; LI, P.; SUN, B. Deep Feature Representation Based Imitation Learning for Autonomous Helicopter Aerobatics. **IEEE Transactions on Artificial Intelligence**, v. 2, n. 5, p. 437–446, 2021.

CHEN, T.; GUESTRIN, C. XGBoost: A Scalable Tree Boosting System. *In*: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. **Proceedings** [...]. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 785–794. ISBN 9781450342322. Available at: https://doi.org/10.1145/2939672.2939785.

CHO, K.; MERRIëNBOER, B. van; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *In*: MOSCHITTI, A.; PANG, B.; DAELEMANS, W. (Ed.). **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. **Proceedings** [...]. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1724–1734. Available at: https://aclanthology.org/D14-1179.

CHRISTENSEN, C.; SALMON, J. An agent-based modeling approach for simulating the impact of small unmanned aircraft systems on future battlefields. **The Journal of Defense Modeling and Simulation**, SAGE Publications Sage UK: London, England, v. 19, n. 3, p. 481–500, 2022.

CIOPPA, T. M.; LUCAS, T. W. Efficient Nearly Orthogonal and Space-Filling Latin Hypercubes. **Technometrics**, Taylor & Francis, v. 49, n. 1, p. 45–55, 2007.

CLIVE, P. D.; JOHNSON, J. A.; MOSS, M. J.; ZEH, J. M.; BIRKMIRE, B. M.; HODSON, D. D. Advanced Framework for Simulation, Integration and Modeling (AFSIM). *In*: THE STEERING COMMITTEE OF THE WORLD CONGRESS IN COMPUTER SCIENCE, COMPUTER ENGINEERING AND APPLIED COMPUTING (WORLDCOMP). **The 13th International Conference on Scientific Computing (CSC'15)**. **Proceedings** [...]. Las Vegas, NV, USA, 2015. p. 73.

COHEN, J. **Statistical Power Analysis for the Behavioral Sciences**. 2nd. ed. New York: Routledge, 1988. ISBN 978-0805802832.

COHEN, J.; COHEN, P.; WEST, S. G.; AIKEN, L. S. **Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences**. New York, NY, USA: Routledge, 2013.

COLLEDANCHISE, M.; ÖGREN, P. **Behavior Trees in Robotics and AI: An Introduction**. Boca Raton, FL: CRC Press, 2020. (Chapman & Hall / CRC Artificial Intelligence and Robotics Series). ISBN 9780367571337.

COSTA, A. N. **Sequential Optimization of Formation Flight Control Method Based on Artificial Potential Fields**. Dissertation (Master's) — Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil, 2019.

COSTA, A. N.; DANTAS, J. P. A.; SCUKINS, E.; MEDEIROS, F. L. L.; ÖGREN, P. Simulation and Machine Learning in Beyond Visual Range Air Combat: A Survey. **IEEE Access**, v. 13, p. 76755–76774, 2025.

COSTA, A. N.; MEDEIROS, F. L.; DANTAS, J. P. A.; GERALDO, D.; SOMA, N. Y. Formation control method based on artificial potential fields for aircraft flight simulation. **SIMULATION**, v. 98, n. 7, p. 575–595, 2022. Available at: https://doi.org/10.1177/00375497211063380.

DAHLBOM, A. A comparison of two approaches for situation detection in an air-to-air combat scenario. *In*: SPRINGER. **Modeling Decisions for Artificial Intelligence: 10th International Conference, MDAI 2013**. **Proceedings** [...]. Barcelona, Spain: Springer, 2013. p. 70–81.

DANTAS, J. P.; COSTA, A. N.; GERALDO, D.; MAXIMO, M. R.; YONEYAMA, T. PoKER: a probability of kill estimation rate model for air-to-air missiles using machine learning on stochastic targets. **The Journal of Defense Modeling and Simulation**, v. 0, n. 0, p. 15485129241309675, 2025. Available at: https://doi.org/10.1177/15485129241309675.

DANTAS, J. P. A. **Apoio à Decisão para o Combate Aéreo Além do Alcance Visual: Uma Abordagem por Redes Neurais Artificiais.** Dissertation (Master's) — Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil, 2018.

DANTAS, J. P. A. Autonomous Pop-Up Attack Maneuver in Air Combat Using Imitation and Generative Learning. *In*: **Proceedings of the Winter Simulation Conference, PhD Colloquium**. **Proceedings** [...]. Orlando, FL, USA: IEEE, 2024.

DANTAS, J. P. A.; COSTA, A. N.; GERALDO, D.; MAXIMO, M. R. O. A.; YONEYAMA, T. Engagement Decision Support for Beyond Visual Range Air Combat. *In*: **Proceedings of the 2021 Latin American Robotics Symposium, 2021 Brazilian Symposium on Robotics, and 2021 Workshop on Robotics in Education**. **Proceedings** [...]. Natal, RN, Brazil: IEEE, 2021. p. 96–101.

DANTAS, J. P. A.; COSTA, A. N.; GERALDO, D.; MAXIMO, M. R. O. A.; YONEYAMA, T. Weapon Engagement Zone Maximum Launch Range Estimation Using a Deep Neural Network. *In*: BRITTO, A.; DELGADO, K. V. (Ed.). **Intelligent Systems**. **Proceedings** [...]. Cham: Springer, 2021. p. 193–207. ISBN 978-3-030-91699-2.

DANTAS, J. P. A.; COSTA, A. N.; GOMES, V. C. F.; KUROWSKI, A. R.; MEDEIROS, F. L. L.; GERALDO, D. **ASA: A Simulation Environment for Evaluating Military Operational Scenarios**. arXiv.org, July 2022. Available at: https://arxiv.org/abs/2207.12084.

DANTAS, J. P. A.; COSTA, A. N.; MEDEIROS, F. L. L.; GERALDO, D.; MAXIMO, M. R. O. A.; YONEYAMA, T. Supervised Machine Learning for Effective Missile Launch Based on Beyond Visual Range Air Combat Simulations. *In*: **Proceedings of the Winter Simulation Conference**. **Proceedings** [...]. Singapore: IEEE, 2022. (WSC '22).

DANTAS, J. P. A.; GERALDO, D.; COSTA, A. N.; MAXIMO, M. R. O. A.; YONEYAMA, T. ASA-SimaaS: Advancing Digital Transformation through Simulation Services in the Brazilian Air Force. *In*: **Simpósio de Aplicações Operacionais em Áreas de Defesa (SIGE2023)**. **Proceedings** [...]. São José dos Campos, Brazil: Instituto Tecnológico de Aeronáutica (ITA), 2023. p. 6. ISSN 1983 7402. Available at: https://www.sige.ita.br/edicoes-anteriores/2023/st/235455_1.pdf.

DANTAS, J. P. A.; GERALDO, D.; MEDEIROS, F. L. L.; MAXIMO, M. R. O. A.; YONEYAMA, T. Real-Time Surface-to-Air Missile Engagement Zone Prediction Using Simulation and Machine Learning. *In*: **Proceedings of the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC)**. **Proceedings** [...]. Orlando, FL, USA: National Training and Simulation Association (NTSA), 2023.

DANTAS, J. P. A.; MAXIMO, M. R. O. A.; COSTA, A. N.; GERALDO, D.; YONEYAMA, T. Machine Learning to Improve Situational Awareness in Beyond Visual Range Air Combat. **IEEE Latin America Transactions**, v. 20, n. 8, 2022. Available at: https://latamt.ieeer9.org/index.php/transactions/article/view/6530.

DANTAS, J. P. A.; MAXIMO, M. R. O. A.; YONEYAMA, T. Autonomous Agent for Beyond Visual Range Air Combat: A Deep Reinforcement Learning Approach. *In*: **Proceedings of the 2023 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation (SIGSIM-PADS'23)**. **Proceedings** [...]. New York, NY, USA: Association for Computing Machinery, 2023. (SIGSIM-PADS'23), p. 13–24. ISBN 979-8-4007-0030-0. Available at: https://doi.org/10.1145/3573900.3593631.

DANTAS, J. P. A.; MAXIMO, M. R. O. A.; YONEYAMA, T. Loyal Wingman Assessment: Social Navigation for Human-Autonomous Collaboration in Simulated Air Combat. *In*: **Proceedings of the 38th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation**. **Proceedings** [...]. New York, NY, USA: Association for Computing Machinery, 2024. (SIGSIM-PADS '24), p. 61–62. ISBN 9798400703638. Available at: https://doi.org/10.1145/3615979.3662149.

DANTAS, J. P. A.; MAXIMO, M. R. O. A.; YONEYAMA, T. Autonomous Aircraft Tactical Pop-Up Attack Using Imitation and Generative Learning. **IEEE Access**, v. 13, p. 81204–81217, 2025.

DANTAS, J. P. A.; SILVA, S. R.; GOMES, V. C. F.; COSTA, A. N.; SAMERSLA, A. R.; GERALDO, D.; MAXIMO, M. R. O. A.; YONEYAMA, T. AsaPy: A Python Library for Aerospace Simulation Analysis. *In*: **Proceedings of the 38th ACM SIGSIM Conference on Principles of Advanced Discrete Simulation**. **Proceedings** [...]. New York, NY, USA: Association for Computing Machinery, 2024. (SIGSIM-PADS '24), p. 15–24. ISBN 9798400703638. Available at: https://doi.org/10.1145/3615979.3656063.

DAVIS, P. K.; BRACKEN, P. Artificial Intelligence for Wargaming and Modeling. **The Journal of Defense Modeling and Simulation**, SAGE Publications Sage UK: London, England, p. 15485129211073126, 2022.

DAVIS, P. K.; HENNINGER, A. E. **Analysis, Analysis Practices, and Implications for Modeling and Simulation**. Santa Monica, CA, USA: RAND Corporation, 2007.

DEB, K.; PRATAP, A.; AGARWAL, S.; MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 2, p. 182–197, 2002.

DEPARTMENT OF DEFENSE. **Military Handbook: Missile Flight Simulation Part One: Surface-to-Air Missiles (MIL-HDBK-1211)**. Washington, D.C., USA, 1995. Available at: https://quicksearch.dla.mil/qsDocDetails.aspx?ident_number=116871.

DEPARTMENT OF DEFENSE. **DoD Dictionary of Military and Associated Terms**. Washington, USA, 2021.

DEUTSCH, J. L.; DEUTSCH, C. V. Latin Hypercube Sampling with Multidimensional Uniformity. **Journal of Statistical Planning and Inference**, Elsevier, v. 142, n. 3, p. 763–772, 2012.

DIETL, T.; PRAKASHA, P. S.; SCHMITZ, M.; ZILL, T.; NAGEL, B.; PINSON, N. Fighter Design and Fleet Effectiveness Evaluation via System of Systems Battlespace Simulation. *In*: **AIAA AVIATION 2023 Forum**. American Institute of Aeronautics and Astronautics, 2023. Eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2023-3519. Available at: https://arc.aiaa.org/doi/abs/10.2514/6.2023-3519.

DIGINEXT. **DirectCGF**. 2023. `https://www.directcgf.com/`. Last accessed 15 April 2023.

DING, Y.; YANG, L.; HOU, J.; JIN, G.; ZHEN, Z. **Multi-target Collaborative Combat Decision-Making by Improved Particle Swarm Optimizer**. February 2018. ISSN: 1005-1120. Publisher: Transactions of Nanjing University of Aeronautics and Astronautics. Available at: https://tnuaa.nuaa.edu.cn/html/2018/1/E201801020.htm.

DING, Y.; YANG, Z.; PHAM, Q.; HU, Y.; ZHANG, Z.; SHIKH-BAHAEI, M. Distributed Machine Learning for UAV Swarms: Computing, Sensing, and Semantics. **IEEE Internet of Things Journal**, IEEE, 2023.

DOANE, D. P. Aesthetic Frequency Classifications. **The American Statistician**, Taylor & Francis, v. 30, n. 4, p. 181–183, 1976.

DOERRY, A. W. **Radar Operation in a Hostile Electromagnetic Environment**. Albuquerque, NM, USA, 2014.

DOSHI, F.; LESSEM, R.; MOONEY, D. The Safe Distance Between Airplanes and the Complexity of an Airspace Sector. **The UMAP Journal**, v. 21, n. 3, p. 257–67, 2000.

DOU, X.; TANG, G.; ZHENG, A.; WANG, H.; LIANG, X. Research on Autonomous Decision-Making in Manned/Unmanned Coordinated Air Combat. *In*: **Proceedings of the 9th International Conference on Control, Automation and Robotics (ICCAR)**. **Proceedings** [...]. Beijing, China: IEEE, 2023. p. 1–6. Available at: https://dx.doi.org/10.1109/ICCAR57134.2023.10151765.

DROZD, J.; RAK, L.; ZAHRADNÍČEK, P.; STODOLA, P.; HODICKỲ, J. Effectiveness evaluation of aerial reconnaissance in battalion force protection operation using the constructive simulation. **The Journal of Defense Modeling and Simulation**, SAGE Publications Sage UK: London, England, v. 20, n. 2, p. 181–196, 2023.

DU, P.; LIU, H. Study on Air Combat Tactics Decision-Making Based on Bayesian Networks. *In*: IEEE. **2010 2nd IEEE International Conference on Information Management and Engineering**. **Proceedings** [...]. Chengdu, China: IEEE, 2010. p. 252–256.

EAGLE DYNAMICS. **DCS World: Digital Combat Simulator**. Moscow, Russia, 2024. Available at: https://www.digitalcombatsimulator.com.

ENDSLEY, M. R. Toward a Theory of Situation Awareness in Dynamic Systems. **Human Factors: The Journal of the Human Factors and Ergonomics Society**, v. 37, n. 1, p. 32–64, 1995. ISSN 0018-7208. Available at: http://journals.sagepub.com/doi/10.1518/001872095779049543.

ESIM GAMES. **Steel Beasts Pro**. 2023. `https://www.esimgames.com/`. Last accessed 15 April 2023.

EVANS, E. Radar Systems and Technologies for Navy Air and Missile Defense. *In*: **2000 IEEE International Radar Conference**. **Proceedings** [...]. Washington, DC: IEEE, 2000. p. 1–6. Available at: https://dx.doi.org/10.1109/RADAR.2000.851795.

EVENSEN, P. **Modelling and implementation of a generic active protection system for entities in Virtual Battlespace (VBS)**. Kjeller, Norway, 2017.

FAN, Z.; XU, Y.; KANG, Y.; LUO, D. Air Combat Maneuver Decision Method Based on A3C Deep Reinforcement Learning. **Machines**, MDPI, v. 10, n. 11, p. 1033, 2022.

FARLIK, J.; CASAR, J.; STARY, V. Simplification of missile effective coverage zone in air defence simulations. *In*: IEEE. **2017 International Conference on Military Technologies (ICMT)**. **Proceedings** [...]. Brno, Czech Republic: IEEE, 2017. p. 733–737.

FEDERAL AVIATION ADMINISTRATION. **Airplane Flying Handbook**. Oklahoma City, OK: United States Department of Transportation, Airman Testing Standards Branch, 2021.

