Linköping University | Department of Management and Engineering
Master's Thesis 30 ECTS | Division of Production Economics
Spring 2023

A Quantitative Framework for Technology Road-mapping

A Game-theoretic Approach for Technology Planning, Applied in a Military Setting

Authors | Lukas Peterson, Isak Berntsson
Supervisors | Mathias Henningsson†, Marina Rantanen Modéer‡
Examiner | Jörgen Blomvall†
ISRN | LIU-IEI-TEK-A–23/04694–SE
† | Linköping University
‡ | Saab AB

Linköping University
SE-581 83 Linköping
+46 13 28 10 00
www.liu.se

Copyright

Abstract

To identify critical technologies and make improved investment decisions, the technique of *technology road-mapping* is commonly employed. However, most methods used today rely on a qualitative approach, failing to capture the game-theoretic nature between competing actors and the stochasticity of technology development. This raises an important question: How can game theory be applied to model technological development quantitatively? The objective of this thesis is to develop a quantitative, game-theoretic framework that simulates technology investments while considering the stochasticity of scientific impacts. A minimax optimization model using LSS provided by Mazumdar et al. (2019), has been implemented to approximate equilibrium points between the actors. The estimation of the probability distribution describing the likelihood of a specific scientific impact has been performed by utilizing citation data from academic papers related to relevant technologies. The data were fitted to a log-normal curve using MLE.

The framework establishes a continuous two-player zero-sum game in which players compete to increase their probability of winning simulated military combat. Each player is provided with a technology portfolio consisting of various technologies available for investment. Investing in a technology leads to stochastic progress in that specific technology. The level of advancement in a player's technology portfolio determines their military capabilities, which are tested against each other on a simulated battlefield. This incentivizes players to make investments that increase (or avoid to decrease) their probability of winning. Upon completion of the game, various scenarios of accumulated technology investments will be available for analysis.

The results show that (i) the proposed method successfully uses game-theoretic concepts to model decisions of technology investments (ii) the estimated probability distributions follow the findings done by Radicchi et al. (2008), that citation data can be well-fitted to a log-normal PDF at a significance level of 5% (iii) the LSS algorithm developed by Mazumdar et al. (2019) show successful convergence towards (local) Nash equilibrium in the implemented game. From the results of the technology road map, it can be concluded that the potential of a quantitative framework, such as the one proposed in this study, is of great significance as an aid to qualitative methods that are used in the field today. With further work, the framework can serve as a valuable tool for incorporating a quantitative perspective into the decision-making process of allocating technology resources.

Keywords— Mathematical Modeling, Technology Road-Mapping, Game theory, Saddle point optimization, Principal component analysis, Maximum-Likelihood estimation

Acknowledgments

This thesis is a result of, and made possible by the assistance and support received from many people, including engineers at Saab Dynamics and faculty at Linköping University, whose impact we would like to pay attention and express our gratitude to.

Firstly, we would like to express our sincerest gratitude to our supervisor, Dr. Mathias Henningsson, for his invaluable guidance and support throughout this research project. His insightful feedback, positive reinforcements, and encouragement have been instrumental in the shaping of the thesis, especially during the early stages of the project. We also would like to thank Dr. Jörgen Blomwall, the examiner of this thesis, for his critical reading of our thesis and for providing constructive feedback that helped us to improve the quality and clarity of our work.

We would like to extend our heartfelt thanks to our external supervisor, Dr. Marina Rantanen Modéer from Saab Dynamics, for her valuable input, continuous support throughout all stages of this study and for sharing her expertise in the field of Tactical Autonomy. Her guidance, encouragement, and availability have been crucial in helping us to navigate the complexities of this topic.

Gratitude shall also be placed with Combitech who were very generous in allowing us to remotely run our simulations using their hardware. Without their help, this study would be infeasible and the scope of this thesis would have had to be limited even further.

Lukas Peterson, Isak Berntsson
Linköping, June 2023

Contents

Contents IV
List of Figures VII
List of Tables VIII
1 Introduction 1
1.1 Purpose 2
1.2 Delimitations 2
1.2.1 Technologies 3
1.2.2 The Game 3
1.2.3 Combat Model 3
1.3 Background 4
1.3.1 Tactical Autonomy 4
1.3.2 Quantitative Approach for Technology Road-mapping 5
1.3.3 Saab AB 5
2 Overview of Scientific Method 6
2.1 Literature Study 7
2.2 Required Data 7
2.3 Technology Readiness and Sigmoid Functions 8
2.4 Citation Distributions to Measure Scientific Impact 8
2.5 Mapping of Technologies to Military Capabilities 8
2.6 Evaluating Combat Effectiveness 9
3 Theoretical Background 10
3.1 Military Context 11
3.1.1 Tactical Autonomy 11
3.1.2 Technologies Within Tactical Autonomy 12
3.2 Elements of Probability Theory 13
3.2.1 Log-normal Distribution 14
3.2.2 Maximum-Likelihood Estimation 14
3.2.3 Monte Carlo-simulation 15
3.2.4 Kolmogorov–Smirnov Test 16
3.2.5 χ^2 Test 17
3.3 Combat Models 18
3.3.1 General Salvo Combat Model 18

3.3.2 Stochastic Salvo Combat Model 19
3.3.3 Sequential Stochastic Salvo Combat Model 21
3.4 Game Theory 21
3.4.1 Nash Equilibrium 22
3.4.2 Zero-sum Games 22
3.5 Sigmoid Functions 23
3.6 Unsupervised Learning Techniques 24
3.6.1 Principal Component Analysis 24
3.6.2 K-means Clustering 25
3.7 Optimization Methods 26
3.7.1 Minimax Optimization 26
3.8 Gradient Descent Ascent 26
3.8.1 Local Symplectic Surgery 27
4 Method **30**
4.1 General Dynamics of the Problem 30
4.2 Data Collection 32
4.2.1 Technology Identification 33
4.2.2 Citation Data for Research Progress Distributions 33
4.2.3 Survey for Conversion Matrix C and Starting State X_0 34
4.3 Parameter Estimation 35
4.3.1 Parameterization of Research Progress Distributions 36
4.3.2 Parameterization of the TRL Curve 37
4.3.3 Parameterization of the Conversion Matrix C 40
4.4 Constructing the Game 40
4.4.1 Combat Model 43
4.4.2 Choosing the Optimal Action 47
4.4.3 Updating the State 49
4.4.4 Evaluating the State 50
4.4.5 Mathematical Formulation 51
4.5 Running the Game 53
5 Results and Analysis **56**
5.1 Research Progress Distribution Parameters 56
5.2 Technology Road-map 57
5.3 Objective Function Values Throughout the Game 61
5.4 Principal Component and Clustering Analysis 62
5.5 Action Convergence 65
5.5.1 Inspection of Hessian Matrices 65

5.5.2 Inspecting Convergence and Step Sizes 67
6 Discussion **69**
6.1 Evaluation of Results 69
6.2 Evaluation of Methodology Choices 70
6.2.1 Modeling Research Progress 70
6.2.2 Constructing the Conversion Matrix C 72
6.2.3 Combat Model 73
6.2.4 Optimization Hyperparameters 74
6.2.5 Solution Methodology 75
6.2.6 Filter Actions, F_u 76
6.2.7 Implementation Environment 77
6.3 Future Research 77
6.4 Ethical Aspects 78
6.5 Conclusions 78

Appendix **84**
A The Silhouette Method **84**
B Queries for Citation Data **85**
C Survey for Parameterization **87**
D Figures of Results **98**
D.1 Research Progress Distributions 98
D.2 Technology Development TRL-curves 102
D.3 Trajectories 104

List of Figures

Figure 1	Overview of the scientific method adopted in the study	6
Figure 2	Overview of the theoretical framework adopted in the study	10
Figure 3	K-means algorithm	25
Figure 4	LSS algorithm	29
Figure 5	Representation of the spanned game tree	31
Figure 6	Data collection in method overview	32
Figure 7	Parameterization of research progress distributions in method overview	36
Figure 8	Parameterization of the TRL curve in method overview	37
Figure 9	TRL curve used for all studied technologies	39
Figure 10	parameterization of the conversation matrix C in method overview	40
Figure 11	Flowchart of the algorithm constructing the game	41
Figure 12	Visualization of the general process of the game	42
Figure 13	Combat model in method overview	43
Figure 14	No initiative, both sides fire simultaneously	45
Figure 15	Player A has initiative and fires first	45
Figure 16	Player B has initiative and fires first	46
Figure 17	Choosing the optimal action in method overview	47
Figure 18	Updating the state in method overview	49
Figure 19	Evaluating the state in method overview	50
Figure 20	Estimated probability distributions for research progress	56
Figure 21	Two examples of technological development obtained during the game	58
Figure 22	Average technology investments compared with weights in C	60
Figure 23	Sampled trajectories of states throughout the game	61
Figure 24	Objective function $f(\theta)$ values throughout the game	62
Figure 25	PCA of actions states and battle parameters	63
Figure 26	Explained variance of principal components	65
Figure 27	Hessian matrix during optimization	66
Figure 28	Gradient, step-length and objective function during optimization	68
Figure 29	Presentation of the fit of the estimated log-normal PDFs.	100
Figure 30	Evolution of TRL during the game	103
Figure 31	Presentations of trajectories	107

List of Tables

Table 1	Parameters in general Salvo combat model	19
Table 2	Parameters in stochastic Salvo Combat model	19
Table 3	List of identified technologies, constituting the technology portfolio	33
Table 4	Filtration used to collect citation data	34
Table 5	Numerical mapping of a technology's impact on a battle parameter	34
Table 6	Presentation of battle parameters in extended stochastic Salvo combat model	44
Table 7	Presentation of discretionary parameters in the game	53
Table 8	hyperparameters to run the game	54
Table 9	Specifications of PC used for simulation workload	55
Table 10	Results for the parameterization of f_ξ	57
Table 11	Presentation of results covering technological development	59

Notation

Abbreviations

AI	Artificial Intelligence
CDF	Cumulative density function
ECDF	Empirical cumulative distribution function
GA	Gradient ascent
GD	Gradient descent
GDA	Gradient descent ascent
KS	Kolmogorov – Smirnov
LASE	Locally asymptotically stable equilibria
LSS	Local symplectic surgery
MCS	Monte-Carlo Simulation
ML	Machine Learning
MLE	Maximum-Likelihood estimation
NE	Nash equilibrium
PC	Principal Component
PCA	Principal Component Analysis
PDF	Probability density function
SVD	Singular Value Decomposition
TA	Tactical autonomy
TRL	Technology readiness level
TRM	Technology road-mapping
UAV	Unmanned aerial vehicles
UGV	Unmanned ground vehicles