FELDKAMP, N.; BERGMANN, S.; STRASSBURGER, S. Knowledge discovery in simulation data. **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, ACM New York, NY, USA, v. 30, n. 4, p. 1–25, 2020.

FLOYD, M. W.; KARNEEB, J.; AHA, D. W. Case-Based Team Recognition Using Learned Opponent Models. *In*: AHA, D. W.; LIEBER, J. (Ed.). **Case-Based Reasoning Research and Development**. **Proceedings** [...]. Cham: Springer International Publishing, 2017. (Lecture Notes in Computer Science), p. 123–138. ISBN 978-3-319-61030-6.

FLOYD, M. W.; KARNEEB, J.; MOORE, P.; AHA, D. W. A Goal Reasoning Agent for Controlling UAVs in Beyond-Visual-Range Air Combat. *In*: **Proceedings of the 26th International Joint Conference on Artificial Intelligence**. **Proceedings** [...]. Melbourne, Australia: AAAI Press, 2017. (IJCAI'17), p. 4714–4721. ISBN 9780999241103.

FOO, P. H.; NG, G. W.; NG, K. H.; YANG, R. Application of Intent Inference for Air Defense and Conformance Monitoring. **Journal of Advances in Information Fusion (JAIF)**, v. 4, n. 1, p. 3–26, 2009.

FRIEDMAN, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. **Journal of the American Statistical Association**, Taylor & Francis, v. 32, n. 200, p. 675–701, 1937. Available at: https://doi.org/10.1080/01621459.1937.10503522.

FU, L.; LONG, X.; HE, W. Air Combat Assignment Problem Based on Bayesian Optimization Algorithm. **Journal of Shanghai Jiaotong University (Science)**, Springer, p. 1–7, 2021.

FU, L.; XIE, F.; WANG, D.; MENG, G. The Overview for UAV Air-Combat Decision Method. *In*: **The 26th Chinese Control and Decision Conference (2014 CCDC)**. **Proceedings** [...]. Changsha, China: IEEE, 2014. p. 3380–3384.

FUJIMOTO, S.; HOOF, H. van; MEGER, D. Addressing function approximation error in actor-critic methods. *In*: **Proceedings of the 35th International Conference on Machine Learning**. **Proceedings** [...]. Stockholm, Sweden: PMLR, 2018. (Proceedings of Machine Learning Research, v. 80), p. 1587–1596.

GAO, W.; YANG, Z.; HUANG, J.; CHAI, S.; ZHOU, D. Design of Dynamic Tactical Control Range in Air Combat based on Three-Party Game. *In*: IEEE. **2023 35th Chinese Control and Decision Conference (CCDC)**. **Proceedings** [...]. Shenyang, China: IEEE, 2023. p. 3946–3951.

GAO, W.; YANG, Z.; SUN, Z.; PIAO, H.; HE, Y.; ZHOU, D. Real-time Calculation of Tactical Control Range in Beyond Visual Range Air Combat. *In*: IEEE. **2022 IEEE International Conference on Unmanned Systems (ICUS)**. **Proceedings** [...]. Guangzhou, China: IEEE, 2022. p. 76–80.

GAO, Y.; HUANG, C. Evaluation of Socially-Aware Robot Navigation. **Frontiers in Robotics and AI**, Frontiers, p. 420, 2021.

GARCIA, E.; TRAN, D. M.; CASBEER, D.; MILUTINOVIC, D.; PACHTER, M. Beyond visual range tactics. *In*: **AIAA Scitech 2021 Forum**. **Proceedings** [...]. Virtual Event: American Institute of Aeronautics and Astronautics (AIAA), 2021. p. 1229.

GERON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. Sebastopol, CA, USA: O'Reilly Media, Inc., 2019.

GIBBONS, I. F.; BOTHA, J. Tactical Radar Missile Challenges. *In*: IEEE. **2015 IEEE Radar Conference (RadarCon)**. **Proceedings** [...]. Arlington, VA, USA: IEEE, 2015. p. 563–568. Available at: https://dx.doi.org/10.1109/RadarConf.2015.7411852.

GILL, J.; GRIEGER, D.; WONG, J.; CHAU, M. Regression analysis, multiple comparisons and ranking sensitivity. *In*: **Proceedings of the 2018 Winter Simulation Conference (WSC)**. **Proceedings** [...]. Gothenburg, Sweden: IEEE, 2018. p. 3789–3800. Available at: https://www.informs-sim.org/wsc18papers/includes/files/338.pdf.

GLOZMAN, T.; NARKAWICZ, A.; KAMON, I.; CALLARI, F.; NAVOT, A. A Vision-based Solution to Estimating Time to Closest Point of Approach for Sense and Avoid. *In*: **Proceedings of the AIAA SciTech 2021 Forum**. **Proceedings** [...]. Virtual Event: American Institute of Aeronautics and Astronautics (AIAA), 2021. p. 0450.

GOBI, G. H.; BOTELHO, P. L. R. **AsaGym: a Python Library for Reinforcement Learning in Aerospace Operational Scenarios**. 2023. Undergraduate Thesis – Instituto Tecnológico de Aeronáutica, São José dos Campos.

GONG, X.; CHEN, W.; CHEN, Z. Singular-perturbation-based intelligent midcourse guidance for air-to-air missiles equipped with dual-pulse motor. *In*: **Proceedings of the 14th International Conference on Mechanical and Aerospace Engineering (ICMAE)**. **Proceedings** [...]. Porto, Portugal: IEEE, 2023. p. 419–424. Available at: https://doi.org/10.1109/ICMAE59650.2023.10424613.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, MA, USA: MIT Press, 2016. `http://www.deeplearningbook.org`.

GOOGLE. **Protocol Buffers Documentation**. 2023. Available at: https://protobuf.dev.

GORTON, P.; ASPRUSTEN, M.; BRATHEN, K. **Imitation Learning for Modelling Air Combat Behaviour — An Exploratory Study**. Kjeller, Norway, January 2023. Available at: https://ffi-publikasjoner.archive.knowledgearc.net/handle/20.500.12242/3136.

GORTON, P. R.; STRAND, A.; BRATHEN, K. **A survey of air combat behavior modeling using machine learning**. 2024. Available at: https://arxiv.org/abs/2404.13954.

GREENBERG, C. I.; STRUBE, M. J.; MYERS, R. A. A multitrait-multimethod investigation of interpersonal distance. **Journal of Nonverbal Behavior**, Springer, v. 5, n. 2, p. 104–114, 1980.

GUHL, J. Air Combat: An Improved Rafale. **Asia-Pacific Defence Reporter (2002)**, v. 39, n. 1, p. 64–67, 2013.

GUNSCH, G. H.; DYER, D. E.; GERKEN, M. J.; MERKLE, L. D.; WHELAN, M. A. Autonomous Agents as Air Combat Simulation Adversaries. *In*: **Applications of Artificial Intelligence 1993: Knowledge-Based Systems in Aerospace and Industry**. **Proceedings** [...]. Orlando, FL, USA: SPIE, 1993. p. 50–60. Available at: https://dx.doi.org/10.1117/12.141754.

HA, J.; CHAE, H.; CHOI, H. A Stochastic Game-Theoretic Approach for Analysis of Multiple Cooperative Air Combat. *In*: **2015 American Control Conference (ACC)**. **Proceedings** [...]. Chicago, IL, USA: IEEE, 2015. p. 3728–3733. ISSN: 2378-5861. Available at: https://ieeexplore.ieee.org/abstract/document/7171909.

HA, J.; CHAE, H.; CHOI, H. A Stochastic Game-Based Approach for Multiple Beyond-Visual-Range Air Combat. **Unmanned Systems**, World Scientific, v. 06, n. 01, p. 67–79, 2018. Available at: https://doi.org/10.1142/S2301385018500048.

HAARNOJA, T.; ZHOU, A.; ABBEEL, P.; LEVINE, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *In*: **Proceedings**

of the 35th International Conference on Machine Learning. **Proceedings** [...]. Stockholm, Sweden: PMLR, 2018. (Proceedings of Machine Learning Research, v. 80), p. 1861–1870.

HALL, E. **The Hidden Dimension**. Knopf Doubleday Publishing Group, 1990. ISBN 9780385084765. Available at: https://books.google.com.br/books?id=AB6zwwEACAAJ.

HAN, Y.; PIAO, H.; HOU, Y.; SUN, Y.; SUN, Z.; ZHOU, D.; YANG, S.; PENG, X.; FAN, S. Deep Relationship Graph Reinforcement Learning for Multi-Aircraft Air Combat. *In*: **2022 International Joint Conference on Neural Networks (IJCNN)**. **Proceedings** [...]. Padua, Italy: IEEE, 2022. p. 1–8. ISSN: 2161-4407. Available at: https://ieeexplore.ieee.org/abstract/document/9892208.

HANCOCK, P. A.; VINCENZI, D. A.; WISE, J. A.; MOULOUA, M. **Human factors in simulation and training**. Boca Raton, FL, USA: CRC Press, 2008.

HAOYU, L.; YUNFEI, Z.; SHIHAO, L. Simulation and Effectiveness Analysis on One versus One Beyond Visual Range Air Combat. **MATEC Web of Conferences**, v. 151, p. 05001, publisher: EDP Sciences, 2018. ISSN 2261-236X. Available at: https://www.matec-conferences.org/articles/matecconf/abs/2018/10-/matecconf_acmae2018_05001/matecconf_acmae2018_05001.html.

HARRIS, C. R.; MILLMAN, K. J.; WALT, S. J. van der; GOMMERS, R.; VIRTANEN, P.; COURNAPEAU, D.; WIESER, E.; TAYLOR, J.; BERG, S.; SMITH, N. J.; KERN, R.; PICUS, M.; HOYER, S.; KERKWIJK, M. H. van; BRETT, M.; HALDANE, A.; RÍO, J. F. del; WIEBE, M.; PETERSON, P.; GÉRARD-MARCHANT, P.; SHEPPARD, K.; REDDY, T.; WECKESSER, W.; ABBASI, H.; GOHLKE, C.; OLIPHANT, T. E. Array programming with NumPy. **Nature**, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, September 2020. Available at: https://doi.org/10.1038/s41586-020-2649-2.

HARTLEY, D. S. Verification & validation in military simulations. *In*: **Proceedings of the 29th Conference on Winter Simulation**. **Proceedings** [...]. USA: IEEE Computer Society, 1997. (WSC '97), p. 925–932. ISBN 078034278X. Available at: https://doi.org/10.1145/268437.268682.

HE, H.; BAI, Y.; GARCIA, E. A.; LI, S. ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. *In*: **IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)**. **Proceedings** [...]. Hong-Kong, China, June 1st-8th: IEEE, 2008. p. 1322–1328.

HE, H.; DONG, Q.; SHANG, X.; YANG, Y.; WEI, Q.; WANG, L. Autonomous Decision-Making Algorithm for Multi-Agent Beyond-Visual-Range Air Combat. *In*: LI, X.; SONG, X.; ZHOU, Y. (Ed.). **Proceedings of 2023 7th Chinese Conference on Swarm Intelligence and Cooperative Control**. **Proceedings** [...]. Singapore: Springer Nature, 2024. p. 646–660. Available at: https://doi.org/10.1007/978-981-97-3336-1_55.

HE, S.; GAO, Y.; ZHANG, B.; CHANG, H.; ZHANG, X. Advancing Air Combat Tactics with Improved Neural Fictitious Self-Play Reinforcement Learning. *In*: HUANG, D.-S.; PREMARATNE, P.; JIN, B.; QU, B.; JO, K.-H.; HUSSAIN, A. (Ed.).

Advanced Intelligent Computing Technology and Applications. **Proceedings** [...]. Singapore: Springer Nature, 2023. (Lecture Notes in Computer Science), p. 653–666. ISBN 978-981-9947-61-4.

HEINZE, C.; SMITH, B.; CROSS, M. Thinking quickly: Agents for modeling air warfare. *In*: ANTONIOU, G.; SLANEY, J. (Ed.). **Advanced Topics in Artificial Intelligence**. **Proceedings** [...]. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998. p. 47–58. ISBN 978-3-540-49561-1.

HELTON, J. C.; DAVIS, F. J. Latin Hypercube Sampling and the Propagation of Uncertainty in Analyses of Complex Systems. **Reliability Engineering & System Safety**, Elsevier, v. 81, n. 1, p. 23–69, 2003.

HENDERSON, P.; ISLAM, R.; BACHMAN, P.; PINEAU, J.; PRECUP, D.; MEGER, D. Deep Reinforcement Learning That Matters. *In*: **Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)**. **Proceedings** [...]. New Orleans, Louisiana, USA: AAAI Press, 2018. p. 3207–3214.

HERRMANN, J. Air-to-air missile engagement analysis using the USAF Trajectory Analysis Program (TRAP). *In*: **Proceedings of the AIAA Flight Simulation Technologies Conference**. **Proceedings** [...]. San Diego, CA, USA: American Institute of Aeronautics and Astronautics, 1996. p. 3489.

HIGBY, L. C. P.; COL, U. Promise and Reality: Beyond Visual Range (BVR) Air-to-Air Combat. **Air War College (AWC) Electives Program: Air Power Theory, Doctrine, and Strategy: 1945–Present**, v. 30, 2005.

HILL, R. R.; MILLER, J. O.; MCINTYRE, G. A. Applications of discrete event simulation modeling to military problems. *In*: IEEE. **Proceeding of the 2001 Winter Simulation Conference (Cat. No. 01CH37304)**. **Proceedings** [...]. Arlington, USA: IEEE, 2001. v. 1, p. 780–788.

HILL, R. R.; TOLK, A.; HODSON, D. D.; MILLAR, J. R. Open Challenges in Building Combat Simulation Systems to Support Test, Analysis and Training. *In*: **2018 Winter Simulation Conference (WSC)**. **Proceedings** [...]. Gothenburg, Sweden: Association for Computing Machinery, 2018. p. 3730–3741.

HINTON, G. E.; SRIVASTAVA, N.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. R. **Improving neural networks by preventing co-adaptation of feature detectors**. 2012. Available at: https://arxiv.org/abs/1207.0580.

HO, J.; ERMON, S. Generative adversarial imitation learning. **Advances in neural information processing systems**, v. 29, p. 4565–4573, 2016.

HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. **Neural Comput.**, MIT Press, Cambridge, MA, USA, v. 9, n. 8, p. 1735–1780, nov. 1997. ISSN 0899-7667. Available at: https://doi.org/10.1162/neco.1997.9.8.1735.

HODSON, D. D.; GEHL, D. P. The Mixed Reality Simulation Platform (MIXR). *In*: THE STEERING COMMITTEE OF THE WORLD CONGRESS IN COMPUTER SCIENCE, COMPUTER ENGINEERING AND APPLIED COMPUTING (WORLDCOMP). **The 16th International Conference on Scientific Computing (CSC'18)**. **Proceedings** [...]. Las Vegas, USA, 2018. p. 73.

HOMEM-DE-MELLO, T.; BAYRAKSAN, G. Monte Carlo Sampling-Based Methods for Stochastic Optimization. **Surveys in Operations Research and Management Science**, Elsevier, v. 19, n. 1, p. 56–85, 2014.

HOORN, M. V. **Optimizing Air-to-Air Missile Guidance using Reinforcement Learning**. Dissertation (Master's) — Delft University of Technology, March 2019. Available at: `http://resolver.tudelft.nl/uuid:6b26d24a-780c-47ee-9456-7af07490a317`.

HORNE, G.; SEICHTER, S. Data Farming in Support of NATO Operations - Methodology and Proof-of-Concept. *In*: IEEE. **Proceedings of the Winter Simulation Conference 2014**. **Proceedings** [...]. Savannah, GA, USA: IEEE, 2014. p. 2355–2363.

HOTA, H. S.; HANDA, R.; SHRIVAS, A. K. Time Series Data Prediction Using Sliding Window Based RBF Neural Network. **International Journal of Computational Intelligence Research**, Research India Publications, v. 13, n. 5, p. 1145–1156, 2017. ISSN 0973-1873. Available at: https://www.ripublication.com/ijcir17/ijcirv13n5_46.pdf.

HU, D.; YANG, R.; ZHANG, Y.; YUE, L.; YAN, M.; ZUO, J.; ZHAO, X. Aerial combat maneuvering policy learning based on confrontation demonstrations and dynamic quality replay. **Engineering Applications of Artificial Intelligence**, v. 111, p. 104767, May 2022. ISSN 0952-1976. Available at: https://www.sciencedirect.com/science/article/pii/S0952197622000586.

HU, D.; YANG, R.; ZUO, J.; ZHANG, Z.; WU, J.; WANG, Y. Application of Deep Reinforcement Learning in Maneuver Planning of Beyond-Visual-Range Air Combat. **IEEE Access**, v. 9, p. 32282–32297, 2021. ISSN 2169-3536. Available at: https://ieeexplore.ieee.org/abstract/document/9358136.

HU, J.; WANG, L.; HU, T.; GUO, C.; WANG, Y. Autonomous Maneuver Decision Making of Dual-UAV Cooperative Air Combat Based on Deep Reinforcement Learning. **Electronics**, v. 11, n. 3, p. 467, publisher: Multidisciplinary Digital Publishing Institute (MDPI), January 2022. ISSN 2079-9292. Available at: https://www.mdpi.com/2079-9292/11/3/467.

HUANG, H.; DING, L.; YANG, L.; HAN, B. Air Combat Effectiveness Evaluation for Fighter Based on Relevance Vector Machine. *In*: **2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)**. **Proceedings** [...]. Dalian, China: IEEE, 2020. p. 275–279. Available at: https://ieeexplore.ieee.org/abstract/document/9182585.

HUANG, H.; JIA, H.; YIN, H.; CHANG, S.; YANG, J. Analysis of Transmission and Application of UV Radiance of Missile Plume. *In*: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. **International Conference on Optical Instruments and Technology: Optoelectronic Imaging and Process Technology**. **Proceedings** [...]. Beijing, China: SPIE, 2012. p. 870–879. Available at: https://dx.doi.org/10.1117/12.977627.

HUI, Y.; NAN, Y.; CHEN, S.; DING, Q.; WU, S. Dynamic attack zone of air-to-air missile after being launched in random wind field. **Chinese Journal of Aeronautics**, Elsevier, v. 28, n. 5, p. 1519–1528, 2015.

HUSSEIN, A.; GABER, M. M.; ELYAN, E.; JAYNE, C. Imitation Learning: A Survey of Learning Methods. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 50, n. 2, p. 1–35, apr 2017. ISSN 0360-0300. Available at: https://doi.org/10.1145/3054912.

HUSSLAGE, B. G.; RENNEN, G.; DAM, E. R. V.; HERTOG, D. D. Space-Filling Latin Hypercube Designs For Computer Experiments. **Optimization and Engineering**, Springer, v. 12, p. 611–630, 2011.

HYNDMAN, R. J.; KOEHLER, A. B. Another look at measures of forecast accuracy. **International Journal of Forecasting**, Elsevier, v. 22, n. 4, p. 679–688, 2006.

IHAKA, R.; GENTLEMAN, R. R: A Language for Data Analysis and Graphics. **Journal of Computational and Graphical Statistics**, American Statistical Association, Taylor & Francis, Ltd., Institute of Mathematical Statistics, Interface Foundation of America, v. 5, n. 3, p. 299–314, 1996. ISSN 10618600. Available at: http://www.jstor.org/stable/1390807.

IHS Jane's. Journal, **White Paper Fast Jet Cost Per Flight Hour (CPFH)**. Coulsdon, Surrey, UK: IHS Global Ltd, 2012. 1–14 p.

ISBY, D. C. Enabling Defense Transformation: Network-Centric Warfare and Ballistic Missile Defense. **Defense & Security Analysis**, v. 19, n. 1, p. 33–46, 2003. Available at: https://dx.doi.org/10.1080/01495930390237104.

JAISWAL, N. K. **Military Operations Research: Quantitative Decision Making**. New York, USA: Springer Science & Business Media, 2012.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. **An Introduction to Statistical Learning**. New York, NY, USA: Springer, 2013.

JENI, L. A.; COHN, J. F.; TORRE, F. D. L. Facing Imbalanced Data–Recommendations for the Use of Performance Metrics. *In*: **Proceedings of the Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII)**. **Proceedings** [...]. Geneva, Switzerland, September 2nd-5th: IEEE, 2013. p. 245–251.

JIA, Y.; YANG, Z.; HE, Y.; WANG, X.; ZHOU, D. Dynamic effectiveness evaluation method for beyond-visual-range air-to-air missile after launch. *In*: **Proceedings of the 2023 IEEE International Conference on Unmanned Systems (ICUS)**. **Proceedings** [...]. Hefei, China: IEEE, 2023. p. 482–487. Available at: https://doi.org/10.1109/ICUS58632.2023.10318387.

JIA, Z.; KIANG, J. War Game between Two Matched Fleets with Goal Options and Tactical Optimization. **AI**, v. 3, n. 4, p. 890–930, publisher: Multidisciplinary Digital Publishing Institute (MDPI), December 2022. ISSN 2673-2688. Available at: https://www.mdpi.com/2673-2688/3/4/54.

JIANG, Y.; YU, J.; LI, Q. A novel decision-making algorithm for beyond visual range air combat based on deep reinforcement learning. *In*: **2022 37th Youth Academic Annual Conference of Chinese Association of Automation (YAC)**. **Proceedings** [...]. Kunming, China: IEEE, 2022. p. 516–521. Available at: https://ieeexplore.ieee.org/abstract/document/10023870.

JOLLIFFE, I. **Principal Component Analysis**. New York, NY, USA: Springer, 2002. (Springer Series in Statistics). ISBN 9780387954424.

JSBSIM DEVELOPMENT TEAM. **JSBSim: An Open Source Flight Dynamics Model**. Open Source Project, 2024. Version 1.2.1. Available at: http://jsbsim.sourceforge.net/.

JUNIOR, I. M.; ROCHA, A. V.; MOTA, E. B.; QUINTELLA, O. M. **Gestão da qualidade e processos**. Rio de Janeiro, Brazil: Editora FGV, 2021.

KÄLLSTRÖM, J.; HEINTZ, F. Agent coordination in air combat simulation using multi-agent deep reinforcement learning. *In*: IEEE. **2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)**. **Proceedings** [...]. Toronto, ON, Canada: IEEE, 2020. p. 2157–2164.

KANG, Y.; LIU, Z.; PU, Z.; YI, J.; ZU, W. Beyond-Visual-Range Tactical Game Strategy for Multiple UAVs. *In*: **2019 Chinese Automation Congress (CAC)**. **Proceedings** [...]. Hangzhou, China: IEEE, 2019. p. 5231–5236. ISSN: 2688-0938. Available at: https://ieeexplore.ieee.org/abstract/document/8996232.

KARELAHTI, J.; VIRTANEN, K.; RAIVIO, T. Game Optimal Support Time of a Medium Range Air-to-Air Missile. **Journal of Guidance, Control, and Dynamics**, v. 29, n. 5, p. 1061–1069, 2006. Available at: https://doi.org/10.2514/1.17910.

KARLE, D. D.; HALL, J. B. **Integrating Strategic and Tactical Airpower in Conventional Warfare: B-52 Employment**. Maxwell Air Force Base, Alabama, 1988. Available at: https://apps.dtic.mil/sti/pdfs/ADA202217.pdf.

KARNEEB, J.; FLOYD, M. W.; MOORE, P.; AHA, D. W. Distributed Discrepancy Detection for BVR Air Combat. *In*: **Proceedings of the Workshop on Goal Reasoning (held at the 25th International Joint Conference on Artificial Intelligence)**. **Proceedings** [...]. New York, NY, USA: IJCAI, 2016. Available at: https://www.ijcai.org/Proceedings/16/Papers/Workshop/GR16.pdf.

KATO, Y.; KANDA, T.; ISHIGURO, H. May I Help You? Design of Human-like Polite Approaching Behavior. *In*: **Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction**. **Proceedings** [...]. New York, USA: ACM, 2015. (HRI '15), p. 35–42. ISBN 9781450328838. Available at: https://doi.org/10.1145/2696454.2696463.

KATUKURI, M. Target-Interception Feasibility Assessment Methodology for Air-Launched Weapon Systems. *In*: IEEE. **2023 3rd International Conference on Range Technology (ICORT)**. **Proceedings** [...]. Chandipur, India: IEEE, 2023. p. 1–5.

KAZIL, J.; JARMUL, K. **Data Wrangling with Python: Tips and Tools to Make Your Life Easier**. O'Reilly Media, 2016. ISBN 9781491948774. Available at: https://books.google.com.br/books?id=OmeDCwAAQBAJ.

KHOSHGOFTAAR, T. M.; GOLAWALA, M.; HULSE, J. V. An Empirical Study of Learning from Imbalanced Data Using Random Forest. *In*: **19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007)**. **Proceedings** [...]. Patras, Greece, October 29th-31st: IEEE, 2007. v. 2, p. 310–317.

KINGMA, D.; BA, J. Adam: A Method for Stochastic Optimization. *In*: **International Conference on Learning Representations (ICLR)**. **Proceedings** [...]. San Diego, CA, USA: ICLR, 2014.

KINGMA, D. P.; WELLING, M. Auto-Encoding Variational Bayes. **arXiv preprint arXiv:1312.6114**, 2013.

KLEE, H. **Simulation of Dynamic Systems with MATLAB and Simulink**. Boca Raton, USA: CRC Press, 2018.

KLEIJNEN, J. P. C.; SANCHEZ, S. M.; LUCAS, T. W.; CIOPPA, T. M. State-of-the-Art Review: A User's Guide to the Brave New World of Designing Simulation Experiments. **INFORMS J. on Computing**, INFORMS, Linthicum, MD, USA, v. 17, n. 3, p. 263–289, jul 2005. ISSN 1526-5528. Available at: https://doi.org/10.1287/ijoc.1050.0136.

KOTHARI, P.; KREISS, S.; ALAHI, A. Human Trajectory Forecasting in Crowds: A Deep Learning Perspective. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, 2021.

KRAVCHENKO, M. **Future ui**. 2021. `https://br.pinterest.com/krava88/future-ui/`. (Accessed on 06/11/2021).

KRUSCHKE, J. K. **Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan**. Cambridge, MA, USA: Academic Press, 2013.

KRUSE, T.; PANDEY, A. K.; ALAMI, R.; KIRSCH, A. Human-aware robot navigation: A survey. **Robotics and Autonomous Systems**, Elsevier, v. 61, n. 12, p. 1726–1743, 2013.

KUHL, F.; WEATHERLY, R.; DAHMANN, J. **Creating Computer Simulation Systems: An Introduction to the High Level Architecture**. Upper Saddle River, NJ: Prentice Hall, 1999. ISBN 9780130225115.

KUHN, M.; JOHNSON, K. **Feature Engineering and Selection: A Practical Approach for Predictive Models**. Boca Raton, FL, USA: Taylor & Francis, 2019. ISBN 978-1-138-07922-9.

KUHN, M.; JOHNSON, K. *et al.* **Applied Predictive Modeling**. New York, NY, USA: Springer, 2013.

KUMAR, V. R. S.; RAGHUNANDAN, B. N.; KAWAKAMI, T. Studies on Internal Ballistics of Dual-Thrust Motors for Nozzleless Propulsion. *In*: **46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit**. **Proceedings** [...]. Nashville, TN, USA: American Institute of Aeronautics and Astronautics (AIAA), 2010.

KUNG, C.; CHIANG, F. A study of missile maximum capture area and fighter minimum evasive range for negotiation team air combat. *In*: **2015 15th International Conference on Control, Automation and Systems (ICCAS)**. **Proceedings** [...]. IEEE Press, 2015. p. 207–212. Available at: https://doi.org/10.1109/ICCAS.2015.7364908.

KURNIAWAN, B.; VAMPLEW, P.; PAPASIMEON, M.; DAZELEY, R.; FOALE, C. An Empirical Study of Reward Structures for Actor-Critic Reinforcement Learning in Air Combat Manoeuvring Simulation. *In*: LIU, J.; BAILEY, J. (Ed.). **AI 2019: Advances in Artificial Intelligence**. **Proceedings** [...]. Cham, Switzerland: Springer International Publishing, 2019. p. 54–65. ISBN 978-3-030-35288-2.

KURODA, T.; IMADO, F. Improved advanced missile guidance system against a hypersonic target with short maneuvering time. *In*: AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS. **AIAA Guidance, Navigation, and Control Conference**. **Proceedings** [...]. Portland, OR, USA: AIAA, 1990. p. 3379. Available at: https://dx.doi.org/10.2514/6.1990-3379.

KUROSWISKI, A. R. **Modelagem e Simulação Baseada em Agentes como Ferramenta de Apoio à Avaliação de Capacidades de Defesa Aérea.** Dissertation (Master's) — Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil, 2020.

KUROSWISKI, A. R.; MEDEIROS, F. L. L.; MARCHI, M. M. D.; PASSARO, A. Beyond visual range air combat simulations: validation methods and analysis using agent-based models. **The Journal of Defense Modeling and Simulation**, p. 15485129231211915, publisher: SAGE Publications, November 2023. ISSN 1548-5129. Available at: https://doi.org/10.1177/15485129231211915.

LAMINAR RESEARCH. **X-Plane**. Columbia, South Carolina, 2024. Available at: https://www.x-plane.com.

LAURENT, C.; PEREYRA, G.; BRAKEL, P.; ZHANG, Y.; BENGIO, Y. Batch normalized recurrent neural networks. *In*: **Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing**. **Proceedings** [...]. Shanghai, China, March 20th-25th: IEEE, 2016. p. 2657–2661.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep Learning. **Nature**, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015.

LEE, K.; LEE, G.; RABELO, L. A Systematic Review of the Multi-Resolution Modeling (MRM) for Integration of Live, Virtual, and Constructive Systems. **Information**, v. 11, p. 480, 10 2020.

LEI, X.; DENG, S.; TANG, S.; HUANG, C.; DONG, K.; ZHANG, Z. Beyond visual range maneuver intention recognition based on attention enhanced tuna swarm optimization parallel BiGRU. **Complex & Intelligent Systems**, October 2023. ISSN 2198-6053. Available at: https://doi.org/10.1007/s40747-023-01257-3.