Variables, Parameters, and Sets

Technology Readiness Level

K	Slope of the TRL curve
D	Shift of the TRL curve

Research Progress Distribution

$\boldsymbol{f}_\xi$	Vector $\boldsymbol{f}_\xi \in \mathbb{R}^N$ of PDFs describing research progress
f_{ξ_n}	An element in $\boldsymbol{f}_\xi$ representing the PDF of research progress for technology n, defined by the log-normal function $f_\xi(\cdot)$
μ_ξ	Mean value of the log-normal distribution $f_\xi(\cdot)$
σ_ξ	Standard deviation of the log-normal distribution $f_\xi(\cdot)$
ξ	Random sample $\xi \sim \ln \mathcal{N}(\mu_\xi, \sigma_\xi)$ from the log-normal distribution $f_\xi(\cdot)$
c	Total number of citations that an article or conference paper has received
c_0	Average number of citations received by articles published in its research area, in the same year
c_f	Relative indicator $c_f = c/c_0$, which is standardized citation data by research area
d_n	The number of citation data points collected for research-area n

Combat Model

A, B	Beginning force strength (number of units)
n_α, n_β	Number of offensive actions per unit
p_α, p_β	Probability of offensive actions being accurate
n_y, n_z	Number of defensive actions per unit
p_y, p_z	Probability of defensive actions being successful
u, v	enemy damage sustained per successful action
w, x	Staying power per unit
μ_{net}	The mean of successful actions taken by a player
σ_{net}	The standard deviation of successful actions taken by a player
μ_{dam}	The mean of the units lost to each hit
σ_{dam}	The standard deviation of the units lost to each hit

μ_{nom}	The mean of the nominal number of remaining units of a player
σ_{nom}	The standard deviation of the nominal number of remaining units of a player
μ_{act}	The mean of remaining units after a Salvo fired by the other side
σ_{act}	The standard deviation of remaining units after a Salvo fired by the other side
K_3, D_3	Scaling factor of slope for calculations of $p_\alpha.p_\beta$
K_5, D_5	Scaling factor of shift for calculations of p_y, p_z
σ_h	The standard deviation of initiative distribution-PDF
c	Factor to scale sensitivity of initiative calculation

The Game

T	time horizon in the simulation of the game, where each time step is equivalent to 5 actual years
t	Value of time step in the simulation of the game, $t = 0 \ldots T$
P	Set of players $P = \{A, B\}$
p	Player index, $p \in P$
N	Number of technologies available to players
d	Player investment budget
n	Technology index, $n = 1 \ldots N$
M	Maximum number of optimization iterations
Θ	Number of Salvo combat model parameters
k	Salvo combat model parameters index $k \in 1 \ldots \Theta$
$\boldsymbol{\theta_i}$	Matrix of Salvo combat model parameters $\theta_i \in \mathbb{R}^{\Theta \times 2}$ in node i. Also denoted "Battle parameters".
$\theta_{k,p}$	An element in the battle parameters-matrix $\boldsymbol{\theta_i}$ for battle parameter k and player p
$\boldsymbol{\theta}^0$	Baseline parameters used to fix starting state parameters.
I	Total set of nodes in the tree that is spanned during one game
ω	Factor when mapping technologies to battle parameters
I_t	Set of nodes belonging to time step t
I_ξ	Number of random actions sampled from a hypercube with dimension N

I^*_{max}	Maximum number of actions allowed to take from each node i
$\boldsymbol{X}_i$	State matrix $\boldsymbol{X}_i \in \mathbb{R}^{N\times 2}$, representing the cumulative research progress done in technology n for player p, in node i
$x_{n,p}$	An element in state matrix $\boldsymbol{X}_i$ for technology n and player p
$\boldsymbol{u}^{\xi}_{i,j}$	Random action matrix $\boldsymbol{u}^{\xi}_{i,j} \in \mathbb{R}^{N\times 2}$, initiated from node i, with index j
$u^{\xi}_{n,p}$	An element in random action matrix $\boldsymbol{u}^{\xi}_{i,j}$ for technology n and player p
$\mathcal{U}^{\xi}_i$	Set of random actions initiated from node i
$\boldsymbol{u}^{\mathrm{NE}}_{i,j}$	Action matrix that constitutes a NE between the players $\boldsymbol{u}^{\mathrm{NE}}_{i,j} \in \mathbb{R}^{N\times 2}$, initiated from node i, with index j
$u^{\mathrm{NE}}_{n,p}$	An element in NE action matrix $\boldsymbol{u}^{\mathrm{NE}}_{i,j}$ for technology n and player p
$\mathcal{U}^{\mathrm{NE}}_i$	Set of NE actions initiated from node i
$\boldsymbol{u}^{*}_{i,j}$	Filtered NE-action matrix $\boldsymbol{u}^{*}_{i,j} \in \mathbb{R}^{N\times 2}$, initiated from node i, with index j
$u^{*}_{n,p}$	An element in the filtered action matrix $\boldsymbol{u}^{*}_{i,j}$ for technology n and player p
$\mathcal{U}^{*}_i$	Set of filtered NE actions initiated from node i
v	Limiting process of Local Symplectic surgery
v_0	Initial value of v
a_i	Step size scaling of u during optimization
b_i	Step size scaling of v during optimization
ϵ_1	Convergence criterion for maximum value in gradient vector
ϵ_2	Convergence criterion for maximum step length in u
k	Number of consecutive iterations that fulfill convergence criteria for the process to be considered converged.
C	Conversion Matrix from technologies to Salvo battle parameters $C \in \mathbb{R}^{\Theta\times N}$
$\boldsymbol{\xi}_i$	Random matrix $\boldsymbol{\xi}_i \in \mathbb{R}^{N\times 2}$ representing stochastic research progress during an action taken from node i
$\xi_{n,p}$	An element in the random matrix $\boldsymbol{\xi}_i$, representing a sample drawn from the technology-specific log-normal distribution $f_{\xi_n}(\cdot)$ for each technology n and player p

Q	Set describing the player with initiative in the battle function
A_j, B_j	Sample of surviving units after fired salvos in a simulated battle, for $j \in Q$
μ_{A_j}	Mean of sample $A_j \sim \mathcal{N}(\mu_{A_j}, \sigma_{A_j})$
σ_{A_j}	Standard deviation of sample $A_j \sim \mathcal{N}(\mu_{A_j}, \sigma_{A_j})$
μ_{B_j}	Mean of sample $B_j \sim \mathcal{N}(\mu_{B_j}, \sigma_{B_j})$
σ_{B_j}	Standard deviation of sample $B_j \sim \mathcal{N}(\mu_{B_j}, \sigma_{B_j})$
R	matrix that flips the positions of the first and second column in any two-column matrix
p_j	Probability that a player has an initial advantage over the other, $j \in Q$.

Functions and Operators

$\mathbb{1}_{\{\cdot\}}$	The indicator function
$f(\cdot)$	Objective function of the game that computes the probability of winning the simulated battle
$f_{\xi_n}(\cdot)$	PDF of research progress for research-area $n = 1 \dots N$
$f_{\text{net}}(\cdot)$	PDF of the distribution of the number of successful actions taken
$F_{\text{net}}(\cdot)$	CDF of the distribution of the number of successful actions taken
$f_{\text{nom}}(\cdot)$	PDF of the distribution of the nominal number of remaining units
$F_{\text{nom}}(\cdot)$	CDF of the distribution of the nominal number of remaining units
$h(\cdot)$	Joint PDF of a player having a significant advantage over the other
$S(\cdot)$	Scoring function to calculate probability of player A *winning*
$G_j(\cdot)$	Battle function to sample outcome of two-sided battle given initiative $j \in Q$
$\Lambda(\cdot)$	Salvo Function to calculate the parameters of the normal distribution of surviving units after receiving salvo
$F_u(\cdot)$	Filtration function that reduces the set of NE-actions $\mathcal{U}_i^{NE}$ to the set of actions taken by the players $\mathcal{U}_i^*$, $F_u : \mathcal{U}_i^{NE} \rightarrow \mathcal{U}_i^*$
$g(\cdot)$	TRL-function, one paradigm
$g_p(\cdot)$	TRL-function, two paradigm
$L(\cdot)$	Log-likelihood function
$n(\cdot)$	The cardinality of a set
$\vert\vert \cdot \vert\vert$	Some vector norm

1 Introduction

Technological development and the introduction of new innovations have a significant impact on organizational structure and increase competition between companies and countries across all industries (Carvalho et al., 2012). Technology road-mapping (TRM) is a widely used technique designed to identify critical technologies and technology gaps, enabling better technology investment decisions and facilitating strategic and long-range planning within an organization (Garcia & Bray, 1997; Phaal et al., 2004). However, most studies in the field rely on qualitative research methods, with relatively little exploration of quantitative research on the topic (Carvalho et al., 2012). This study aims to provide a quantitative framework for technology road-mapping, enhancing the strategic orientation of the technology planning process with a focus on competition and technological insight.

In every industry and around the world, both corporations and public institutions allocate a portion of their revenue to Research and Development (R&D) to deliver superior goods and services in the future. In highly competitive industries, R&D may take on a game-theoretic nature, where organizations carefully consider how their offerings will be positioned relative to their competition (Mcafee & Mcmillan, 1996). In this context, R&D efforts can be viewed as investments in a portfolio of emerging technologies, which may result in probabilistic progress according to some probability distribution. This study proposes a game-theoretic framework for technology road-mapping, where investments can be distributed across a portfolio of emerging technologies based on a player-defined utility function. This approach captures the probabilistic and uncertain nature of technological development and a company's competitive environment (I. G. Clark & Olds, 2005; Smirnova et al., 2018). Both players are considered rational and willing to adopt strategies that maximize their payoff in each round.

In this study, the general case of the proposed framework will be applied and evaluated in a military setting, where each player will be represented as a sovereign entity – investing in different technology areas in order to maximize its payoff on the battlefield. The payoff will be determined through a combat simulation and represented as the probability of defeating the opponent on the battlefield. The military industry is well suited for the application of this framework since it is highly driven by technological progress and has an inherent game-theoretic nature between different competing actors. Today, we are seeing an acceleration in technology development within the military industry, affecting both the capabilities of the actors pursuing the development and the security situation of the world (Försvarsdepartementet, 2017). It is therefore crucial for any military actor to identify, map and pursue emerging technologies that have the possibility to drastically change the conditions on the battlefield, in order to remain competitive (Försvarsdepartementet, 2017).