LEI, Y.; HUO, M.; DENG, Y.; DUAN, H. Multiple UAVs Target Allocation via Stochastic Dominant Learning Pigeon-Inspired Optimization in Beyond-Visual-Range Air Combat. *In*: **2022 12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)**. **Proceedings** [...]. Changbai Mountain, China: IEEE, 2022. p. 1269–1274. ISSN: 2642-6633.

LEI, Y. Y.; JIANG, W. Z.; GAO, B.; ZHU, W. Modeling of Anti-Aircraft Weapon Systems Cooperative Guidance for Anti-Aircraft Missile. **Applied Mechanics and Materials**, v. 719-720, p. 504–513, publisher: Trans Tech Publications Ltd, 2015. ISSN 1662-7482. Available at: https://www.scientific.net/AMM.719-720.504.

LEMAÎTRE, G.; NOGUEIRA, F.; ARIDAS, C. K. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. **Journal of Machine Learning Research**, v. 18, n. 17, p. 1–5, 2017. Available at: http://jmlr.org/papers/v18/16-365.html.

LEONARD, B. **Department of Defense Dictionary of Military and Associated Terms: As Amended Through April 2010**. Darby, USA: DIANE Publishing, 2011.

LI, A.; MENG, Y.; HE, Z. Simulation Research on New Model of Air-to-Air Missile Attack Zone. *In*: IEEE. **2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)**. **Proceedings** [...]. Chongqing, China: IEEE, 2020. v. 1, p. 1998–2002.

LI, B.; BAI, S.; LIANG, S.; MA, R.; NERETIN, E.; HUANG, J. Manoeuvre decision-making of unmanned aerial vehicles in air combat based on an expert actor-based soft actor critic algorithm. **CAAI Transactions on Intelligence Technology**, v. 8, n. 4, p. 1608–1619, March 2023. ISSN 2468-2322. Available at: https://doi.org/10.1049/cit2.12195.

LI, B.; HUANG, J.; BAI, S.; GAN, Z.; LIANG, S.; EVGENY, N.; YAO, S. Autonomous air combat decision-making of uav based on parallel self-play reinforcement learning. **CAAI Transactions on Intelligence Technology**, v. 8, n. 1, p. 64–81, June 2022. ISSN 2468-2322. Available at: https://ietresearch.onlinelibrary.wiley.com/doi/10.1049/cit2.12109.

LI, G.; WANG, Y.; LU, C.; ZHANG, Z. Multi-UAV Air Combat Weapon-Target Assignment Based On Genetic Algorithm And Deep Learning. *In*: IEEE. **2020 Chinese Automation Congress (CAC)**. **Proceedings** [...]. Shanghai, China: IEEE, 2020. p. 3418–3423.

LI, J.; LIU, L. An MCDM Model Based on KL-AHP and TOPSIS and Its Application to Weapon System Evaluation. *In*: QI, E.; SU, Q.; SHEN, J.; WU, F.; DOU, R. (Ed.). **Proceedings of the 5th International Asia Conference on Industrial Engineering and Management Innovation (IEMI2014)**. **Proceedings** [...]. Paris, France: Atlantis Press, 2015. (Proceedings of the International Asia Conference on Industrial Engineering and Management Innovation), p. 257–262. ISBN 978-94-6239-100-0.

LI, L.; LIU, K. The expectation of applying IR guidance in medium range air-to-air missiles. *In*: SPIE. **Infrared Technology and Applications**. **Proceedings** [...]. Baltimore, MD, USA, 2016. p. 12–2244581. Available at: https://dx.doi.org/10.1117/12.2244581.

LI, W.; GUO, L.; SHI, J. Research on Distributed Target Allocation in Multi-UAV Beyond Visual Range Cooperative Air Combat. *In*: YAN, L.; DUAN, H.; YU, X. (Ed.). **Advances in Guidance, Navigation and Control**. **Proceedings** [...]. Singapore: Springer, 2022. (Lecture Notes in Electrical Engineering), p. 1845–1855. ISBN 978-981-15-8155-7.

LI, W.; LYU, Y.; DAI, S.; CHEN, H.; SHI, J.; LI, Y. A Multi-Target Consensus-Based Auction Algorithm for Distributed Target Assignment in Cooperative Beyond-Visual-Range Air Combat. **Aerospace**, v. 9, n. 9, p. 486, publisher: Multidisciplinary Digital Publishing Institute (MDPI), September 2022. ISSN 2226-4310. Available at: https://www.mdpi.com/2226-4310/9/9/486.

LI, W.; SHI, J.; WU, Y.; WANG, Y.; LYU, Y. A Multi-UCAV cooperative occupation method based on weapon engagement zones for beyond-visual-range air combat. **Defence Technology**, v. 18, n. 6, p. 1006–1022, 2022. ISSN 2214-9147. Available at: https://www.sciencedirect.com/science/article/pii/S2214914721000714.

LIMA-FILHO, G. M.; KUROSWISKI, A. R.; MEDEIROS, F. L. L.; VOSKUIJL, M.; MONSUUR, H.; PASSARO, A. Optimization of Unmanned Air Vehicle Tactical Formation in War Games. **IEEE Access**, v. 10, p. 21727–21741, 2022.

LIMA-FILHO, G. M.; MEDEIROS, F. L. L.; PASSARO, A. Decision Support System for Unmanned Combat Air Vehicle in Beyond Visual Range Air Combat Based on Artificial Neural Networks. **Journal of Aerospace Technology and Management**, SciELO Brasil, v. 13, p. 12–28, 2021.

LIMA, L. S.; GIANNICO, R. H.; BRITO, D. D. C.; DANTAS, A. G. S.; DANTAS, J. P. A. Aprendizado de Máquina para a Otimização da Obtenção de Resultados em Simulações de Defesa Aeroespacial. *In*: **Congresso Acadêmico sobre Defesa Nacional**. **Proceedings** [...]. Rio de Janeiro, RJ, Brazil: Escola Superior de Defesa (ESD), 2024.

LIN, Z.; MING'AN, T.; WEI, Z.; SHENGYUN, Z. Sequential maneuvering decisions based on multi-stage influence diagram in air combat. **Journal of Systems Engineering and Electronics**, Elsevier, v. 18, n. 3, p. 551–555, 2007.

LING, B.; BO, L.; BINGZHENG, S.; LINGCUN, Q.; CHENGQI, X.; YAFENG, N. A cognitive study of multicolour coding for the head-up display (HUD) of fighter aircraft in multiple flight environments. *In*: IOP PUBLISHING. **Journal of Physics: Conference Series**. **Proceedings** [...]. Bristol, UK: IOP Publishing, 2019. v. 1215, n. 1, p. 012032.

LIPPMANN, R. P. An introduction to computing with neural nets. **SIGARCH Comput. Archit. News**, Association for Computing Machinery, New York, NY, USA, v. 16, n. 1, p. 7–25, mar 1988. ISSN 0163-5964. Available at: https://doi.org/10.1145/44571.44572.

LITTLEWOOD, P. F. Simulation: Training for industry and the military services. **Interdisciplinary Science Reviews**, SAGE Publications Sage UK: London, England, v. 16, n. 2, p. 143–150, 1991.

LIU, C.; HE, J.; LI, Q.; WANG, M. An Air Combat Simulation System for Intelligent Decision-Making. *In*: **2020 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)**. **Proceedings** [...]. Hangzhou, China: IEEE, 2020. v. 2, p. 104–108.

LIU, J. Air Target Intention Recognition Based on Incomplete Multi-Granulation Rough Set. *In*: **2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)**. **Proceedings** [...]. Dalian, China: IEEE, 2022. p. 944–947. Available at: https://ieeexplore.ieee.org/document/9777527.

LIU, X.; YIN, Y.; SU, Y.; MING, R. A Multi-UCAV Cooperative Decision-Making Method Based on an MAPPO Algorithm for Beyond-Visual-Range Air Combat. **Aerospace**, v. 9, n. 10, 2022. ISSN 2226-4310. Available at: https://www.mdpi.com/2226-4310/9/10/563.

LIU, Y.; YANG, Z.; HUANG, J.; ZHAN, G.; LI, X.; ZHOU, D. A Maneuver Control Method for Stealthy Engagement in Beyond-visual-range Air Combat Based on Sliding Mode Control. *In*: **2022 22nd International Conference on Control, Automation and Systems (ICCAS)**. **Proceedings** [...]. Jeju Island, South Korea: IEEE, 2022. p. 1333–1338. ISSN: 2642-3901. Available at: https://ieeexplore.ieee.org/document/10003726.

LIU, Y.; ZENG, Y.; CHEN, Y.; TANG, J.; PAN, Y. Self-Improving Generative Adversarial Reinforcement Learning. *In*: **Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems**. **Proceedings** [...]. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2019. (AAMAS '19), p. 52–60. ISBN 9781450363099.

LU, H.; WU, B.; CHEN, J. Fighter Equipment Contribution Evaluation Based on Maneuver Decision. **IEEE Access**, v. 9, p. 132241–132254, conference Name: IEEE Access, 2021. ISSN 2169-3536. Available at: https://ieeexplore.ieee.org/document/9547322.

LUCAS, T. W.; KELTON, W. D.; SÁNCHEZ, P. J.; SANCHEZ, S. M.; ANDERSON, B. L. Changing the Paradigm: Simulation, Now a Method of First Resort. **Naval Research Logistics (NRL)**, Wiley Online Library, v. 62, n. 4, p. 293–303, 2015.

LUO, C.; LEI, L.; CAI, W.; CAI, S.; ZHANG, T. A Dual-Mode MAC Protocol over Mobile Ad Hoc Networks. *In*: XING, S.; CHEN, S.; WEI, Z.; XIA, J. (Ed.). **Unifying Electrical Engineering and Electronics Engineering**. **Proceedings** [...]. New York, NY: Springer, 2014. (Lecture Notes in Electrical Engineering), p. 1639–1646. ISBN 978-1-4614-4981-2.

MA, Y.; GUO, J.; TANG, S. High angle of attack command generation technique and tracking control for agile missiles. **Aerospace Science and Technology**, Elsevier, v. 45, p. 324–334, 2015.

MA, Y.; WANG, G.; HU, X.; LUO, H.; LEI, X. Cooperative Occupancy Decision Making of Multi-UAV in Beyond-Visual-Range Air Combat: A Game Theory Approach. **IEEE Access**, v. 8, p. 11624–11634, 2020.

MAK TECHNOLOGIES. **VR-Forces**. 2023. `https://www.mak.com/mak-one/apps/vr-forces/`. Last accessed 15 April 2023.

MAMESSIER, S.; FEIGH, K. M.; PRITCHETT, A.; DICKSON, D. Pilot Mental Models and Loss of Control. *In*: **AIAA Guidance, Navigation, and Control Conference**. **Proceedings** [...]. National Harbor, MD, USA: American Institute of Aeronautics and Astronautics (AIAA), 2014. Available at: https://arc.aiaa.org/doi/abs/10.2514/6.2014-0609.

MANSIKKA, H.; VIRTANEN, K.; HARRIS, D.; SALOMÄKI, J. Live–virtual–constructive simulation for testing and evaluation of air combat tactics, techniques, and procedures, Part 2: demonstration of the framework. **The Journal of Defense Modeling and Simulation**, v. 18, n. 4, p. 295–308, publisher: SAGE Publications, October 2021. ISSN 1548-5129. Available at: https://doi.org/10.1177/1548512919886378.

MANSIKKA, H.; VIRTANEN, K.; HARRIS, D.; JALAVA, M. Measurement of team performance in air combat – have we been underperforming? **Theoretical Issues in Ergonomics Science**, v. 22, n. 3, p. 338–359, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/1463922X.2020.1779382, March 2021. ISSN 1463-922X. Available at: https://doi.org/10.1080/1463922X.2020.1779382.

MANSIKKA, H.; VIRTANEN, K.; MÄKINEN, L.; HARRIS, D. Normative Performance Measurement in Simulated Air Combat. **Aerospace Medicine and Human Performance**, v. 92, n. 11, p. 908–912, November 2021.

MANSIKKA, H.; VIRTANEN, K.; UGGELDAHL, V.; HARRIS, D. Team situation awareness accuracy measurement technique for simulated air combat - Curvilinear relationship between awareness and performance. **Applied Ergonomics**, v. 96, p. 103473, October 2021. ISSN 0003-6870. Available at: https://www.sciencedirect.com/science/article/pii/S0003687021001204.

MANSIKKA, H. P.; VIRTANEN, K. M.; HARRIS, D. Retrospective Verbal Probing in Evaluation of Pilots' Situation Models in Simulated Air Combat. *In*: **Proceedings of the 34th European Association for Aviation Psychology (EAAP) Conference**. **Proceedings** [...]. Gothenburg, Sweden: European Association for Aviation Psychology (EAAP), 2020. p. 73–82.

MAO, Y.; LI, Q.; HE, J.; XIA, Z.; FEI, A. Construction Method of Air Combat Agent Based on Reinforcement Learning. *In*: **Proceedings of the 2022 10th China Conference on Command and Control**. **Proceedings** [...]. Singapore: Springer Nature, 2022. (Lecture Notes in Electrical Engineering), p. 98–110. ISBN 978-981-19-6052-9.

MARQUES, H. C.; GERALDO, D.; MACÊDO, L. H. Arcabouço de Combate BVR - Plataforma de Apoio à Decisão e ao Treinamento de Pilotos. **Revista Spectrum, n. 19, Comando-Geral de Operações Aéreas**, Setembro 2015.

MASA. **Masa Sword**. 2023. `https://www.masasim.com/en/sword`. Last accessed 15 April 2023.

MASEK, M.; LAM, C. P.; KELLY, L.; BENKE, L.; PAPASIMEON, M. A Genetic Programming Framework for Novel Behaviour Discovery in Air Combat Scenarios. *In*: ERNST, A. T.; DUNSTALL, S.; GARCíA-FLORES, R.; GROBLER, M.; MARLOW, D. (Ed.). **Data and Decision Sciences in Action 2**. **Proceedings** [...]. Cham: Springer International Publishing, 2021. (Lecture Notes in Management and Industrial Engineering), p. 263–277. ISBN 978-3-030-60135-5.

MASTERS, D.; LUSCHI, C. **Revisiting Small Batch Training for Deep Neural Networks**. arXiv.org, 2018. Available at: https://arxiv.org/abs/1804.07612.

MATHWORKS. **MATLAB**. Natick, Massachusetts, 2024. Available at: https://www.mathworks.com/products/matlab.html.

MATHWORKS. **Simulink**. Natick, Massachusetts, 2024. Available at: https://www.mathworks.com/products/simulink.html.

MCGREW, J. S. **Real-time maneuvering decisions for autonomous air combat**. Thesis (PhD) — Massachusetts Institute of Technology, 2008.

MCKAY, M. D.; BECKMAN, R. J.; CONOVER, W. J. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. **Technometrics**, Taylor & Francis, Ltd., American Statistical Association, American Society for Quality, v. 21, n. 2, p. 239–245, 1979. ISSN 00401706. Available at: http://www.jstor.org/stable/1268522.

MCKEOWN, D. M.; KLAUDA, E. J.; LIO, J. J.; SIDDHARTH, S. Correlated Terrain for Serious Games: Achieving Interoperability Across Diverse Runtime Environments. **IMAGE**, 2012.

MEI, D.; WANG, L. Research on Beyond Visual Range Air Combat Interception Guidance Law. *In*: **2021 International Conference on Intelligent Computing, Automation and Applications (ICAA)**. **Proceedings** [...]. Nanjing, China: IEEE, 2021. p. 369–373. Available at: https://ieeexplore.ieee.org/abstract/document/9653541.

MENARDI, G.; TORELLI, N. Training and Assessing Classification Rules with Imbalanced Data. **Data Mining and Knowledge Discovery**, Springer, v. 28, n. 1, p. 92–122, 2014.

MENG, D.; WANG, Y.; CHEN, Y.; ZHONG, L. Tactics Decision-Making Based on Granular Computing in Cooperative Team Air Combat. *In*: WEN, Z.; LI, T. (Ed.). **Practical Applications of Intelligent Systems**. **Proceedings** [...]. Berlin, Heidelberg: Springer, 2014. (Advances in Intelligent Systems and Computing), p. 915–925. ISBN 978-3-642-54927-4.

MENTHE, L.; GOLDFELD, D. A.; TINGSTAD, A.; LINGEL, S.; GEIST, E.; BRUNK, D.; WICKER, A.; LOVELL, S.; GINTAUTAS, B.; STICKELLS, A.; CORDOVA, A. **Technology Innovation and the Future of Air Force Intelligence Analysis: Volume 1, Findings and Recommendations**. Santa Monica, CA: RAND Corporation, 2021.

MILLER, J. A.; MINEAR, P. D.; JR, A. F. N.; DELULLO, A. M.; GEIGER, B. R.; LONG, L. N.; HORN, J. F. Intelligent unmanned air vehicle flight systems. **Journal of Aerospace Computing, Information, and Communication**, v. 4, n. 5, p. 816–835, 2007.

MITTAL, V.; DAVIDSON, A. Combining wargaming with modeling and simulation to project future military technology requirements. **IEEE Transactions on Engineering Management**, IEEE, v. 68, n. 4, p. 1195–1207, 2021.

MNIH, V.; BADIA, A. P.; MIRZA, M.; GRAVES, A.; LILLICRAP, T.; HARLEY, T.; SILVER, D.; KAVUKCUOGLU, K. Asynchronous methods for deep reinforcement learning. *In*: **Proceedings of the 33rd International Conference on Machine Learning (ICML)**. **Proceedings** [...]. New York, NY, USA: PMLR, 2016. (Proceedings of Machine Learning Research, v. 48), p. 1928–1937. Available at: https://proceedings.mlr.press/v48/mniha16.html.

MNIH, V.; KAVUKCUOGLU, K.; SILVER, D.; RUSU, A. A.; VENESS, J.; BELLEMARE, M. G.; GRAVES, A.; RIEDMILLER, M.; FIDJELAND, A. K.; OSTROVSKI, G. *et al.* Human-level control through deep reinforcement learning. **Nature**, Nature Publishing Group, v. 518, n. 7540, p. 529–533, 2015.

MONTGOMERY, D. C. **Design and Analysis of Experiments**. Hoboken, NJ, USA: John Wiley & Sons, 2017.

MOORE, T. Solid Propulsion Enabling Technologies and Milestones for Navy Air-launched Tactical Missiles. *In*: AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS. **AIAA Centennial of Naval Aviation Forum "100 Years of Achievement and Progress"**. **Proceedings** [...]. Virginia Beach, VA, USA: American Institute of Aeronautics and Astronautics, 2011. p. 6941. Available at: https://arc.aiaa.org/doi/10.2514/6.2011-6941.

MUKAI, H.; TANIKAWA, A.; TUNAY, I.; OZCAN, I.; KATZ, I.; SCHÄTTLER, H.; RINALDI, P.; WANG, G.; YANG, L.; SAWADA, Y. Sequential linear-quadratic method for differential games with air combat applications. **Computational Optimization and Applications**, Springer, v. 25, n. 1, p. 193–222, 2003.

MURATET, M.; GARBARINI, D. Accessibility and serious games: What about entity-component-system software architecture? *In*: **Games and Learning Alliance: 9th International Conference, GALA 2020, Laval, France, December 9–10, 2020, Proceedings**. **Proceedings** [...]. Berlin, Heidelberg: Springer-Verlag, 2020. p. 3–12. ISBN 978-3-030-63463-6. Available at: https://doi.org/10.1007/978-3-030-63464-3_1.

NAIR, A.; MCGREW, B.; ANDRYCHOWICZ, M.; ZAREMBA, W.; ABBEEL, P. Overcoming exploration in reinforcement learning with demonstrations. *In*: **2018 IEEE International Conference on Robotics and Automation (ICRA)**. **Proceedings** [...]. IEEE Press, 2018. p. 6292–6299. Available at: https://doi.org/10.1109/ICRA.2018.8463162.

NATO. **M&S AS A SERVICE. Allied Framework for Modelling and Simulation as a Service (MSaaS)**. 2023. `https://nmsg.sto.nato.int/themes/msaas`. Accessed on 03 May 2023.

NEMENYI, P. B. **Distribution-free Multiple Comparisons**. Thesis (PhD) — Princeton University, Princeton, NJ, USA, 1963. Ph.D. Dissertation.

NEWTON, C.; SINGLETON, J.; COPLAND, C.; KITCHEN, S.; HUDACK, J. Scalability in modeling and simulation systems for multi-agent, AI, and machine learning applications. *In*: **Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III**. **Proceedings** [...]. SPIE, 2021. v. 11746, p. 534–552. Available at: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11746/1174626/Scalability-in-modeling-and-simulation-systems-for-multi-agent-AI-/10.1117/12.2585723.full.

NG, A. Y. Feature selection, L1 vs. L2 regularization, and rotational invariance. *In*: **Proceedings of the Twenty-First International Conference on Machine Learning**. **Proceedings** [...]. New York, NY, USA: Association for Computing Machinery, 2004. (ICML '04), p. 78. ISBN 1581138385. Available at: https://doi.org/10.1145/1015330.1015435.

NOAMAN, N. M.; MAHIR, R. A.; FAIZ, M. Z. A. Boost-Sustain Missile Motor Performance with Fixed Predetermined Coast Time Interval. **European Journal of Molecular & Clinical Medicine**, v. 7, n. 2, p. 5070–5079, 2020.

OEZTUERK, A. **Grundsatzbetrachtung der FLAMES Runtime Suite (Basic Consideration of the FLAMES Runtime Suite)**. Munich, Germany, 2016.

OPEN'KO, P.; DRANNYK, P.; KOBZEV, V.; BROVKO, M.; ZALEVSKY, G. Substantiation of Reliability Requirements for Mobility Means of Surface-to-Air Missile Systems. **Advances in Military Technology**, v. 12, n. 1, p. 91–99, 2017.

OZDEMIR, M. R.; CEVHER, L.; ERTEKIN, S. AI-based Air-to-Surface Mission Planning using Predictive Launch Acceptability Region Approach. *In*: IEEE. **2021 International Conference on Military Technologies (ICMT)**. **Proceedings** [...]. Brno, Czech Republic: IEEE, 2021. p. 1–8.

PARK, B.; KIM, T.; TAHK, M. Range-to-go weighted optimal guidance with impact angle constraint and seeker's look angle limits. **IEEE Transactions on Aerospace and Electronic Systems**, v. 52, n. 3, p. 1241–1256, 2016.

PARK, S.; PARK, S.; CHOI, H.; AN, K.; KIM, Y. An Expert Data-Driven Air Combat Maneuver Model Learning Approach. *In*: **AIAA Scitech 2021 Forum**. American Institute of Aeronautics and Astronautics, 2021. View Video Presentation: https://doi.org/10.2514/6.2021-0526.vid. Available at: https://arc.aiaa.org/doi/abs/10.2514/6.2021-0526.

PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L.; DESMAISON, A.; KOPF, A.; YANG, E.; DEVITO, Z.; RAISON, M.; TEJANI, A.; CHILAMKURTHY, S.; STEINER, B.; FANG, L.; BAI, J.; CHINTALA, S. **PyTorch: An Imperative Style, High-Performance Deep Learning Library**. 2024. `https://pytorch.org/`. Version 2.4.1.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY Édouard. Scikit-learn: Machine learning in python. **Journal of Machine Learning Research**, v. 12, n. 85, p. 2825–2830, 2011. Available at: http://jmlr.org/papers/v12/pedregosa11a.html.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PENGYUAN, L.; JIANG, X.; WANGWEI, Z. WTA model study of air defense missile system based on particle algorithm. *In*: **Proceedings of the 2014 IEEE Chinese Guidance, Navigation and Control Conference**. **Proceedings** [...]. Yantai, China: IEEE, 2014. p. 1534–1538. Available at: https://ieeexplore.ieee.org/abstract/document/7007420.

PETERSEN, J. F.; AQUINO, M. R. C.; SALLES, R. N. Plataforma AEROGRAF: um SIG voltado para a Força Aérea. **Revista Spectrum, n. 11, Comando-Geral de Operações Aéreas**, Setembro 2008.

PETNEHÁZI, G. **Recurrent Neural Networks for Time Series Forecasting**. 2019. Available at: https://arxiv.org/abs/1901.00069.

PHILLIPS, C. Computational method for determining missile engagement envelopes. *In*: **29th Aerospace Sciences Meeting**. **Proceedings** [...]. Reno, USA: American Institute of Aeronautics and Astronautics (AIAA), 1991. Available at: https://arc.aiaa.org/doi/abs/10.2514/6.1991-588.

PIAO, H.; HAN, Y.; CHEN, H.; PENG, X.; FAN, S.; SUN, Y.; LIANG, C.; LIU, Z.; SUN, Z.; ZHOU, D. Complex relationship graph abstraction for autonomous air combat collaboration: A learning and expert knowledge hybrid approach. **Expert Systems with Applications**, v. 215, p. 119285, April 2023. ISSN 0957-4174. Available at: https://www.sciencedirect.com/science/article/pii/S095741742202303X.

PIAO, H.; SUN, Z.; MENG, G.; CHEN, H.; QU, B.; LANG, K.; SUN, Y.; YANG, S.; PENG, X. Beyond-Visual-Range Air Combat Tactics Auto-Generation by Reinforcement Learning. *In*: **2020 International Joint Conference on Neural Networks (IJCNN)**. **Proceedings** [...]. Glasgow, UK: IEEE, 2020. p. 1–8.

POROPUDAS, J.; VIRTANEN, K. Analyzing air combat simulation results with dynamic bayesian networks. *In*: **2007 Winter Simulation Conference**. **Proceedings** [...]. Washington, D.C., USA: IEEE, 2007. p. 1370–1377.

PORTREY, A. M.; SCHREIBER, B.; BENNETT, W. The pairwise escape-g metric: a measure for air combat maneuvering performance. *In*: **Proceedings of the 37th Conference on Winter Simulation**. **Proceedings** [...]. Orlando, USA: Winter Simulation Conference, 2005. (WSC '05), p. 1101–1108. ISBN 0780395190.

PRABHU, A.; MANIVANNAN, H.; SRIDHAR, K.; AHMED, S.; SHRINIVASAN, L. Decision fusion using type-1 fuzzy logic for situation assessment in aviation scenarios. *In*: **2014 International Conference on Advances in Electronics Computers and Communications**. **Proceedings** [...]. Bangalore, India: IEEE, 2014. p. 1–6.

PREMAKUMARA, N.; JALAIAN, B.; SURI, N.; SAMANI, H. Improving Object Detection Robustness against Natural Perturbations through Synthetic Data Augmentation. *In*: **Proceedings of the 2023 Asia Conference on Computer Vision, Image Processing and Pattern Recognition**. **Proceedings** [...]. New York, NY, USA: Association for Computing Machinery, 2023. (CVIPPR '23). ISBN 9798400700033. Available at: https://doi.org/10.1145/3596286.3596293.

PRESAGIS. **STAGE**. 2023. `https://www.presagis.com/en/product/stage/`. Last accessed 15 April 2023.

PRIDDY, K. L.; KELLER, P. E. **Artificial Neural Networks: An Introduction**. Bellingham, WA, USA: SPIE Press, 2005.

PROCTOR, M. D.; GUBLER, J. C. Military simulation worlds and organizational learning. *In*: **Proceedings of the 30th Conference on Winter Simulation**. **Proceedings** [...]. Washington, DC, USA: IEEE Computer Society Press, 1998. (WSC '98), p. 773–780. ISBN 0780351347.

PULLEN, J. M.; ABABNEH, M.; NICKLAS, L.; CONNOR, M.; BARRETO, A. An Open Source MSDL/C-BML Interface to VR-Forces. **How is M&S Interoperability Different from Other Interoperability Domains?**, p. 28, 2012.

PUTATUNDA, S.; RAMA, K. A Comparative Analysis of Hyperopt as Against Other Approaches for Hyper-Parameter Optimization of XGBoost. *In*: **Proceedings of the 2018 International Conference on Signal Processing and Machine Learning**. **Proceedings** [...]. New York, NY, USA: Association for Computing Machinery, 2018. (SPML '18), p. 6–10. ISBN 9781450366052. Available at: https://doi.org/10.1145/3297067.3297080.

QIAN, C.; HAN, B.; TANG, Y.; DUAN, B.; WU, Y.; WANG, L. Missile Attack Zone Fitting Based on K-SAE-SVM. *In*: IEEE. **2022 2nd International Conference on Networking Systems of AI (INSAI)**. **Proceedings** [...]. Shanghai, China: IEEE, 2022. p. 209–215.

QIAN, C.; ZHANG, X.; LI, L.; ZHAO, M.; FANG, Y. SSRN Scholarly Paper, **H3E: Learning Air Combat with a Three-Level Hierarchical Framework Embedding Expert Knowledge**. Rochester, NY: Social Science Research Network (SSRN), May 2023. Available at: https://papers.ssrn.com/abstract=4454247.

QIAN, C.; ZHANG, X.; LI, L.; ZHAO, M.; FANG, Y. H3e: Learning air combat with a three-level hierarchical framework embedding expert knowledge. **Expert Systems with Applications**, v. 245, p. 1–12, July 2024. ISSN 0957-4174. Available at: https://www.sciencedirect.com/science/article/pii/S0957417423035868.

QIU, X.; YAO, Z.; TAN, F.; ZHU, Z.; LU, J. One-to-one Air-combat Maneuver Strategy Based on Improved TD3 Algorithm. *In*: **Proceedings of the 2020 Chinese Automation Congress (CAC)**. **Proceedings** [...]. Shanghai, China: IEEE, 2020. p. 5719–5725. ISSN: 2688-0938. Available at: https://ieeexplore.ieee.org/abstract/document/9327310.