One area of technology that is of particular interest and is largely driven by commercial interests is that of autonomous systems and artificial intelligence (AI) (Försvarsdepartementet, 2017). In the Third Offset Strategy initiated by the U.S. Secretary of Defense, AI, unmanned vehicles, and machine learning were listed as core drivers to offset Chinese and Russian capabilities (Gentile et al., 2021). Russia is also heavily investing in the development and strengthening of its autonomous platforms (FOI, 2022). In Sweden, the Swedish Defense Research Agency (FOI) concludes that cyber, autonomous systems, and information technology (including AI) are central areas of interest for the Swedish Armed Forces to monitor, develop, and implement (FOI, 2022). This study is a collaboration with Saab AB and is initiated by a project within the field of tactical autonomy (TA).

1.1 Purpose

The purpose of this thesis is to develop a quantitative framework that captures the game-theoretic and stochastic nature of technology development in competitive and consolidated environments. The framework can then be used to simulate different scenarios and thus be helpful in the process of road-mapping future R&D projects. To further decompose the purpose of this study, the following research questions have been formulated:

- How can game theory be applied to quantitatively model technological development, in competitive and consolidated environments?
- How to estimate the probability distribution describing the probability of a specific research progress in a particular technology area, given the size of an investment in that technology area?
- How to find equilibria points in sequential, two-player zero-sum games, constructed in high-dimensional variable spaces?

1.2 Delimitations

To limit the scope of the study, a set of delimitations have been made and formulated under the following 3 categories: *technologies*, *the game*, and *combat model*. Furthermore, research concerning military technology is generally not publicly available, which limits the quality of the data collection and literature search.

1.2.1 Technologies

- The study will only focus on technology areas relevant to the field of tactical autonomy.
- The study will not take correlation and interdependence in technological development between different technology areas into consideration, meaning that progress in one technology does not result in progress in another.

1.2.2 The Game

- The game-theoretic framework will be modeled as a two-player game.
- The players have full knowledge of their own, and their opponents' state in each time step.
- Technology investments done by each player are assumed to be independent from year to year.
- Decisions are made to optimize the outcome for the next time step, and not over the entire time horizon.
- All technologies are assumed to follow the same universal TRL-curve, describing the evolution of the technological development.

1.2.3 Combat Model

- The combat simulation, used for measuring the players' capabilities against each other, will be considerably simplified. The simulation will be based on a Salvo model (Hughes & Wayne, 1995), situated in a marine setting.
- The combat model will not take the complexities of national economies and their ability to develop and produce military equipment during a conflict into consideration.

1.3 Background

The following section gives a brief introduction to the most important aspects needed to answer the purpose; Tactical autonomy, quantitative approaches for technology road-mapping, and Saab AB.

1.3.1 Tactical Autonomy

The notion of autonomy has different contexts, and as a result of recent years' increase in public interest, it has evolved significantly. Generally, autonomy in the context of intelligent systems focuses on developing intelligent decision-making systems that can physically operate autonomously with some degree of self-governance (Hagos & Rawat, 2022). More broadly, autonomous systems are defined as a network of intelligent systems capable of independently performing complex tasks, and making intelligent decisions without any explicit human intervention, or other operations management and control systems (Hagos & Rawat, 2022). Tactical autonomy is the concept of autonomous systems acting in support of tactical, short-term actions associated with a long-term strategic vision (AFRL, 2021).

Historically, technology, and scientific advancements have been essential in militarization. Used to gain an advantage over the opponent in a conflict, wartime advances have played a significant role in creating and commercializing much of the technology we use today. For example, following the creation of computing, nations knew they needed computing supremacy to remain a dominant and relevant force. This prompted a vast increase in technological funding, resulting in new technologies, capabilities, and prerequisites on the military scene. (Epstein, 2022).

Today, new and cutting-edge AI and ML techniques are increasingly used in the military and defense industry for a variety of different successful applications, such as training and simulation (Edwards, 2022), cyber security (Wirkuttis, 2017), logistics, transportation, and navigation (Eliaçık, 2022), protection of critical infrastructure (Dick et al., 2019), and many other domains of technical and strategical significance (Hagos & Rawat, 2022). The potential of AI systems in the military setting mainly revolves around the ability to collect and process large amounts of data and provide operational and strategic support with accelerating decision support. In a political and strategical context, AI systems can be used to destabilize enemies and defend against various forms of adversarial attacks in real-time (Hagos & Rawat, 2022). With the use of multi-domain input data, AI systems have the potential to assist in strategic decision-making at the top command level. At a tactical level, AI can provide a faster and more accurate situational awareness and thus reduce the vulnerability of unmanned vehicles (Hagos & Rawat, 2022). Additionally, it can also automate and provide sensitive threat detection able to interpret and identify suspicious and potentially dangerous activities, unrecognizable to human cognition (Hagos & Rawat, 2022).

Today, the military industry stands before a technological paradigm shift with the deployment of AI-based methods and autonomous systems. Industry experts are calling the use of AI the third revolution in warfare after gunpowder and nuclear weapons (Amyx, 2017). There are good reasons to believe that the actor with the upper hand in military robotics and autonomous systems in the future will have decisive advantages on the battlefield (FMV, 2022). It is therefore of great importance to identify, investigate and evaluate future technologies that can play a major role in the military domain, and to do so in a strategic and structured way in order to gain a competitive edge over competing actors by predicting potential technological development.

1.3.2 Quantitative Approach for Technology Road-mapping

TRM is a technique to identify critical technologies in order for corporations and public institutions to make improved investment decisions for R&D. The majority of frameworks for technology road-mapping that are being practiced, take a qualitative approach as research method (Carvalho et al., 2012). These methods are often based on case studies and may be limited by the subjective manner of the group issuing the empirical research. There is little evidence in the literature of quantitative methods being applied for technology road mapping. A quantitative framework for technology road-mapping could introduce a new approach to support more reliable decision-making in road-mapping processes and reduce underlying biases in the identification process. By introducing a game-theoretic approach to the decision-making of actors operating in competitive and consolidated markets, the interdependence between own and adversary actions can be captured.

1.3.3 Saab AB

Saab AB (Svenska Aeroplan Aktiebolaget) is a Swedish industrial and defense company, with a main focus on military technology, civil security, and aviation. The company was founded in 1937, and its main offices are located in Linköping, Sweden (Saab, 2023b). In 2022, Saab AB was the 35th largest arms manufacturer in the world, with a total defense revenue of $4,107.09 (Defense-News, 2022). In recent years, Saab AB has turned its focus towards emerging technologies, listing digitalization, autonomous systems, additive manufacturing, cyber and artificial intelligence as key focus areas in their future R&D-work (Saab, 2023a).

2 Overview of Scientific Method

This chapter provides the necessary concepts and data to achieve the research objectives. The approach consists of the following components: (i) data collection, encompassing both quantitative and qualitative methods; (ii) mathematical modeling of technological development; (iii) estimation of the probability distribution that describes the likelihood of research progress in a given technology area, based on the size of the investment; (iv) assessment of the extent to which a technology can enhance a military capability; and (v) accurate evaluation of combat effectiveness through numerical simulation of military capabilities in a two-player game. Additionally, the chapter reviews relevant literature on the topic and presents a visualization of the methodology adopted, as illustrated in figure 1.

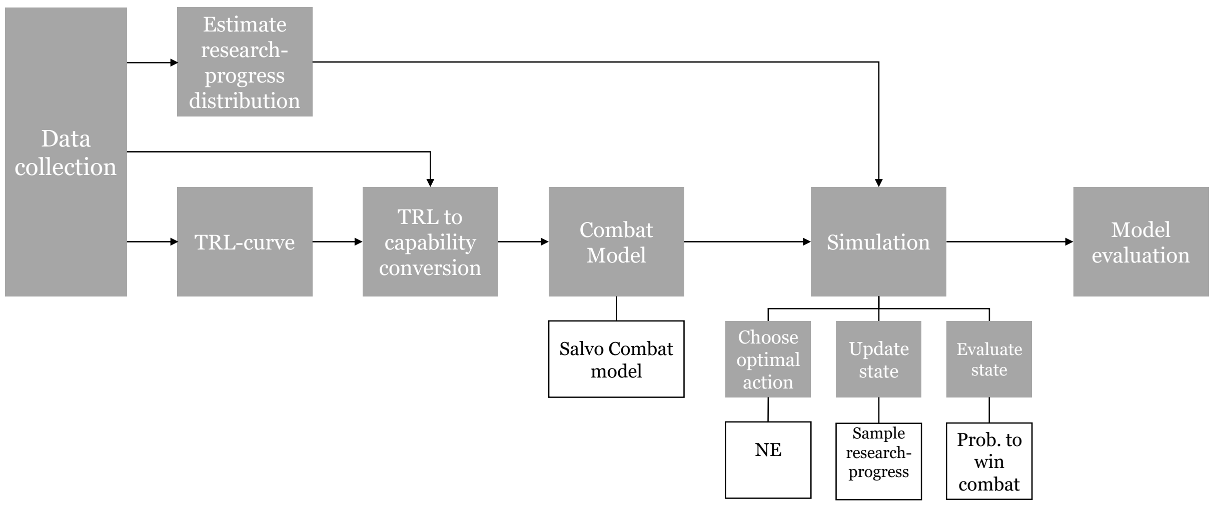


Figure 1: Overview of the scientific method adopted in the study

2.1 Literature Study

This thesis is part of a project within the field of TA at Saab AB. Therefore, a literature study will be conducted to accumulate knowledge and insights from previous work done in the field. The nature of this study implies that data can be highly sensitive and unreliable in the case of it being freely available. This means that much of the research that this study will cover within the field of TA has to be inferred from policy documents and studies available from government-funded research organizations such as the Swedish Defense Research Agency (FOI) and Defense Advanced Research Projects Agency (DARPA). Additionally, confidential and company-classified research performed internally by Saab may be provided by the external supervisor. Such data may be modified slightly in consultation with the external supervisor, in order to ensure publication of the study. The study will also cover various research concerning Technology readiness levels (TRLs), citation distributions, and optimization methods to locate equilibrium points in continuous, two-player zero-sum games, primarily focusing on scientific papers.