RAFFIN, A.; HILL, A.; ERNESTUS, M.; GLEAVE, A.; KANERVISTO, A.; DORMANN, N. **Stable Baselines3**. 2019. `https://github.com/DLR-RM/stable-baselines3`. Last accessed: 15 April 2023.

RAGIRA, O. J. Design of Radar to Detect a Target at an Arbitrary Standoff Range. **International Journal of Engineering Trends and Technology (IJETT)**, v. 46, n. 8, p. 440–444, 2017. Available at: https://dx.doi.org/10.14445/22315381/IJETT-V46P265.

RAIA SOFTWARE. **Tacview**. 2024. `https://www.tacview.net`. Version 1.8.8.

RAO, N.; KASHYAP, S.; GOPALARATNAM, G.; MANDAL, D. Situation and Threat Assessment in BVR Combat. *In*: **AIAA Guidance, Navigation, and Control Conference**. **Proceedings** [...]. Portland, OR, USA: American Institute of Aeronautics and Astronautics (AIAA), 2011. Available at: https://arc.aiaa.org/doi/abs/10.2514/6.2011-6241.

RAO, N. P.; KASHYAP, S. K.; GOPALARATNAM, G.; MANDAL, D. Situation and threat assessment in bvr combat. *In*: **Proceedings of the AIAA Guidance, Navigation, and Control Conference**. **Proceedings** [...]. Portland, Oregon, USA: American Institute of Aeronautics and Astronautics (AIAA), 2011. p. 726–731. Available at: https://doi.org/10.2514/6.2011-6241.

RASHID, A. B.; KAUSIK, A. K.; SUNNY, A. A. H.; BAPPY, M. H. Artificial Intelligence in the Military: An Overview of the Capabilities, Applications, and Challenges. **International Journal of Intelligent Systems**, Wiley Online Library, n. 1, p.

8676366, 2023. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1155/2023/8676366.

RATHER, I. H.; KUMAR, S. Generative adversarial network based synthetic data training model for lightweight convolutional neural networks. **Multimedia Tools Appl.**, Kluwer Academic Publishers, USA, v. 83, n. 2, p. 6249–6271, maio 2023. ISSN 1380-7501. Available at: https://doi.org/10.1007/s11042-023-15747-6.

RAZA, M. A.; LIANG, W. Robust performance optimization of dual thrust rocket motor. **Aircraft Engineering and Aerospace Technology**, Emerald Group Publishing Limited, v. 84, n. 4, p. 244–251, 2012.

REDDEN, J. J. Military keynote address and panel discussion: military simulation and modeling—today—tomorrow. *In*: **Proceedings of the 27th Conference on Winter Simulation**. **Proceedings** [...]. USA: IEEE Computer Society, 1995. (WSC '95), p. 1133. ISBN 0780330188. Available at: https://doi.org/10.1145/224401.224784.

REED, R. D.; MARKS, R. J. **Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks**. Cambridge, MA, USA: MIT Press, 1998. ISBN 0262181908.

REINISCH, F.; STROHAL, M.; STÜTZ, P. Behaviour Modelling of Computer-Generated-Forces in Beyond-Visual-Range Air Combat. *In*: **Proceedings of the 12th International Conference on Simulation and Modeling Methodologies, Technologies and Applications**. **Proceedings** [...]. Lisbon, Portugal: SCITEPRESS - Science and Technology Publications, 2022. p. 327–335. ISBN 978-989-758-578-4. Available at: https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0011306600003274.

RIOS-MARTINEZ, J.; RENZAGLIA, A.; SPALANZANI, A.; MARTINELLI, A.; LAUGIER, C. Navigating between People: A Stochastic Optimization Approach. *In*: **Proceedings of the 2012 IEEE International Conference on Robotics and Automation**. **Proceedings** [...]. Saint Paul, MN, USA: IEEE, 2012. p. 2880–2885.

RIOS-MARTINEZ, J.; SPALANZANI, A.; LAUGIER, C. From proxemics theory to socially-aware navigation: A survey. **International Journal of Social Robotics**, Springer, Heidelberg, Germany, v. 7, n. 2, p. 137–153, 2015.

ROUSSEEUW, P. J.; DRIESSEN, K. V. **Detecting Outliers: Theory and Applications**. Hoboken, NJ, USA: John Wiley & Sons, 2019.

RU, J.; GUAN, Y.; YU, S.; XU, H. A multi-uav pursuit strategy based on two-stage collaborative search. *In*: QU, Y.; GU, M.; NIU, Y.; FU, W. (Ed.). **Proceedings of 3rd 2023 International Conference on Autonomous Unmanned Systems (3rd ICAUS 2023)**. **Proceedings** [...]. Singapore: Springer Nature, 2024. p. 58–67. Available at: https://doi.org/10.1007/978-981-97-1083-6_6.

RUDENKO, A.; PALMIERI, L.; HERMAN, M.; KITANI, K. M.; GAVRILA, D. M.; ARRAS, K. O. Human Motion Trajectory Prediction: A Survey. **The International Journal of Robotics Research**, Sage Publications Sage UK: London, England, v. 39, n. 8, p. 895–935, 2020.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 4th. ed. London: Pearson, 2021.

RYAN, T. R.; MITTAL, V. Potential for Army Integration of Autonomous Systems by Warfighting Function. **Military Review**, v. 99, n. 5, p. 122, 2019.

RYU, H.; BAI, J. H.; TAHK, M.; PARK, S.; YUN, J. No-Escape Envelope with Field of Regard Constraint using Gradient-Based Direct Method for Pursuit-Evasion Games. **International Journal of Aeronautical and Space Sciences**, Springer, v. 19, p. 675–684, 2018.

SAMSONOVICH, A. V.; AHA, D. W. Character-Oriented Narrative Goal Reasoning in Autonomous Actors. *In*: **Proceedings of the 2015 Annual Conference on Advances in Cognitive Systems: Workshop on Goal Reasoning**. **Proceedings** [...]. Cambridge, MA, USA: MIT, 2015. p. 166–181.

SANDSTRÖM, V.; LUOTSINEN, L.; OSKARSSON, D. Fighter Pilot Behavior Cloning. *In*: **Proceedings of the 2022 International Conference on Unmanned Aircraft Systems (ICUAS)**. **Proceedings** [...]. Dubrovnik, Croatia: IEEE, 2022. p. 686–695. Available at: https://dx.doi.org/10.1109/icuas54217.2022.9836131.

SCHADD, M.; REUS, N. de; UILKEMA, S.; VOOGD, J. Data-driven behavioural modelling for military applications. **Journal of Defence & Security Technologies**, v. 4, n. 1, p. 12–36, 2022. Available at: https://www.jdst.eu/publications/data-driven-behavioural-modelling-military-applications.

SCHARRE, P. D. The Opportunity and Challenge of Autonomous Systems. **Autonomous Systems: Issues for Defence Policymakers**, HQ Sact, p. 3–26, 2015.

SCHULKER, D.; LIM, N.; MATTHEWS, L. J.; GRIMM, G. E.; LAWRENCE, A.; FIROZ, P. S. **Can Artificial Intelligence Help Improve Air Force Talent Management? An Exploratory Application**. Santa Monica, CA: RAND Corporation, 2021.

SCHULMAN, J.; WOLSKI, F.; DHARIWAL, P.; RADFORD, A.; KLIMOV, O. Proximal policy optimization algorithms. **arXiv preprint arXiv:1707.06347**, 2017.

SCHUSS, Z. **Theory and Applications of Stochastic Processes: An Analytical Approach**. New York, NY, USA: Springer, 2009. ISBN 978-1-4419-1604-4.

SCHWARTZ, P. J.; O'NEILL, D. V.; BENTZ, M. E.; BROWN, A.; DOYLE, B. S.; LIEPA, O. C.; LAWRENCE, R.; HULL, R. D. AI-enabled wargaming in the military decision making process. *In*: PHAM, T.; SOLOMON, L.; RAINEY, K. (Ed.). **Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications II**. **Proceedings** [...]. SPIE, 2020. v. 11413, p. 114130H. Available at: https://doi.org/10.1117/12.2560494.

SCUKINS, E.; COSTA, A. N.; ÖGREN, P. A data-driven method for estimating formation flexibility in beyond-visual-range air combat. *In*: **Proceedings of the 2024 International Conference on Unmanned Aircraft Systems (ICUAS)**. **Proceedings** [...]. Chania, Crete, Greece: IEEE, 2024. p. 1–7. Available at: https://doi.org/10.1109/ICUAS60882.2024.10234567.

SCUKINS, E.; KLEIN, M.; KROON, L.; ÖGREN, P. Monte Carlo Tree Search and Convex Optimization for Decision Support in Beyond-Visual-Range Air Combat. *In*: **2023 International Conference on Unmanned Aircraft Systems (ICUAS)**. **Proceedings** [...]. Warsaw, Poland: IEEE, 2023. p. 48–55. ISSN: 2575-7296. Available at: https://ieeexplore.ieee.org/abstract/document/10156124.

SCUKINS, E.; KLEIN, M.; KROON, L.; ÖGREN, P. Bvr gym: A reinforcement learning environment for beyond-visual-range air combat. **arXiv preprint arXiv:2403.17533**, 2024.

SCUKINS, E.; KLEIN, M.; KROON, L.; ÖGREN, P. **Deep Learning Based Situation Awareness for Multiple Missiles Evasion**. 2024. Available at: https://arxiv.org/abs/2402.10101.

SCUKINS, E.; KLEIN, M.; ÖGREN, P. Enhancing Situation Awareness in Beyond Visual Range Air Combat with Reinforcement Learning-based Decision Support. *In*: **Proceedings of the 2023 International Conference on Unmanned Aircraft Systems (ICUAS)**. **Proceedings** [...]. Warsaw, Poland: IEEE, 2023. p. 56–62. ISSN: 2575-7296. Available at: https://ieeexplore.ieee.org/abstract/document/10156497.

SEABOLD, S.; PERKTOLD, J. **statsmodels: Econometric and Statistical Modeling with Python**. 2010. `https://www.statsmodels.org`. Accessed: 2024-11-29.

SHI, G.; PU, J.; ZHANG, L.; GENG, X.; ZHOU, Y.; ZHAO, Y. Situation Assessment Based on Multi-Entity Bayesian Network. *In*: **Proceedings of the 2020 IEEE 16th International Conference on Control & Automation (ICCA)**. **Proceedings** [...]. Virtual Conference, originally planned for Sapporo, Hokkaido, Japan: IEEE, 2020. p. 702–707. ISSN: 1948-3457. Available at: https://ieeexplore.ieee.org/abstract/document/9264559.

SHI, W.; LI, H.; HE, W.; ZHANG, C. Threat Assessment of Air Targets Based on Intent Prediction. *In*: **Proceedings of the 2021 IEEE 5th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)**. **Proceedings** [...]. Xi'an, China: IEEE, 2021. v. 5, p. 311–321. ISSN: 2693-3128. Available at: https://ieeexplore.ieee.org/abstract/document/9587069.

SHI, Z.; LIANG, X.; ZHANG, J.; FU, Q. Modeling and Simulation Analysis on Three-Dimensional Air-to-Air Missile Attack Zone of Two Aircrafts. *In*: IEEE. **2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)**. **Proceedings** [...]. Xiamen, China: IEEE, 2018. p. 1–5.

SHROUT, P. E.; FLEISS, J. L. Intraclass correlations: Uses in assessing rater reliability. **Psychological Bulletin**, American Psychological Association, v. 86, n. 2, p. 420–428, 1979.

SHUKLA, D.; KESHMIRI, S.; BECKAGE, N. M. Imitation Learning for Neural Network Autopilot in Fixed-Wing Unmanned Aerial Systems. *In*: **Proceedings of the 2020 International Conference on Unmanned Aircraft Systems (ICUAS)**. **Proceedings** [...]. Athens, Greece: IEEE, 2020. p. 1508–1517. Available at: https://dx.doi.org/10.1109/ICUAS48674.2020.9213850.

SILVA, I. N. da; SPATTI, D. H.; FLAUZINO, R. A.; LIBONI, L. H. B.; ALVES, S. F. dos R. **Artificial Neural Network Architectures and Training Processes**. Cham: Springer International Publishing, 2017. 21–28 p. ISBN 978-3-319-43162-8. Available at: https://doi.org/10.1007/978-3-319-43162-8_2.

SOKOLOWSKI, J. A.; BANKS, C. M. **Modeling and Simulation for Analyzing Global Events**. Hoboken, NJ: John Wiley & Sons, 2009.

SOLEYMAN, S.; HUNG, F.; KHOSLA, D.; CHEN, Y.; FADAIE, J. G.; NADERI, N. Multi-Agent Autonomous Battle Management Using Deep Neuroevolution. *In*: **Unmanned Systems Technology XXIII**. **Proceedings** [...]. Orlando, FL, United States: SPIE, 2021. v. 11758, p. 117580C. Available at: https://www.spiedigitallibrary.org-/conference-proceedings-of-spie/11758/117580C/Multi-agent-autonomous-battle-management-using-deep-neuroevolution/10.1117/12.2585530.short.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. **J. Mach. Learn. Res.**, JMLR.org, v. 15, n. 1, p. 1929–1958, jan. 2014. ISSN 1532-4435.

STILLION, J. **Trends in Air-to-Air Combat: Implications for Future Air Superiority**. Washington, DC: Center for Strategic and Budgetary Assessments, 2015. Available at: `https://csbaonline.org/research/publications/trends-in-air-to-air-combat-implications-for-future-air-superiority`.

SUAREZ, E. J. A.; BAEZA, V. M. Evaluating the Role of Machine Learning in Defense Applications and Industry. **Machine Learning and Knowledge Extraction**, v. 5, n. 4, p. 1557–1569, 2023. ISSN 2504-4990. Available at: https://www.mdpi.com/2504-4990/5/4/78.

SUMILE, M. S. Collaborative modeling & tailored simulation for course of action validation. *In*: **Proceedings of the Military Modeling & Simulation Symposium**. **Proceedings** [...]. San Diego, CA, USA: Society for Computer Simulation International, 2013. (MMS '13). ISBN 9781627480345.

SUMMEY, D. C.; KEKELIS, G. J. Fused Airborne Sensor Technology. *In*: SPIE. **Proceedings of SPIE - The International Society for Optical Engineering**. **Proceedings** [...]. Orlando, FL, USA: SPIE, 1996. v. 2765, p. 56–67. Available at: https://dx.doi.org/10.1117/12.241225.

SUN, S.; CAO, Z.; ZHU, H.; ZHAO, J. A survey of optimization methods from a machine learning perspective. **IEEE Transactions on Cybernetics**, IEEE, Piscataway, USA, v. 50, n. 8, p. 3668–3681, 2020.

SUN, Y.; WANG, X.; WANG, T.; GAO, P. Modeling of Air-to-air Missile Dynamic Attack Zone Based on Bayesian Networks. *In*: **2020 Chinese Automation Congress (CAC)**. **Proceedings** [...]. Shanghai, China: IEEE, 2020. p. 5596–5601. ISSN: 2688-0938. Available at: https://ieeexplore.ieee.org/abstract/document/9327613.