2.2 Required Data

The data collection in this study is categorized into two separate parts; a qualitative and a quantitative part. The qualitative data needed for the analysis mainly revolved around literature on autonomous functions in the military domain, and technological trends relevant to the defense industry. Additionally, a qualitative survey is conducted with the purpose of constructing a conversion matrix C to map technologies to parameters in a combat model, and to accurately set the starting state $\boldsymbol{X}_0$ describing the current readiness level of the technologies for each player. Furthermore, data is also provided by Saab-internal documents and through meetings with industry experts at Saab.

The quantitative data needed to carry out the analysis consist of citation data of articles within different scientific fields, in order to create probability distributions that model the probability of scientific impact within different fields of technology. These data sets are obtained through the Elsevier Scopus database, where the number of citations is counted as the total number of times an article appears as a reference of a more recently published article.

2.3 Technology Readiness and Sigmoid Functions

Technology readiness levels (TRLs) have been used by NASA since the 1970s as a method to measure the maturity of technology (Jim Banke & Administration, 2010). Each TRL represents the evolution of technology from the idea of thought to the full deployment of a product in a marketplace. Traditionally, TRLs are measured on a discrete scale between $1-9$ where a TRL value of 1 indicates that research is taken its first steps from idea to practicality, and a TRL value of 9 indicates that technology is incorporated fully into a larger system (Jim Banke & Administration, 2010). Smith (Smith II, 2005) introduces sigmoid (s-shaped) curves as an alternative way of modeling TRLs. By using sigmoid functions to model the TRL curve, the characteristics are changed from discrete to continuous, which enables differentiation. Jarne et al. (2007), also uses "s-shaped" curves to model different kinds of dynamic economic phenomena, such as economic growth and product life cycles.

2.4 Citation Distributions to Measure Scientific Impact

The first step to model stochastic progress in technology development is to find a probability distribution that accurately describes the probability of a specific progress, given a certain investment. One method closely related to this is presented in a model developed by Radicchi et al. (2008), where scientific impact is measured from a distribution of citations. Radicchi et al. (2008) fits citation observations to a log-normal curve that measures the success of a publication in its field.

2.5 Mapping of Technologies to Military Capabilities

To accurately describe the effect of a technology on a parameter in the combat model, Jouannet (2023) recommended letting experts in defense technology consider the degree to which a certain technology affects each military capability, representing the parameters in the combat model. The purpose of the survey is to conduct a qualitative collection of data in order to quantitatively parameterize a technology to a parameter in the combat model. The concept of mapping an actor's technologies, systems, and platforms to a set of capabilities is a well-studied area within the military domain. The Swedish Defense Research Agency (FOI) discusses the interconnection between technologies, capabilities, and military effect (FOI, 2022).

2.6 Evaluating Combat Effectiveness

As there is limited access to data and knowledge about how military forces interact with each other, a simple model is used in this study. While very advanced and modern battle simulation engines are available, Lucas and McGunnigle (2003) argues that simpler models often lead to better discussions and more insightful analyzes. Therefore, the Salvo model proposed by Hughes and Wayne (1995) is chosen as the basis for the discussion, which has been verified by applying it to historical situations with interesting results (Armstrong & Powell, 2005). To extend the Salvo model to include stochastic elements, Armstrong (2005) has developed a stochastic version, which is further adapted in this thesis to include an intelligence and information gathering parameter, allowing for sequential exchanges of fire.

3 Theoretical Background

This chapter provides a comprehensive introduction to the theoretical concepts employed in the study. Firstly, the military context in which the study is to be applied is introduced, offering an overview of tactical autonomy and relevant technologies in the field. The concept of combat models is also introduced, and the specific combat model used to simulate players' military capabilities against each other is explained. Additionally, important elements of probability theory are included, defining relevant terms and theorems for use in the study, alongside an introduction to key theoretical aspects of game theory, such as zero-sum games and Nash equilibrium. The theory of sigmoid functions is introduced and explained. Also, different unsupervised learning techniques are presented. Finally, the chapter lists and presents optimization methods relevant to the study.

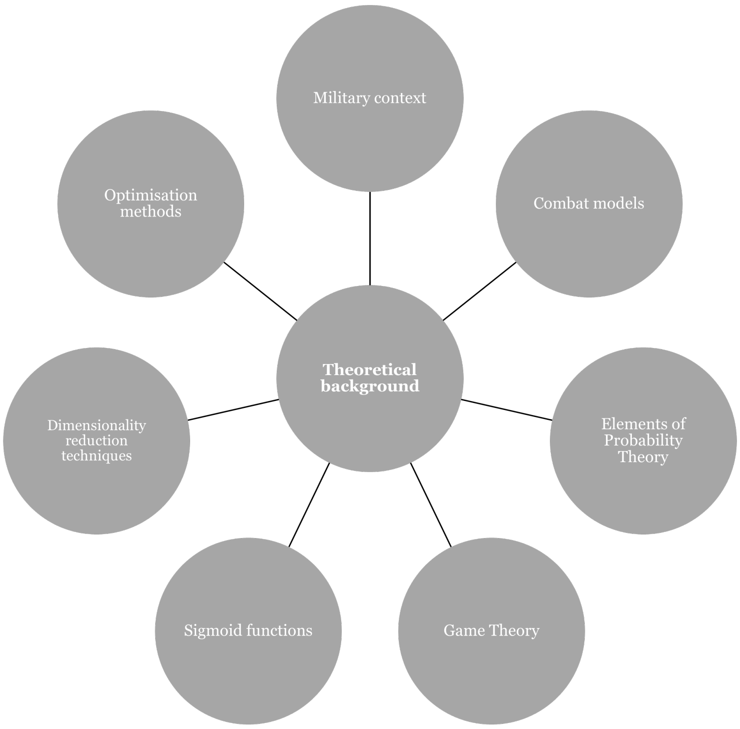


Figure 2: Overview of the theoretical framework adopted in the study

3.1 Military Context

This thesis aims to develop a framework that is intended to be applied in a military context. This section introduces important concepts relating to the military context the thesis is subject to, including *tactical autonomy* and *technologies within tactical autonomy*.

3.1.1 Tactical Autonomy

Tactical autonomy (TA) refers to the ability of a machine or system to operate independently in complex and dynamic environments without the need for constant human input or control (AFRL, 2021). This means that the machine can make decisions and take actions based on its own perception of the environment, its knowledge and understanding of the task at hand, and its ability to reason and adapt to changing circumstances. TA is a relatively new field, emerging in the late 20th century with the advent of unmanned aerial vehicles (UAVs) and other autonomous systems. In the past, machines were typically controlled by humans who provided input and guidance in real-time. However, as machines became more sophisticated and the demand of finding new ways to outmaneuver adversaries forced the development of disruptive innovations forward, researchers began exploring ways to enable greater autonomy and flexibility in their operation. (Hagos & Rawat, 2022)

Important concepts in TA include (i) perception, which refers to the ability of the machine to sense and interpret the environment using various sensors and data sources, such as cameras, lidar, and radar; (ii) reasoning, which refers to the ability of the machine to analyze and interpret the data it has collected and use that information to make decisions about what actions to take; (iii) decision making, which refers to the process of selecting the best course of action based on the available information; and (iv) action execution, which refers to the ability of the machine to carry out the chosen action. (Hagos & Rawat, 2022) In order to achieve tactical autonomy, machines, systems, and platforms must be equipped with advanced algorithms and software that enables (i)–(iv) in complex and dynamic environments. These algorithms often are heavenly reliant on machine learning and AI techniques, such as deep learning and reinforcement learning to enable the machine to learn from its experiences and improve its performance over time (Luotsinen et al., 2019).

TA is being used extensively in the military industry today. UAVs, colloquially known as drones, are the most prominent example of tactical autonomy in military applications. These UAVs can be remotely controlled by human operators, but they can also operate autonomously to carry out missions such as reconnaissance, surveillance, and target acquisition. Also, TA is being used in ground vehicles, such as unmanned ground vehicles (UGVs), to carry out tasks such as logistics, surveillance, and reconnaissance. These UGVs can operate in hazardous environments, such as minefields or areas contaminated by chemical or biological agents, without exposing human personnel to danger. Furthermore, TA is being used in marine applications as well, such as underwater vehicles, swarming surface drones, autonomous ships, and robotic systems for bomb disposal. (FMV, 2022)

Overall, TA is playing an increasingly important role in the military industry, as it enables machines to operate with greater efficiency, flexibility, and effectiveness in a wide range of scenarios. Autonomous systems have the potential to revolutionize the way the military operates, providing new capabilities and enhancing existing ones. By reducing the need for human personnel in dangerous situations, TA can help to save lives and reduce the risk of casualties.

3.1.2 Technologies Within Tactical Autonomy

Technology plays a critical role in the development of TA. Autonomous systems operating in a dynamic tactical environment require a range of capabilities that rely on advanced technical solutions, including for example sensing, computing, and actuation, in order to operate safely and effectively in different scenarios. The development of autonomous systems requires expertise across multiple scientific fields, such as computer science, electrical engineering, and mechanical engineering amongst others. Additionally, TA technologies raise important ethical and legal questions regarding the use of lethal force and accountability – raising the complexity of technological development within the field even more. The use of autonomous technologies in a military context can significantly improve operational efficiency, for example, enabling tasks to be completed more quickly and accurately without the risk of endangering human personnel (FOI, 2022). In addition, the development of new technologies in TA drives the creation of new capabilities on the battlefield that were previously impossible. One example of such a capability is the development of swarm robotics, which involves the use of a large number of autonomous robots working together to achieve a common goal, which in turn has been made possible by technological advances in sensing, computing, and communication (FMV, 2022).

Based on literature from FOI (FOI, 2022), FMV (FMV, 2022), and DARPA (Yang et al., 2019) (B. Clark et al., 2020), the most important technology areas withing TA include AI and ML, sensor technology, collaborative systems, integration to existing systems, safety and reliability, adaptability, navigation and localization, human-machine interaction, data fusion and processing, power and energy management, materials and manufacturing, cyber security, simulation and modeling, and ethics and regulations. Relevant trends in TA indicate the increasing use of AI and ML, which enables autonomous systems to learn and adapt to changing conditions more effectively and enhances the systems with more sophisticated decision-making processes. Another increasing trend is that autonomous systems are increasingly being designed to integrate multiple sensors, such as radars, lidars, and cameras, in order to gain complete awareness of the environment. Technological improvements in this area will enable autonomous systems to make better-informed decisions and operate more safely and effectively. Other trends include the development of swarming technologies, which are becoming more prevalent in TA. An example of this trend has been evident in the war between Ukraine and Russia, where swarming systems, such as drones, have been used by Ukrainian forces to target Russian artillery and military bases (NewsHour, 2023) (Center for Strategic and International Studies, 2023). Additionally, autonomous systems are predicted to be used more frequently in the logistics and supply chain management of tactical operations, where they can be used to optimize operations, relieve military personnel by transporting and carrying heavy equipment, and deliver material in dangerous and inaccessible environments (FMV, 2022). Lastly, an increasing emphasis on cyber security can be seen within the field of TA (FOI, 2022). As autonomous systems become more prevalent in military and security operations, there is increasing importance on cyber security to ensure that these systems are not vulnerable to cyber-attacks and malware.

3.2 Elements of Probability Theory

A random variable $X \in \mathbb{R}$ is a variable whose value is unknown and changes randomly, according to some probability distribution. A random variable X is a measurable function $X : \Omega \to E$ from a sample space Ω as a set of possible outcomes to a measurable space E. Similarly, multiple random variables together form a random vector $\boldsymbol{X} = (X_1, X_2, \dots, X_d)^T$. Random variables can be classified as discrete, which are variables that take specific values, or continuous, which are variables that can take values within a continuous range. Random variables are used in numerous scientific fields to model stochastic behavior and to predict outcomes of stochastic entities, as they give rise to probability distributions. Random variables, in this way, allow us to understand the world around us based on sample data. (Blom et al., 2016a)

A probability distribution is the mathematical description of the probabilities of events (subsets of the sample space, often denoted Ω). The sample space represents the set of all possible outcomes of a random phenomenon being observed; which can be any set: a set of real numbers, a set of vectors, or a set of any arbitrary non-numerical values. (Blom et al., 2016a)

Theorem 1 (**Central limit theorem**). Let $X_1, X_2, \ldots, X_n$ be a sequence of i.i.d. random variables drawn from a distribution with $\mathbb{E}[X_i] = \mu$ and $Var[X_i] = \sigma^2 < \infty$. Let $\bar{X_n}$ be the sample average. Then as n approaches infinity, the random variables $\sqrt{n}(\bar{X_n} - \mu)$ converge in distribution to a normal $\mathcal{N}(0, \sigma^2)$ as follows:

$$\sqrt{n}(\bar{X_n} - \mu) \quad \xrightarrow{d} \quad \mathcal{N}(0, \sigma^2)$$

3.2.1 Log-normal Distribution

A log-normal distribution is a continuous probability distribution that models the logarithm of a random variable as a normal distribution. This means that, if a random variable X is log-normally distributed, then $Y = \ln(X)$ has a normal distribution. This is equivalent to if Y has a normal distribution, then the exponential of Y, $X = \exp(Y)$, has a log-normal distribution. One property of a random variable which is log-normally distributed that is beneficial for modeling certain types of natural events, is that it only takes positive real values. The log-normal distribution is often used in finance, economics, and engineering to model variables that are inherently positive and have a skewed distribution.

Let X be a positive random variable and $\ln(X) \sim \mathcal{N}(\mu, \sigma^2)$, then the probability density function of X is defined as follows:

$$f_X(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\ln x - \mu)^2}{2\sigma^2}\right) \tag{3.1}$$

where μ is the mean and σ is the standard deviation.

3.2.2 Maximum-Likelihood Estimation

Maximum-Likelihood estimation (MLE) is a general method of estimating parameters of a probability distribution – a procedure that finds parameter values best fitted for the observed data (Myung, 2001). MLE is the most commonly used technique in statistical inference, holding many statistical benefits such as consistency (true parameter value that generated the data recovered asymptotically); sufficiency (complete information about the parameter of interest contained in its MLE estimator); efficiency (lowest-possible variance of parameter estimates achieved asymptotically) (Pan & Fang, 2002) (Myung, 2001).

The principle of MLE states that the desired probability distribution is the one that is "most likely" given the observed data, which is equivalent to finding the parameter vector that maximizes the likelihood function $L(\theta)$. Given a sample $x_i, \quad i = 1 \dots n$ from the stochastic variables $X_i, \quad i = 1 \dots n$ with the probability density function $f(x_1, x_2, \dots, x_n; \theta)$, where θ is an unknown parameter vector, the likelihood function is defined as

$$L(\theta) = f(x_1, x_2, \dots, x_n; \theta). \tag{3.2}$$

The estimate produced by the MLE is called θ^*, which maximizes

$$\theta^* = \underset{\theta}{\operatorname{argmax}} L(\theta). \tag{3.3}$$

The desired probability distribution for the assumed statistical model is then described by the following expression

$$f(u_n, u_{u-1}, \dots, u_1; \theta) = f(u_1; \theta) \prod_{i=2}^{n} f(u_i \mid u_{i-1}, \dots, u_2; \theta). \tag{3.4}$$

Since the logarithmic function is monotonically increasing, (3.4), can be formulated as

$$\ln L(\theta) = \sum_{i=1}^{n} \ln f(u_i \mid u_{i-1}, \dots, u_2; \theta) \tag{3.5}$$

By then maximizing (3.5), with respect to θ we get the Maximum-Likelihood estimation for the parameters, θ^* for the assumed statistical model.

3.2.3 Monte Carlo-simulation

In many applications, it is desirable to evaluate and estimate the mean value of some random entity, expressed in the form $\mu = \mathbb{E}[g(\mathbf{X})]$ for some function $g(\cdot)$ and a random vector $\mathbf{X} \in \mathbb{R}^d$. If the random vector follows some (continuous) probability function defined by the pdf f on the sample space $\Omega \subseteq \mathbb{R}^d$, the Law of the unconscious statistician (Loève, 1977) says that the mean can be calculated by:

$$\mu = \mathbb{E}[g(\mathbf{X})] = \int_{x \in \Omega} g(x) f(x) dx_1 \dots dx_d \tag{3.6}$$

A common problem one often finds when estimating the mean as stated in (3.6), is that such integrals often are complex and often lack an analytical solution. However, by the Strong Law of Large Numbers (Blom et al., 2016b), the following holds asymptotically:

Theorem 2 (**Strong Law of Large Numbers**)**.** If $\{z\}_{i=1}^N$ are i.i.d. observations from a random variable Z with $\mu = \mathbb{E}[Z]$, then:

$$\mu \overset{a.s}{=} \lim_{N \to +\infty} \frac{1}{N} \sum_{i=1}^{N} z_i$$

This implies that (3.6), can be numerically approximated with:

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} g(x_i) \approx \mu, \tag{3.7}$$

where $\{x_i\}_{i=1}^N$ is a sequence of numbers generated from the probability distribution defined by f, for a suitably large value of N.

3.2.4 Kolmogorov–Smirnov Test

The Kolmogorov–Smirnov test (KS test) is a non-parametric test of the equality of continuous, one-dimensional probability distributions, that can be used to compare a sample with a reference probability distribution (one-sample KS test) or to compare two samples and determine whether they are drawn from the same probability distribution (two-sample KS test). It does so by comparing the empirical cumulative distribution function (ECDF) of a sample with a theoretical cumulative distribution function (CDF). The test involves calculating the maximum difference between the ECDF of the sample and the CDF of the theoretical distribution. The maximum difference is then used as the test statistic in order to reject the null hypothesis; that the sample comes from the theoretical distribution.

Definition 1 (One-sample Kolmogorov–Smirnov statistic). Let F_n be the ECDF for n independent and identically distributed (i.i.d.) ordered observations X_i. Then,

$$F_n(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{(-\infty,x]}(X_i),$$

where $\mathbb{1}_{(-\infty,x]}(X_i)$ is the indicator function, equal to 1 if $X_i \leq x$ and equal to 0 otherwise. Let F be the CDF of a given theoretical distribution, and d_n be the Kolmogorov–Smirnov statistic. Then, d_n is defined by,

$$d_n = \sup_x |F_n(x) - F(x)|,$$

where $\sup_x$ is the supremum of the set of distances.

The p-value in the KS test represents the probability of obtaining a test statistic as extreme, or more extreme than the observed value, assuming the null hypothesis is true. More precisely, the p-value quantifies the strength of evidence against the null hypothesis. The p-value can be calculated using the Kolmogorov–Smirnov distribution, which gives the distribution of the KS statistic under the null hypothesis, or be obtained from a table. Let $c(\alpha)$ be the critical value of d_n for a significance level α. If $d_n > c(\alpha)$, reject the null hypothesis and conclude that the sample does not come from the theoretical distribution. If $d_n \leq c(\alpha)$ and if $p \leq \alpha$, reject the null hypothesis and conclude that the sample does not come from the theoretical distribution. Otherwise, fail to reject the null hypothesis.

3.2.5 χ^2 Test

The χ^2-test is a statistical test used to determine if a set of observed data fits a particular theoretical distribution. The test is based on comparing the observed frequencies of the data to the expected frequencies under the null hypothesis that the data follows the theoretical distribution.

Definition 2 (χ^2 statistic). Let $O_1, O_2, \ldots, O_k$ be observed frequencies of data with sample size N on k intervals, p_i the probability of the data falling in interval i under the theoretical distribution F, and $E_i = N \cdot p_i$ the expected frequency of interval i. Then, the χ^2 statistic is defined by,

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - E_i)^2}{E_i}$$

Let $c(\alpha, \nu)$ be the critical value of d_n for a significance level α and degrees of freedom $\nu = k - p - 1$, where p is the number of estimated parameters in the theoretical distribution. If $d_n > c(\alpha, \nu)$, reject the null hypothesis and conclude that the sample does not come from the theoretical distribution. If $d_n \leq c(\alpha)$ and if $p \leq \alpha$, reject the null hypothesis and conclude that the sample does not come from the theoretical distribution. Otherwise, fail to reject the null hypothesis.

3.3 Combat Models

Combat models aim to quantitatively model military combats in order to analyze modern combat characteristics, and how resources such as troops or fires evolve over time in simulated combat (Hughes & Wayne, 1995). This section presents various versions of the Salvo combat model, including *The Salvo model of modern missile combat* (Hughes & Wayne, 1995), *The stochastic Salvo model*, (Armstrong, 2005) and *The stochastic Salvo model with sequential fire* (Armstrong, 2014).