SUN, Z.; PIAO, H.; YANG, Z.; ZHAO, Y.; ZHAN, G.; ZHOU, D.; MENG, G.; CHEN, H.; CHEN, X.; QU, B.; LU, Y. Multi-agent hierarchical policy gradient for Air Combat Tactics emergence via self-play. **Engineering Applications of Artificial Intelligence**,

Elsevier, v. 98, p. 104112, February 2021. ISSN 0952-1976. Available at: https://www.sciencedirect.com/science/article/pii/S0952197620303547.

SUSENO, P. A. P.; SASONGKO, R. Development of Air Combat Effectiveness Simulation and Analysis Scheme for Beyond Visual Range (BVR) Case. **Applied Mechanics and Materials**, Trans Tech Publications Ltd, Switzerland, v. 842, p. 329–336, publisher: Trans Tech Publications Ltd, 2016. ISSN 1662-7482. Available at: https://www.scientific.net/AMM.842.329.

SUTTON, R. S.; BARTO, A. G. **Reinforcement Learning: An Introduction**. Cambridge, MA, USA: MIT Press, 2018.

SVENMARCKT, P.; DEKKER, S. Decision support in fighter aircraft: from expert systems to cognitive modelling. **Behaviour & Information Technology**, Taylor & Francis, v. 22, n. 3, p. 175–184, 2003.

TAYLOR, J. R. **An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements**. Sausalito, CA, USA: University Science Books, 2017.

TERNION. **FLAMES Development Suite**. 2023. `https://flamesframework.com/`. Last accessed 15 April 2023.

TOMEK, I. Two Modifications of CNN. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 6, n. 11, p. 769–772, 1976.

TOUBMAN, A. Validating Air Combat Behaviour Models for Adaptive Training of Teams. *In*: SOTTILARE, R. A.; SCHWARZ, J. (Ed.). **Adaptive Instructional Systems**. **Proceedings** [...]. Cham, Switzerland: Springer International Publishing, 2019. (Lecture Notes in Computer Science), p. 557–571. ISBN 978-3-030-22341-0.

TOUBMAN, A.; ROESSINGH, J.; POPPINGA, G.; HOU, M.; LUOTSINEN, L.; LØVLID, R. A.; MEYER, C.; RIJKEN, R.; TURCANIK, M. Modeling CGF Behavior with Machine Learning Techniques. National Aerospace Laboratory NLR, 2015.

TOUBMAN, A.; ROESSINGH, J. J.; SPRONCK, P.; PLAAT, A.; HERIK, J. V. D. Rapid Adaptation of Air Combat Behaviour. *In*: **Proceedings of the Twenty-Second European Conference on Artificial Intelligence**. **Proceedings** [...]. The Hague, The Netherlands: IOS Press, 2016. (ECAI'16), p. 1791–1796. ISBN 9781614996712. Available at: https://doi.org/10.3233/978-1-61499-672-9-1791.

TOWERS, M.; TERRY, J. K.; KWIATKOWSKI, A.; BALIS, J. U.; COLA, G. d.; DELEU, T.; GOULAO, M.; KALLINTERIS, A.; KG, A.; KRIMMEL, M.; PEREZ-VICENTE, R.; PIERRE, A.; SCHULHOFF, S.; TAI, J. J.; SHEN, A. T. J.; YOUNIS, O. G. **Gymnasium**. Zenodo, March 2023. Available at: https://zenodo.org/record/8127025.

TRAN, C.; ABRAHAM, A.; JAIN, L. Adaptation of a Mamdani Fuzzy Inference System Using Neuro-Genetic Approach for Tactical Air Combat Decision Support System. *In*: **Proceedings of the 15th Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence**. **Proceedings** [...]. Berlin, Heidelberg: Springer-Verlag, 2002. (AI '02), p. 672–680. ISBN 3540001972.

TREATY, N. A. Modelling and Simulation Group 145: Operationalization of Standardized C2-Simulation Interoperability. 2021.

TUKEY, J. W. **Exploratory Data Analysis**. Reading, MA, USA: Addison-Wesley, 1977.

UMMAH, K.; SETIADI, H.; PASARIBU, H. M.; ANANDITO, D. A Simple Fight Decision Support System for BVR Air Combat Using Fuzzy Logic Algorithm. **AVIA: International Journal of Aviation Science and Engineering**, Faculty of Mechanical and Aerospace Engineering, Institut Teknologi Bandung, Bandung, Indonesia, v. 1, n. 1, number: 1, June 2019. ISSN 2715-6958. Available at: https://avia.ftmd.itb.ac.id/index.php/jav/article/view/8.

UNITED STATES AIR FORCE. **Multi-Command Handbook 11-F16, Volume 5: Air Combat Command**. Washington, D.C., USA, May 1996. Volume 5.

VENKATESAN, R. H.; SINHA, N. K. A New Guidance Law for the Defense Missile of Nonmaneuverable Aircraft. **IEEE Transactions on Control Systems Technology**, v. 23, n. 6, p. 2424–2431, 2015.

VERGEZ, P. L. Tactical Missile Guidance with Passive Seekers Under High Off-Boresight Launch Conditions. **Journal of Guidance, Control, and Dynamics**, v. 21, n. 3, p. 465–470, 1998. Available at: https://doi.org/10.2514/2.4259.

VERGEZ, P. L.; MCCLENDON, J. R. Optimal control and estimation for strapdown seeker guidance of tactical missiles. **Journal of Guidance, Control, and Dynamics**, v. 5, n. 3, p. 225–226, 1982.

VIRTANEN, K.; KARELAHTI, J.; RAIVIO, T. Modeling Air Combat by a Moving Horizon Influence Diagram Game. **Journal of Guidance, Control, and Dynamics**, v. 29, n. 5, p. 1080–1091, 2006. Available at: https://doi.org/10.2514/1.17168.

VIRTANEN, K.; MANSIKKA, H.; KONTIO, H.; HARRIS, D. Weight watchers: NASA-TLX weights revisited. **Theoretical Issues in Ergonomics Science**, Taylor & Francis, v. 23, n. 6, p. 725–748, November 2022. ISSN 1463-922X. Available at: https://doi.org/10.1080/1463922X.2021.2000667.

VIRTANEN, P.; GOMMERS, R.; OLIPHANT, T. E.; HABERLAND, M.; REDDY, T.; COURNAPEAU, D.; BUROVSKI, E.; PETERSON, P.; WECKESSER, W.; BRIGHT, J. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. **Nature methods**, Nature Publishing Group, v. 17, n. 3, p. 261–272, 2020.

VISCARDI, M.; DANTAS, J.; GERALDO, D.; PASSARO, A. Manobra winding: Defesa contra mísseis passivos e semiativos superfície-ar. **Spectrum - The Journal of Operational Applications in Defense Areas**, v. 25, n. 1, p. 12–17, out. 2024. Available at: https://spectrum.ita.br/index.php/spectrum/article/view/403.

WAN, S.; CHANG, X.; LI, Q.; YAN, J. Suboptimal Midcourse Guidance with Terminal-Angle Constraint for Hypersonic Target Interception. **International Journal of Aerospace Engineering**, Hindawi, v. 2019, p. e6161032, publisher: Hindawi, April 2019. ISSN 1687-5966. Available at: https://www.hindawi.com/journals/ijae/2019/6161032/.

WAN, Z.; LI, Y.; ZHANG, Z. Modeling and Analysis of Integrated Combat Network System Based on VOODAC. *In*: **Proceedings of the International Conference on Industrial Control Network and System Engineering Research (ICNSER2019)**. **Proceedings** [...]. Shenyang, Liaoning, China: Association for Computing Machinery, 2019. p. 81–86. ISBN 9781450366274. Available at: https://doi.org/10.1145/3333581.3333598.

WANG, H.; LIU, X.; ZHOU, X. Autonomous UAV Interception via Augmented Adversarial Inverse Reinforcement Learning. *In*: **Proceedings of 2021 International Conference on Autonomous Unmanned Systems (ICAUS 2021)**. **Proceedings** [...]. Singapore: Springer, 2022. p. 2073–2084. ISBN 978-981-16-9492-9. Available at: https://doi.org/10.1007/978-981-16-9492-9_205.

WANG, J. **Formal Methods in Computer Science**. Boca Raton, FL, USA: CRC Press, 2019. ISBN 9781498775359.

WANG, J.; CHAN, W. P.; CARRENO-MEDRANO, P.; COSGUN, A.; CROFT, E. Metrics for Evaluating Social Conformity of Crowd Navigation Algorithms. **arXiv preprint arXiv:2202.01045**, 2022.

WANG, L.; HU, J.; XU, Z.; ZHAO, C. Autonomous maneuver strategy of swarm air combat based on DDPG. **Autonomous Intelligent Systems**, Springer, v. 1, n. 1, p. 15, December 2021. ISSN 2730-616X. Available at: https://doi.org/10.1007/s43684-021-00013-z.

WANG, L.; WEI, H. Research on Autonomous Decision-Making of UCAV Based on Deep Reinforcement Learning. *In*: **Proceedings of the 2022 3rd Information Communication Technologies Conference (ICTC)**. **Proceedings** [...]. Nanjing, China: IEEE, 2022. p. 122–126. Available at: https://doi.org/10.1109/ICTC55111.2022.9778652.

WANG, L.; YANG, G.; LI, Z.; WEI, Y. An ELM Evaluation Model for Fighter Air Combat Effectiveness Against Gross Errors. *In*: **Proceedings of the 2023 IEEE International Conference on Image Processing and Computer Applications (ICIPCA)**. **Proceedings** [...]. Changchun, China: IEEE, 2023. p. 1360–1364. Available at: https://ieeexplore.ieee.org/abstract/document/10257758.

WANG, N.; LI, Z.; LIANG, X.; HOU, Y.; YANG, A. A review of deep reinforcement learning methods and military application research. **Mathematical Problems in Engineering**, v. 2023, n. 1, p. 7678382, 2023. Available at: https://doi.org/10.1155/2023/7678382.

WANG, N.; WANG, L.; BU, Y.; ZHANG, G.; SHEN, L. Tactical Aircraft Pop-Up Attack Planning Using Collaborative Optimization. *In*: SPRINGER. **Proceedings of the Intelligent Computing 5th International Conference on Emerging Intelligent Computing Technology and Applications**. **Proceedings** [...]. Berlin, Heidelberg, Germany: Springer-Verlag, 2009. (ICIC'09), p. 361–370. ISBN 3642040195.

WANG, X.; YANG, Z.; LI, X.; PIAO, H.; HUANG, J.; ZHOU, D. A Beyond Visual Range Air Combat Integrated Threat Assessment Method Based on Target Intention and Event. *In*: YAN, L.; DUAN, H.; DENG, Y. (Ed.). **Advances in Guidance,**

Navigation and Control. **Proceedings** [...]. Harbin, China: Springer Nature, 2023. (Lecture Notes in Electrical Engineering), p. 189–200. ISBN 978-981-19661-3-2.

WANG, X.; YANG, Z.; ZHAN, G.; HUANG, J.; CHAI, S.; ZHOU, D. Tactical Intention Recognition Method of Air Combat Target Based on BiLSTM network. *In*: **Proceedings of the 2022 IEEE International Conference on Unmanned Systems (ICUS)**. **Proceedings** [...]. Guangzhou, China: IEEE, 2022. p. 63–67. ISSN: 2771-7372. Available at: https://ieeexplore.ieee.org/abstract/document/9986667.

WANG, Z.; LI, K.; WANG, L.; LIU, X. Research on an Air Combat Maneuvering Decision Control Method. *In*: **Proceedings of the 42nd Chinese Control Conference (CCC)**. **Proceedings** [...]. Xi'an, China: IEEE, 2023. p. 1–6. Available at: https://dx.doi.org/10.23919/CCC58697.2023.10240373.

WEI, Y.; ZHANG, H.; HUANG, C. **Maneuver Decision-Making For Autonomous Air Combat Through Curriculum Learning And Reinforcement Learning With Sparse Rewards**. 2023. Available at: https://arxiv.org/abs/2302.05838.

WEI, Y.; ZHANG, H.; WANG, Y.; HUANG, C. Autonomous Maneuver Decision-Making Through Curriculum Learning and Reinforcement Learning With Sparse Rewards. **IEEE Access**, IEEE, v. 11, p. 73543–73555, 2023. ISSN 2169-3536. Available at: https://ieeexplore.ieee.org/abstract/document/10188394.

WEILIN, Y.; WEI, D.; SHUANGCHUN, P.; YU, X.; LIANG, P. Decision-making of One-on-one Beyond-Visual-Range Air Combat Based on Improved Q-Network. *In*: **2018 IEEE International Conference on Mechatronics and Automation (ICMA)**. **Proceedings** [...]. Changchun, China: IEEE, 2018. p. 809–815. ISSN: 2152-744X. Available at: https://ieeexplore.ieee.org/abstract/document/8484466.

WEYLAND, L. F.; SCHADD, M. P. D.; HENDERSON, H. C. Exploring the Fidelity of Synthetic Data for Decision Support Systems in Military Applications. *In*: **Proceedings of the 2024 International Conference on Military Communication and Information Systems (ICMCIS)**. **Proceedings** [...]. Koblenz, Germany: IEEE, 2024. p. 1–8. Available at: https://ieeexplore.ieee.org/document/10540934.

WHITTY, L. C. R. The Battle with DataRealities of Bringing Artificial Intelligence to the Battlefield. **Expeditions with MCUP**, Marine Corps University Press, v. 2022, n. 1, p. 1–17, 2022.

WILSON, D. L. Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 2, n. 3, p. 408–421, 1972.

WITSKEN, J. R. **Network-Centric Warfare: Implications for Operational Design**. U.S. Army Command and General Staff College, 2002. Available at: https://apps.dtic.mil/sti/tr/pdf/ADA403832.pdf.

XIA, J.; CHEN, M.; FANG, W. Air Combat Intention Recognition with Incomplete Information Based on Decision Tree and GRU Network. **Entropy**, v. 25, n. 4, p. 671, publisher: Multidisciplinary Digital Publishing Institute, April 2023. ISSN 1099-4300. Available at: https://www.mdpi.com/1099-4300/25/4/671.

XIA, Z.; MAO, Y.; HE, J.; CHEN, J.; LI, Q. Air combat agent construction based on hybrid self-play deep reinforcement learning. *In*: COMMAND, C. I. of; CONTROL (Ed.). **Proceedings of 2023 11th China Conference on Command and Control**. **Proceedings** [...]. Singapore: Springer Nature, 2024. p. 13–21. Available at: https://doi.org/10.1007/978-981-99-9021-4_2.

XIANG-JUN, K.; ZHENG-HONG, G.; HAI-HUA, H. Simulation methodology for fighter conceptual design. *In*: **Proceedings of the 7th International Conference on Computer-Aided Industrial Design and Conceptual Design**. **Proceedings** [...]. Hangzhou, China: IEEE, 2006. p. 1–4. Available at: https://doi.org/10.1109/CAIDCD.2006.329387.