3.3.1 General Salvo Combat Model

The Salvo combat model was first introduced by Hughes and Wayne (1995), who argued for the need for a simple and interpretable model. In *The Salvo model of modern missile combat*, Hughes and Wayne (1995) presents a framework where the number of units lost by either side can be described as:

$$\nabla B = \frac{\alpha A - b_3 B}{b_1} \quad , \quad \nabla A = \frac{\beta B - a_3 A}{a_1}, \tag{3.8}$$

where the different parameters are presented in table 1 below:

Table 1: Presentation of the parameters used in general Salvo combat model, listing their symbol and description

Symbol	Description
A, B	The number of units in forces A and B
α, β	The number of well-aimed missiles fired by each unit in A and B
a_1, b_1	The number of hits needed to put a single unit in forces A and B out of action
a_3, b_3	The number of missiles destroyed by each unit of forces A and B
$\nabla A, \nabla B$	The number of units forces A and B lost to one opponent Salvo of missiles.

3.3.2 Stochastic Salvo Combat Model

Armstrong (2005) reworked the general Salvo combat model and introduced stochasticity, in order to capture the volatile nature of a battle. The model uses the parameters presented in table 2 below:

Table 2: Presentation of the parameters used in stochastic Salvo combat model, listing their symbol and description

Symbols, player A & B	Description
A, B	Beginning force strength (number of units)
n_α, n_β	Number of offensive actions per unit
p_α, p_β	Probability of offensive actions being accurate
n_y, n_z	Number of defensive actions per unit
p_y, p_z	Probability of defensive actions being successful
u, v	enemy damage sustained per successful action
w, x	Staying power per unit

With this model, the outcome is a stochastic variable that depends on the parameters mentioned above. The number of successful actions taken by the player follows a normal distribution with a mean of $\mu_{\text{net}_{\alpha,\beta}}$ and a variance of $\sigma^2_{\text{net}_{\alpha,\beta}}$ when player A is playing against player B.

$$\mu_{\text{net}_{\alpha,\beta}} = A \cdot n_\alpha \cdot p_\alpha - B \cdot n_z \cdot p_z \tag{3.9}$$

$$\sigma^2_{\text{net}_{\alpha,\beta}} = A \cdot n_\alpha \cdot p_\alpha \cdot (1 - p_\alpha) + B \cdot n_z \cdot p_z \cdot (1 - p_z) \tag{3.10}$$

Thus the distribution of the number of successful actions taken is $\mathcal{N}(\mu_{\text{net}_{\alpha,\beta}}, \sigma^2_{\text{net}_{\alpha,\beta}})$ with PDF f_{net} and the CDF F_{net}. The nominal number of remaining units of player B is is also normally distributed with mean μ_{nom_β} and variance $\sigma^2_{\text{nom}_\beta}$. Additionally, the expected number of units lost to each hit μ_{dam} is $\frac{u}{x}$ and the standard deviation σ_{dam} is assumed to be 0.5 as per Armstrong (2005).

$$\mu_{\text{nom}_\beta} = B - \mu_{\text{net}_{\alpha,\beta}} \cdot \mu_{\text{dam}} \tag{3.11}$$

$$\begin{aligned}\sigma^2_{\text{nom}_\beta} &= \mu_{\text{net}_{\alpha,\beta}} \cdot \sigma_{\text{dam}} + \sigma^2_{\text{net}_{\alpha,\beta}} \cdot \mu^2_{\text{dam}} - 2 \cdot \sigma^2_{\text{dam}} \cdot \mu_{\text{net}_{\alpha,\beta}} \cdot F_{\text{net}}(0) \\ &+ 2 \cdot \sigma^2_{\text{dam}} \cdot \sigma^2_{\text{net}_{\alpha,\beta}} \cdot f_{\text{net}}(0)\end{aligned} \tag{3.12}$$

Using the distribution of successful actions taken, as shown above, the nominal number of defenders can then be calculated. As the normal distribution is not bounded, the nominal distribution will contain a non-zero probability that the defender is left with a number of units smaller than 0 or larger than its initial value. Thus the normal distribution of the nominal number of remaining units is $\mathcal{N}(\mu_{\text{nom}_\beta}, \sigma^2_{\text{nom}_\beta})$ with PDF f_{nom} and CDF F_{nom}. The probability of player B surviving the Salvo can be computed using the CDF as $P_{\text{B lives}} = 1 - F_{\text{nom}}(\mu_{\text{dam}})$. To compensate for the fact that the number of hits must be a whole number and we approximate this using a continuous normal distribution F_{nom} evaluated with a correction factor μ_{dam}. (Armstrong, 2005).

Using the nominal distribution of surviving forces we can finally calculate the actual distribution such that it is limited to the range $[0 \ldots B]$. The parameters for the actual distribution μ_{act_β} and $\sigma^2_{\text{act}_\beta}$ are:

$$\begin{aligned}\mu_{\text{act}_\beta} &= \mu_{\text{nom}_\beta} \cdot \left(F_{\text{nom}}(B - \frac{\mu_{\text{dam}}}{2}) - F_{\text{nom}}(\frac{\mu_{\text{dam}}}{2})\right) \\ &- \sigma^2_{\text{nom}_\beta} \cdot \left(f_{\text{nom}}(B - \frac{\mu_{\text{dam}}}{2}) - f_{\text{nom}}(\frac{\mu_{\text{dam}}}{2})\right) \\ &+ B \cdot \left(1 - F_{\text{nom}}(B - \frac{\mu_{\text{dam}}}{2})\right)\end{aligned} \tag{3.13}$$

$$\begin{aligned}\sigma^2_{\text{act}_\beta} &= (\mu^2_{\text{nom}_\beta} + \sigma^2_{\text{nom}_\beta}) \cdot \left(F_{\text{nom}}(B - \frac{\mu_{\text{dam}}}{2}) - F_{\text{nom}}(\frac{\mu_{\text{dam}}}{2})\right) \\ &+ B^2 \cdot \left(1 - F_{\text{nom}}(B - \frac{\mu_{\text{dam}}}{2}) - \mu^2_{\text{act}_\beta}\right. \\ &- \sigma^2_{\text{nom}_\beta} \cdot ((B - \frac{\mu_{\text{dam}}}{2} + \mu_{\text{nom}_\beta}) \cdot f_{\text{nom}}\left(B - \frac{\mu_{\text{dam}}}{2}\right) \\ &\qquad - ((\frac{\mu_{\text{dam}}}{2} + \mu_{\text{nom}_\beta}) \cdot f_{\text{nom}}(\frac{\mu_{\text{dam}}}{2}))\Big)\end{aligned} \tag{3.14}$$

Equations (3.13), and (3.14), allows one to calculate the distribution of remaining units after a Salvo has been fired by the other side. A function Λ can be constructed such that $[\mu_{\text{act}_\beta}, \sigma^2_{\text{act}_\beta}] = \Lambda(\boldsymbol{\theta})$ where $\boldsymbol{\theta}$ is a matrix of the battle parameters as described in table 2. Note that it is possible to construct this distribution for the two players independently. This feature is taken advantage of when developing the sequential stochastic Salvo combat model.

3.3.3 Sequential Stochastic Salvo Combat Model

The sequential stochastic Salvo combat model differs from the stochastic Salvo combat model in one key way. The two sides do not fire at the same time but one player is instead assumed to have the initiative and is therefore able to fire on and damage their opponent before the opponent has a chance to return fire (Armstrong, 2014). By varying the order in which Λ is called and updating the values for A and B, several different scenarios of sequential battles can be analyzed.

3.4 Game Theory

Game theory is the applied maths branch established by John Von Neumann and John F. Nash, which is the study of mathematical models in conflict and cooperation between intelligent, rational, decision-makers (Najera, 2019). Game theory is the study of how interacting choices of economic agents produce outcomes with respect to the utilities of those agents, where the outcomes in question might not have been intended by any of the agents (Ross, 2021). To fundamentally grasp the idea of game theory, it is important to understand the concept of a *game* in the context of game theory. A *game* in game theory is any interaction between multiple players in which each player's payoff is affected by their decisions, and the decisions of others (Najera, 2019). Namely, any interaction between multiple actors can be analyzed with game theory in order to produce the most beneficial outcome.

To begin to understand how an economic agent makes decisions, and what they are basing decisions on, the concept of *utility* is introduced. An economic agent is, by definition, an entity with *preferences* (Ross, 2021). Economists and philosophers studying rational decision-making, describe these means of an abstract concept called *utility* (Ross, 2021). The utility can be explained as a ranking, on some specified scale, of the subjective welfare an agent receives from an event or object (Ross, 2021). Mathematically, this can be expressed by the acquired utility $U(x)$ of event $x \in \{x_1, x_2, \ldots, x_n\}$ taking place.

In game theory, players are assumed to have capacities that are typically referred to in the literature as comprising "rationality". The assumption that players are rational refers to a behavior where players manifest rational information processing: rational expectations are idealized beliefs that reflect statistically accurately weighted use of all information available to an agent (Ross, 2021). Von Neumann and Morgenstern (1966) argue that the motives of a rational player can be explained by the individual's desire to maximize its utility or satisfaction, and consequently act accordingly. For a player to be classified as an economically rational player, they must (i) possess the ability to evaluate outcomes by ranking them according to their impact on the player's welfare; (ii) they can determine the possible paths leading to these outcomes by identifying which sequences of actions are probabilistically associated with which outcomes; and (iii) they can choose the best possible course of action from sets of alternatives that will result in the most-preferred outcomes, taking into account the actions of other players involved (Ross, 2021).

3.4.1 Nash Equilibrium

The concept of a solution to a game in game theory is often referred to as an *equilibria*. When a system is said to be in equilibrium, it means that the system is in a *stable state*, one in which all the causal internal forces cancel out each other and leave it "at rest" until it is perturbed by an external force (Ross, 2021). Equivalently, economic equilibria are derived from the view of economic systems being networks of mutually constraining relations, where the equilibria of such systems are their endogenously stable states (Investopedia, 2023).