XIAO, B.; PENG, W.; WANG, R.; XU, Y.; WANG, J. A New Guidance Superiority Model for Cooperative Air Combat. *In*: **Fifth Symposium on Novel Optoelectronic Detection Technology and Application**. **Proceedings** [...]. Xi'an, China: SPIE, 2019. v. 11023, p. 1228–1237. Available at: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11023/110234X/A-new-guidance-superiority-model-for-cooperative-air-combat/10.1117/12.2520397.full.

XIE, L.; WANG, Y.; TANG, S.; HUANG, C.; LI, Y.; DONG, K.; SONG, T. A novel adaptive parameter strategy differential evolution algorithm and its application in midcourse guidance maneuver decision-making. **Complex & Intelligent Systems**, August 2023. ISSN 2198-6053. Available at: https://doi.org/10.1007/s40747-023-01186-1.

XU, X.; DOU, L.; XIN, B. Generalized Target-Radar Assignment Using Genetic Algorithm. *In*: **2017 Chinese Automation Congress (CAC)**. **Proceedings** [...]. Jinan, China: IEEE, 2017. p. 5287–5292. Available at: https://ieeexplore.ieee.org/abstract/document/8243720.

YAGCI, O.; NIKBAY, M. Evasive Maneuver Trajectory Optimization of an UCAV Against an Air to Air Missile. *In*: **AIAA AVIATION 2022 Forum**. **Proceedings** [...]. Chicago, IL, USA: American Institute of Aeronautics and Astronautics (AIAA), 2022. p. 3791.

YANG, R.; HUANG, F. M.; GONG, H. J. Best Attack Position Model for BVR Multi-Target Air Combat. **Advanced Materials Research**, v. 1016, p. 511–515, publisher: Trans Tech Publications Ltd, 2014. ISSN 1662-8985. Available at: https://www.scientific.net/AMR.1016.511.

YANG, Z.; SUN, Z.; PIAO, H.; ZHAO, Y.; ZHOU, D.; KONG, W.; ZHANG, K. An Autonomous Attack Guidance Method with High Aiming Precision for UCAV Based on Adaptive Fuzzy Control under Model Predictive Control Framework. **Applied Sciences**, v. 10, n. 16, p. 5677, number: 16 Publisher: Multidisciplinary Digital Publishing Institute, January 2020. ISSN 2076-3417. Available at: https://www.mdpi.com/2076-3417/10/16/5677.

YANG, Z.; SUN, Z.; PIAO, H.; HUANG, J.; ZHOU, D.; REN, Z. Online Hierarchical Recognition Method for Target Tactical Intention in Beyond-Visual-Range Air Combat. **Defence Technology**, Elsevier, v. 18, n. 8, p. 1349–1361, August 2022. ISSN 2214-9147. Available at: https://www.sciencedirect.com/science/article/pii/S2214914722000253.

YANG, Z.; ZHOU, D.; PIAO, H.; ZHANG, K.; KONG, W.; PAN, Q. Evasive Maneuver Strategy for UCAV in Beyond-Visual-Range Air Combat Based on Hierarchical Multi-Objective Evolutionary Algorithm. **IEEE Access**, IEEE, v. 8, p. 46605–46623, 2020. ISSN 2169-3536. Available at: https://ieeexplore.ieee.org/abstract/document/9026933.

YAO, H.; SONG, Z.; CHEN, B.; LIU, L. ControlVAE: Model-Based Learning of Generative Controllers for Physics-Based Characters. **ACM Trans. Graph.**, Association for Computing Machinery, New York, NY, USA, v. 41, n. 6, nov. 2022. ISSN 0730-0301. Available at: https://doi.org/10.1145/3550454.3555434.

YAO, J.; HUANG, Q.; WANG, W. Adaptive CGFs Based on Grammatical Evolution. **Mathematical Problems in Engineering**, v. 2015, n. 1, p. 197306, 2015. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1155/2015/197306.

YAO, J.; HUANG, Q.; WANG, W. Adaptive Human Behavior Modeling for Air Combat Simulation. *In*: **2015 IEEE/ACM 19th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)**. **Proceedings** [...]. Chengdu, China: IEEE, 2015. p. 100–103. ISSN: 1550-6525. Available at: https://ieeexplore.ieee.org/abstract/document/7395921.

YAO, K. Study on the BVR Cooperative Air Combat Based on BP Neural Network. **Journal of Physics: Conference Series**, IOP Publishing, v. 1744, n. 4, p. 042171, feb 2021. Available at: https://dx.doi.org/10.1088/1742-6596/1744/4/042171.

YE, H.; LI, M.; YANG, Z. Research on the influence of time delay on combat effectiveness of aircraft fleet. **Journal of Physics: Conference Series**, v. 2658, n. 1, p. 012039, December 2023. Available at: https://doi.org/10.1088/1742-6596/2658/1/012039.

YIN, H.; LI, D.; WANG, Y.; LI, X. Adaptive Dynamic Occupancy Guidance for Air Combat of UAV. **Unmanned Systems**, p. 1–18, publisher: World Scientific Publishing Co., October 2022. ISSN 2301-3850. Available at: https://www.worldscientific.com/doi/abs/10.1142/S2301385024500031.

YOON, K. S.; PARK, J. H.; KIM, I. G.; RYU, K. S. New modeling algorithm for improving accuracy of weapon launch acceptability region. *In*: **Proceedings of the 29th Digital Avionics Systems Conference (DASC)**. **Proceedings** [...]. Salt Lake City, UT, USA: IEEE, 2010. p. 6.D.4–1–6.D.4–6.

YOU, H.; HAN, Q.; YU, M.; JI, H.; YE, Z.; ZHANG, X. A Method to Solve the Unreachable Zone of Mid-range Air-to-Air Missile. *In*: **2019 IEEE 2nd International Conference on Electronic Information and Communication Technology (ICEICT)**. **Proceedings** [...]. Harbin, China: IEEE, 2019. p. 649–654. Available at: https://ieeexplore.ieee.org/abstract/document/8846324.

YU, Y.; ZHU, Y.; LI, S.; WAN, D. Time Series Outlier Detection Based on Sliding Window Prediction. **Mathematical Problems in Engineering**, v. 2014, n. 1, p. 879736, 2014. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1155/2014/879736.

YUAN, G.; HU, L.; YONGLIANG, T. Inverse design of mission success space for combat aircraft contribution evaluation. **Chinese Journal of Aeronautics**, v. 33, n. 8, p. 2189–2203, 2020. ISSN 1000-9361. Available at: https://www.sciencedirect.com/science/article/pii/S1000936120302569.

YUAN, K.; LIU, D.; JIANG, D.; ZHANG, Z.; LEI, X. Design of Target Aircraft Auto Air-Combat Tactics Decision System. *In*: ZHANG, L.; SONG, X.; WU, Y. (Ed.). **Theory, Methodology, Tools and Applications for Modeling and Simulation of Complex Systems**. **Proceedings** [...]. Singapore: Springer Singapore, 2016. p. 288–296. ISBN 978-981-10-2666-9.

YUE, L.; XIAOHUI, Q.; XIAODONG, L.; QUNLI, X. Deep reinforcement learning and its application in autonomous fitting optimization for attack areas of UCAVs. **Journal of Systems Engineering and Electronics**, v. 31, n. 4, p. 734–742, 2020.

ZARE, M.; KEBRIA, P. M.; KHOSRAVI, A.; NAHAVANDI, S. **A Survey of Imitation Learning: Algorithms, Recent Developments, and Challenges**. 2023. Available at: https://arxiv.org/abs/2309.02473.

ZHANG, H.; WEI, Y.; ZHOU, H.; HUANG, C. Maneuver Decision-Making for Autonomous Air Combat Based on FRE-PPO. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 12, n. 20, p. 10230, publisher: Multidisciplinary Digital Publishing Institute, October 2022. ISSN 2076-3417. Available at: https://www.mdpi.com/2076-3417/12/20/10230.

ZHANG, H.; YU, C.; ZHANG, G.; WAN, Z. Research on High Dynamic Flight Simulation Test Technology. *In*: **AOPC 2021: Optical Sensing and Imaging Technology**. **Proceedings** [...]. Beijing, China: SPIE, 2021. v. 12065, p. 198–203. Available at: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/12065/120650V-/Research-on-high-dynamic-flight-simulation-test-technology/10.1117/12.2604719.full.

ZHANG, H.; ZHOU, H.; WEI, Y.; HUANG, C. Autonomous maneuver decision-making method based on reinforcement learning and Monte Carlo tree search. **Frontiers in Neurorobotics**, v. 16, 2022. ISSN 1662-5218. Available at: https://www.frontiersin.org/articles/10.3389/fnbot.2022.996412.

ZHANG, J.; HE, Y.; PENG, Y.; LI, G. Cooperative Path Planning for Adversarial Target based on Neural Network and Artificial Potential Field. *In*: **2018 IEEE CSAA Guidance, Navigation and Control Conference (CGNCC)**. **Proceedings** [...]. Xiamen, China: IEEE, 2018. p. 1–5. Available at: https://ieeexplore.ieee.org/abstract/document/9019020.

ZHANG, L. A.; XU, J.; GOLD, D.; HAGEN, J.; KOCHHAR, A. K.; LOHN, A. J.; OSOBA, O. A. **Air Dominance Through Machine Learning: A Preliminary Exploration of Artificial Intelligence-Assisted Mission Planning**. Santa Monica, CA: RAND Corporation, 2020.

ZHANG, T.; WANG, Y.; SUN, M.; CHEN, Z. Air combat maneuver decision based on deep reinforcement learning with auxiliary reward. **Neural Comput. Appl.**, Springer-Verlag, Berlin, Heidelberg, v. 36, n. 21, p. 13341–13356, abr. 2024. ISSN 0941-0643. Available at: https://doi.org/10.1007/s00521-024-09720-z.

ZHANG, Y.; LIU, Y.; GUO, Y.; WANG, X.; FENG, L.; LI, G.; GUO, Y. Research on Recognition Method of Maneuver and Formation Tactics in BVR Cooperative Combat Based on Dynamic Bayesian Network. *In*: SUN, J.; WANG, Y.; HUO, M.; XU, L. (Ed.). **Signal and Information Processing, Networking and Computers**. **Proceedings** [...]. Singapore: Springer Nature, 2023. (Lecture Notes in Electrical Engineering), p. 1365–1375. ISBN 978-981-19338-7-5.

ZHANG, Y.; ZHANG, W.; MA, Q.; GAO, D.; HUANG, X.; LI, H. Multi-target Optimized Cooperative Attack Task Allocation Technology. *In*: **2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID)**. **Proceedings** [...]. Guangzhou, China: IEEE, 2021. p. 690–695. Available at: https://ieeexplore.ieee.org/document/9456587.

ZHAO, L.; XING, C.; YANG, X. Research on Beyond-Visual-Range Air Combat Decision Making Based on Improved Self-Play Deep Reinforcement Learning. *In*: **Proceedings of the 2024 7th International Conference on Machine Learning and Natural Language Processing (MLNLP)**. **Proceedings** [...]. Chengdu, China: IEEE, 2024. p. 1–7. Available at: https://doi.org/10.1109/MLNLP63328.2024.10800451.

ZHAO, Y.; CHEN, Y.; ZHEN, Z.; JIANG, J. Multi-weapon multi-target assignment based on hybrid genetic algorithm in uncertain environment. **International Journal of Advanced Robotic Systems**, v. 17, n. 2, p. 1729881420905922, publisher: SAGE Publications, March 2020. ISSN 1729-8806. Available at: https://doi.org/10.1177/1729881420905922.

ZHOU, D.; PAN, Q.; ZHANG, K. An improved discrete Shuffled Frog Leaping Algorithm for cooperative Multi-Target assignment of BVR air combat. *In*: **2014 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)**. **Proceedings** [...]. Guilin, China: IEEE, 2014. p. 686–691. Available at: https://ieeexplore.ieee.org/abstract/document/6986283.

ZHOU, Y.; YANG, F. F.; ZHANG, C.; LI, S.; WANG, Y. Cooperative decision-making algorithm with efficient convergence for ucav formation in beyond-visual-range air combat based on multi-agent reinforcement learning. **Chinese Journal of Aeronautics**, v. 37, n. 8, p. 311–328, August 2024. ISSN 1000-9361. Available at: https://www.sciencedirect.com/science/article/pii/S1000936124001389.

ZOUJIE; WUWEI. Research of the network structure in beyond-visual-range coordinated air combat. *In*: **2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)**. **Proceedings** [...]. Nanjing, China: IEEE, 2016. p. 473–480.

ÖSTROM, J.; SAILARANTA, T.; VIRTANEN, K. Effects of datalink target data on air-to-air missile performance. **The Journal of Defense Modeling and Simulation**, p. 15485129231200026, publisher: SAGE Publications, September 2023. ISSN 1548-5129. Available at: https://doi.org/10.1177/15485129231200026.

FOLHA DE REGISTRO DO DOCUMENTO

[1.] CLASSIFICAÇÃO/TIPO	[2.] DATA	[3.] REGISTRO N°	[4.] N° DE PÁGINAS
TD	27 de junho de 2025	DCTA/ITA/TD-019/2025	345

[5.] TÍTULO E SUBTÍTULO:

Simulation and Machine Learning for Decision Support and Autonomy in Air Combat Operations

[6.] AUTOR(ES):

João Paulo de Andrade Dantas

7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES):

Instituto Tecnológico de Aeronáutica – ITA

[8.] PALAVRAS-CHAVE SUGERIDAS PELO AUTOR:

Air Combat; Autonomous Systems, Decision Support Systems; Machine Learning; Simulation

9.PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO:

Combate aéreo; Simulação; Aprendizado de máquina; Veículos autônomos; Sistema de apoio à decisão; Computação.

[10.] APRESENTAÇÃO: **(X) Nacional** **( ) Internacional**

ITA, São José dos Campos. Curso de Doutorado. Programa de Pós-Graduação em Engenharia Eletrônica e Computação. Área de Informática. Orientador: Prof. Dr. Takashi Yoneyama; coorientador: Prof. Dr. Marcos Ricardo Omena de Albuquerque Maximo. Defesa em 13/06/2025. Publicada em 27/06/2025.

[11.] RESUMO:

This thesis presents a research framework for integrating simulation and machine learning to support decision-making in air combat operations. Focused on Beyond Visual Range (BVR) air combat, which represents the modern modality of aerial warfare, the study explores five key areas: simulation tools and services, weapon systems, in-flight tactical decision-making, aerospace data analytics, and aerial autonomous agents. It reviews existing applications, identifies research gaps, and proposes targeted solutions across these domains. The contributions include the development of simulation tools and services tailored to defense scenarios; the modeling of weapon systems with real-time engagement zone prediction and kill probability estimation; the use of machine learning techniques to enhance in-flight tactical decision-making through improved missile launch timing, maneuver planning, and situational assessment; the application of data-driven approaches to support large-scale aerospace analytics; and the training of autonomous agents using imitation, generative, and reinforcement learning to perform complex air combat tasks. The proposed research framework is intended to support ongoing efforts to advance decision support and autonomy in modern air forces.

[12.] GRAU DE SIGILO:

(X) OSTENSIVO **( ) RESERVADO** **( ) SECRETO**