A "solution" to a game is called the *Nash equilibrium* (NE) of the game. NE applies or fails to apply, to whole sets of strategies – one for each player in a game, where a set of strategies is a NE if and only if no player could improve their payoff, given the strategies of all other players in the game, by changing their own strategy (Ross, 2021). NE was formulated by Nash (1951) with the following mathematical definition:

Theorem 3 (Nash equilibrium). Let S_i be the set of all possible strategies for player $i = 1 \dots N$. Let $s^* = (s_i^*, s_{-i}^*)$ be a set consisting of one strategy for each player, where s_i^* is the strategy for player i and s_{-i}^* denotes the $N-1$ strategies of all the players except i. Let $u_i(s_i^*, s_{-i}^*)$ be the utility function of the strategies for player i. Then, the strategy s^* is a NE if:

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*) \quad \forall s_i \in S_i$$

If the inequality in theorem 3 is strict, the strategy s^* is called a *strict* Nash equilibrium and is the unique best response.

3.4.2 Zero-sum Games

One special class of games where a Nash equilibrium (NE) is not only guaranteed but also *sufficient* as a solution concept, is *zero-sum* games (Ross, 2021). In a zero-sum game, the total utility, or payoff, earned by one player is equal and opposite to the utility earned by the other player. This means that any gain by one player is always accompanied by an equal loss by the other player, and the total sum of their payoffs is always zero. Because of this property, any strategy that one player chooses will necessarily affect the other player's payoff, and vice versa. Therefore, both players will always be incentivized to choose their best possible strategies given the other player's strategy, which is the definition of a NE.

One important property of zero-sum games is that the players' interests are directly opposed to each other. In other words, what is good for one player is bad for the other player. This creates a situation of pure conflict, where the players must "compete" against each other to maximize their own payoff. Because of this opposition of interests, zero-sum games are often used to model situations where there is a fixed amount of resources or value that must be divided between the players (Pettinger, 2023). Examples of zero-sum games include many two-player games such as chess, poker, and rock-paper-scissors.

3.5 Sigmoid Functions

A sigmoid function is a mathematical function with a characteristic "S"-shaped curve, that maps any input value to a value between 0 and 1. Because of this property, sigmoid functions are often used to model processes of growth, for example, knowledge, maturity, or readiness (Smith II, 2005). The growth dynamics of a sigmoid curve are at an initial stage approximately exponential; then, as saturation begins, the growth rate decelerates to linear, and at maturity, growth stops (Jarne et al., 2007). The most common example of a sigmoid function is the *logistic function* defined by

$$\frac{1}{1+\exp\left(\frac{-x}{a}+b\right)} \tag{3.15}$$

where a is the logistic growth rate, or steepness, of the curve, and b is the value of the function's midpoint or its shift. The logistics function can be used to illustrate the progress of innovation through its life cycle (Kuo-Pin Yang & Kuo, 2010). Furthermore, Rogers (2003) uses the logistic function to describe how, and at what rate innovations and new technology spread. Rogers (2003) argues that after the initial period of inertia when a new idea has been introduced, there is an intense amount of R&D and public interest, which leads to dramatic improvements in product quality and cost reduction. This leads to a period of fast industry growth and market traction. Once significant advancements and cost-cutting measures have been utilized, the product or process reaches extensive utilization with minimal prospective consumers, resulting in markets becoming saturated.

3.6 Unsupervised Learning Techniques

Unsupervised learning techniques are a set of machine learning methods that are used to find patterns and structures in unlabeled data. These techniques are used to identify hidden relationships and dependencies in data based on the inherent structure of the data, without any prior knowledge of the output. This section gives an introduction to two unsupervised learning techniques: *Principal Component Analysis (PCA)* and *K-means Clustering*.

3.6.1 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical method used to reduce the dimensionality of a dataset while retaining most of its variance. In other words, PCA finds a smaller set of uncorrelated variables, known as principal components, that explain the maximum amount of variation in the original dataset (Abdi & Williams, 2010). PCA analysis works by computing the eigenvectors and eigenvalues of the covariance matrix of the dataset. The eigenvectors represent the principal components, and the eigenvalues represent the amount of variance explained by each principal component. The first principal component accounts for the most variance in the data, the second principal component accounts for the second most variance, and so on.

Definition 3 (The Singular Value Decomposition). Every matrix $A \in \mathbb{R}^{m\times n}$ has a decomposition

$$A = U\Sigma V^T,$$

where U and V are orthogonal and $\Sigma \in \mathbb{R}^{m\times n}$ is *diagonal* with diagonal elements $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(n,m)} \geq 0$ called *singular values*. Let $A = U\Sigma V^T \in \mathbb{R}^{m\times n}$. Then

$$A^T A = V(\Sigma^T\Sigma)V^T, \quad \text{and} \quad U(\Sigma\Sigma^T)U^T.$$

So (σ_i^2, v_i) and (σ_i^2, u_i) are eigen pairs of $A^T A$ and AA^T.

Given a data matrix $X \in \mathbb{R}^{m\times n}$ and the centered data matrix $Z \in \mathbb{R}^{m\times n}$, obtained by subtracting X with the mean of X. By performing SVD on $Z = U\Sigma V^T$, the principal components of X are obtained through the column vectors of the matrix U, and the amount of variation explained by each principal component is determined by the singular values $\{\sigma_i\}_{i=1}^n$ in Σ. The transformed data $Y \in \mathbb{R}^{m\times n}$ is given by $Y = U\Sigma$, and is equivalent to the projection of Z onto the principal components.

3.6.2 K-means Clustering

K-means is an unsupervised machine learning algorithm used for clustering data into groups, first used by MacQueen (1967). The technique works by partitioning the data points into k clusters, where k is a user-defined parameter. The goal of the K-means algorithm is to minimize the within-cluster sum of squares, which measures the sum of squared Euclidean distances between each data point and its assigned centroid. This means that the algorithm is trying to find the best way to group the data points such that the distances between the points and their assigned centroids are as small as possible. Once the algorithm has converged, the resulting k clusters can be used to identify patterns in the dataset. The K-means algorithm is as follows:

Algorithm 1: K-means

Input : X, k, ϵ
$t = 0$
Randomly assign k centroids: $\mu_1^t, \mu_2^t, \ldots, \mu_k^t \in \mathbb{R}^d$
while $\sum_{i=1}^{k} ||\mu_i^t - \mu_i^{t-1}||^2 > \epsilon$ **do**
  $t \leftarrow t + 1$
  $C_j \leftarrow 0$ for all $j = 1, \ldots, k$
  for x_j *in* X **do**
    $j^* \leftarrow \text{argmin}_i\{||x_j - \mu_i^t||^2\}$ // `Assign` x_j `to closest centroid`
    $C_{j^*} \leftarrow C_{j^*} \cup \{x_j\}$
  end
  // `Centroid update step`
  for $i = 1, \ldots, k$ **do**
    $\mu_i^t \leftarrow \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$
  end
end
Output : $\{\mu_i^t\}_{i=1}^d$

Figure 3: K-means algorithm

One limitation of the K-means algorithm is the need to specify the number of clusters in advance. One method to determine the optimal number of clusters k is *The Silhouette method* (Shutaywi & Kachouie, 2021), described in appendix A.

3.7 Optimization Methods

Optimization methods are essential in various fields of study, including mathematics, engineering, economics, and computer science, as they allow for finding the optimal solution to a given problem. This section provides an introduction to several optimization methods used in this thesis, such as Minimax Optimization, Simultaneous Gradient Ascent, and Local Symplectic Surgery.

3.7.1 Minimax Optimization

Minimax optimization is the mathematical representation of finding equilibria in *sequential*, two-player zero-sum games, where a leader can commit to an action and a follower who responds after observing the leader's action (Wang et al., 2019). By the nature of a zero-sum game, the leader wants to maximize their reward and the follower wants to minimize, or vice-versa. Consider a differentiable sequential zero-sum game with two players: one with decision variable $\boldsymbol{x}$ and the other with decision variable $\boldsymbol{y}$. Then the minimax optimization formulation is:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \max_{\boldsymbol{y}\in\mathbb{R}^m} f(\boldsymbol{x}, \boldsymbol{y}). \tag{3.16}$$

A global solution to (3.16), is an action pair $(\boldsymbol{x}^*, \boldsymbol{y}^*)$, such that $\boldsymbol{x}^*$ is the global action for the leader, and $\boldsymbol{y}^*$ is the global response to $\boldsymbol{x}^*$ for the follower, assuming that both players always play the global optimal response (Wang et al., 2019). However, finding a global solution is often intractable due to high-dimensional parameter spaces and non-convexity; therefore a *local minimax* is taken as a surrogate.

Definition 4. $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is a local minimax for $f(\boldsymbol{x}, \boldsymbol{y})$ if $\boldsymbol{y}^*$ is a local maximum of $f(\boldsymbol{x}^*, \cdot)$, and $\boldsymbol{x}^*$ is a local minimum of $\phi(\boldsymbol{x}) := f(\boldsymbol{x}, r(\boldsymbol{x}))$, where $r(\boldsymbol{x})$ is the implicit function defined by $\nabla_y f(\boldsymbol{x}, \boldsymbol{y}) = 0$ in a neighborhood of $\boldsymbol{x}^*$ with $r(\boldsymbol{x}^*) = \boldsymbol{y}^*$.

$$\nabla f(\boldsymbol{x}, \boldsymbol{y}) = \begin{bmatrix} \nabla_x f \\ \nabla_y f \end{bmatrix}.$$

3.8 Gradient Descent Ascent

Gradient Descent Ascent (GDA) methods are the most common algorithms for minimax optimization (Li et al., 2022). For a minimax optimization problem on the form described in (3.16), GDA algorithms work by taking one step of gradient descent (GD) over the minimization variable $\boldsymbol{x}$, and another step of gradient ascent (GA) over the maximization variable $\boldsymbol{y}$. When the objective function is convex in $\boldsymbol{x}$, and concave in $\boldsymbol{y}$, Li et al. (2022) states that a lot of convergence results have been established by implementations of GDA. They also conclude that the algorithm, however, is not robust and easily converges to limit cycles or diverges even in the case of convex-concave functions.

3.8.1 Local Symplectic Surgery

Local symplectic surgery (LSS) is an optimization technique to find local Nash equilibria (and local Nash equilibria only) in continuous Zero-sum games (Mazumdar et al., 2019). The limitations that gradient-based optimization methods face in game-theoretic problems are that they are known to be difficult to tune and train and that they almost surely avoid a subset of the (local) Nash equilibrium (NE) in general-sum games (Mazumdar et al., 2019). Mazumdar et al. (2019) addresses a fundamental issue that arises when applying gradient-based methods in zero-sum games, that is, the set of attracting fixed points for the gradient dynamics in zero-sum games can include critical points that are not NE. In fact, any saddle point of the underlying function that does not satisfy a particular alignment condition of a NE is a candidate for attracting equilibrium for the gradient dynamics. Mazumdar et al. (2019) present a gradient-based method for finding (local) NE of two-player zero-sum games that only converge to stationary points that are (local) NE, which is called LSS.

Consider a two-player game, in which one player tries to minimize a function, $f : \mathbb{R}^d \to \mathbb{R}$, with respect to their decision variable $x \in \mathbb{R}^{d_x}$, and the other player aims to maximize f with respect to their decision variable $y \in \mathbb{R}^{d_y}$, where $d = d_y + d_x$. Such a game can be written as $\mathcal{G} = \{(f, -f), \mathbb{R}^d\}$. To simplify the notation, we let $z = (x, y)$, and define the vector-valued gradient function $\omega : \mathbb{R}^d \to \mathbb{R}^d$ as:

$$\omega(z) = J_f(z) = \begin{bmatrix} D_x f(x, y) \\ -D_y f(x, y) \end{bmatrix}. \tag{3.17}$$

Now, study the Jacobian matrix of w, defined by the Hessian matrix $H : \mathbb{R}^d \to \mathbb{R}^{d \times d}$:

$$H(z) = J(\omega(z)) = \begin{bmatrix} D^2_{xx} f(x, y) & D^2_{yx} f(x, y) \\ -D_{xy} f(x, y) & -D^2_{yy} f(x, y) \end{bmatrix}. \tag{3.18}$$

Definition 5. A strategy $z \in \mathbb{R}^d$ is a critical point of ω if $\omega(z) = 0$. It is a hyperbolic critical point if $Re(\lambda) \neq 0$ for all $\lambda \in eig(H(z^*))$, where $Re(\lambda)$, denotes the real part of the eigenvalue λ of $H(z^*)$.

As we are interested in finding (local) NE in the set of critical points, of particular interest are locally asymptotically stable equilibria (LASE) of the dynamics.

Definition 6. A strategy $z^* \in \mathbb{R}^d$ is a LASE of the continuous-time dynamics $\dot{z} = -\omega(z)$ if $\omega(z^*) = 0$ and $Re(\lambda) > 0$ for all $\lambda \in eig(H(z^*))$.

LASE possess a desirable characteristic where they attract locally and exponentially under the influence of $-\omega$. This means that if $z = \omega(z)$ is discretized properly, it will also converge exponentially fast in the vicinity of such points. By studying all LASE of a given continuous-time dynamical system, we can describe all the attracting hyperbolic equilibria. As shown in Ratliff et al. (2013) and Nagarajan and Kolter (2017), the fact that all differential NE are critical points of ω coupled with the structure of H in zero-sum games guarantees that all differential NE of the game is LASE of the gradient dynamics (Mazumdar et al., 2019).

To find LASE points of a zero-sum continuous game, Mazumdar et al. (2019) constructs a limiting differential equation with a continuous-time flow, and then formulates a discrete-time algorithm that follows that ODE. Consider the continuous-time flow:

$$\dot{z} = -h(z) = -\frac{1}{2}\left(\omega(z) + H^T(z)(H^T(z)H(z) + \lambda(z)I)^{-1}H^T(z)\omega(z)\right), \tag{3.19}$$

where $\lambda \in \mathcal{C}^2(\mathbb{R}^2, \mathbb{R})$ is such that $0 \leq \lambda(z) \leq \xi$ for all $z \in \mathbb{R}^d$ and $\xi > 0$ and $\lambda(z) = 0 \iff \omega(z) = 0$. In (3.19), the dynamics can be viewed as a modified form of the gradient dynamics. The adjustment term permits the trajectories to converge only towards the critical points of ω along the players' axes. When a critical point is not locally optimal for one of the players, meaning it is a non-Nash critical point, that player can influence the dynamics to move away from that point's vicinity (Mazumdar et al., 2019). To formulate a discrete-time algorithm solving the limiting ODE in (3.19), a straightforward Euler discretization could be performed on the following form:

$$z_{n+1} = z_n - \gamma h(z_n). \tag{3.20}$$

However, such an implementation would be prohibitively expensive in high-dimensional parameter spaces due to matrix inversions. To solve this computational problem, Mazumdar et al. (2019) introduces a two-timescale approximation to (3.19), given by:

$$
\begin{aligned}
z_{n+1} &= z_n - a_n h_1(z_n, v_n) \\
v_{n+1} &= v_n - b_n h_2(z_n, v_n),
\end{aligned}
$$

where h_1 and h_2 are defined as :

$$
\begin{aligned}
h_1(z, v) &= \frac{1}{2}(\omega(z) + H^T(z)v) \\
h_2(z, v) &= H^T(z)J(z)v - H^T(z)\omega(z) + \lambda(z)v,
\end{aligned}
$$

where the step-sizes a_n and b_n satisfy $\sum_{i=1}^{\infty} a_i^2 < \infty$, $\sum_{i=1}^{\infty} b_i^2 < \infty$, and $\lim_{n\to\infty} \frac{a_n}{b_n} = 0$. The v process performs gradient descent on a regularized version of least squares, where the regularization is governed by $\lambda(z)$. If the v_n process is on a faster time scale, the intuition is that it will first converge to $(H^T(z_n)H(z_n) + \lambda(z_n)I)^{-1}\omega(z_n)$, and z_n will track the limiting ODE in (3.19).

Algorithm 2: Local Symplectic Surgery

Input : Functions f, ω, H, λ; Step sizes a_n, b_n; Initial values (x_0, y_0, v_0)
Initialise : $(x, y, v, n) \leftarrow (x_0, y_0, v_0, 0)$
while *not converged* **do**
  $g_x \leftarrow [f(x, y) + \omega^T(x, y)v]$
  $g_y \leftarrow [-f(x, y) + \omega^T(x, y)v]$
  $g_v \leftarrow [||H(x, y)v - \omega(x, y)||_2^2 + \lambda(x, y)||v||_2^2]$
  $x \leftarrow x - a_n g_x$
  $y \leftarrow y - a_n g_y$
  $v \leftarrow v - b_n g_v$
  $n \leftarrow n + 1$
end
Output : $(x^*, y^*) \leftarrow (x, y)$

Figure 4: LSS algorithm

4 Method

This chapter provides a detailed description of the methodology used to address the research objectives, utilizing the theoretical concepts introduced in Chapter 3. Firstly, the general dynamics of the problem are introduced where the extensive approach to the problem is presented. Secondly, the chapter outlines the method for collecting and assessing data necessary to construct the technology portfolio, estimate the research progress distribution, and compose the conversion matrix C and starting state $\boldsymbol{X}_0$. After that, the parameterization of these concepts is explained and accounted for. The chapter then presents a comprehensive report on how the game is designed to generate various scenarios of technology development, including the design of the combat model, the process of determining an optimal action for technology investments, and how the state is updated and evaluated.

4.1 General Dynamics of the Problem

The aim of this work is to create a quantitative framework for technology road-mapping that captures the stochastic and game-theoretic nature of technology development. To achieve this, a continuous two-player zero-sum game is constructed where the players compete for the probability of winning simulated military combat. Each player has a technology portfolio comprising various technologies relating to autonomy they can invest in. An investment in a technology results in a stochastic progression in that technology, and the level of advancement in each technology determines the players' military capabilities, which are tested against each other on a simulated battlefield. Monte-Carlo simulating multiple battles allows for the calculation of the expected win-probability of the players, thus creating a feedback loop that incentivizes the players to invest in technologies that increase (or avoid decreasing) their probability of winning in the simulated battle given the other player's technology maturity.

The problem is therefore mainly focused on the discretization of the continuous solution space of possible technology investments and identifying possible trajectories of favorable decisions taken by the players. The methodology used to do this is to, for each state (constituted by the maturity of the technology portfolio), find the set of actions (investments) that constitutes a Nash equilibrium (NE) between the players. A NE is, by definition, a contract between players such that neither player is benefited by changing their current position, and can in the context of this game be identified as plausible scenarios of actions taken by the players. The dynamics of the discretization of the game result in the spanning of a graph over the states, with a depth equivalent to the time horizon of the simulation T, where each node in the graph corresponds to a NE between the players.

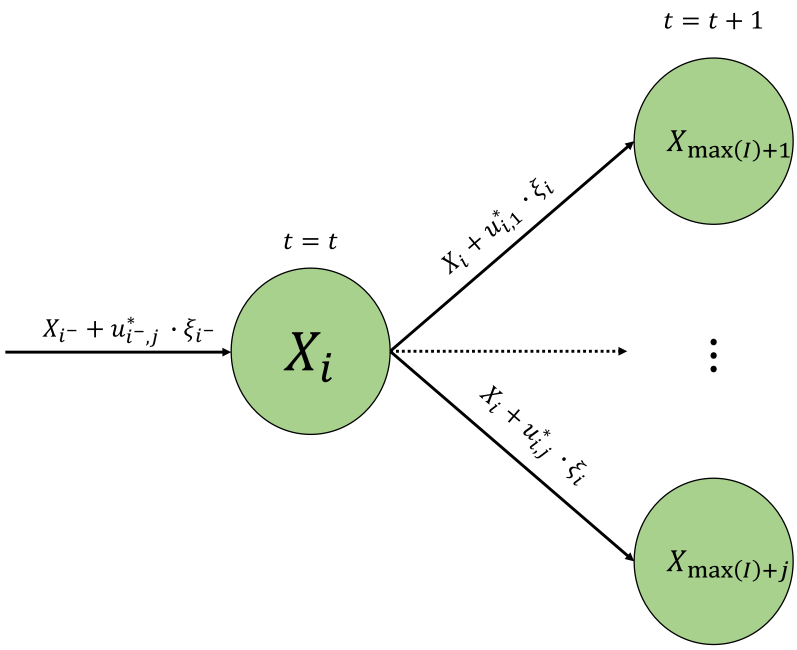


Figure 5: Representation of the tree that is spanned during the simulation of the game. Each node represents a state $\boldsymbol{X}_i$ which is a NE constituted between player A and B, I is the total set of nodes $i \in I$, i^- is the precursor to node i, and $j = 1 \ldots n(\mathcal{U}_i^*)$

The set of end-nodes $\boldsymbol{X}_i$, where $i \in I_T$, represents different possible scenarios of technology maturity after T time steps. By applying various data analysis techniques to the set, it is possible to identify trajectories of technology development that are more common than others. These trajectories can serve as a basis for further analysis and discussion.

4.2 Data Collection

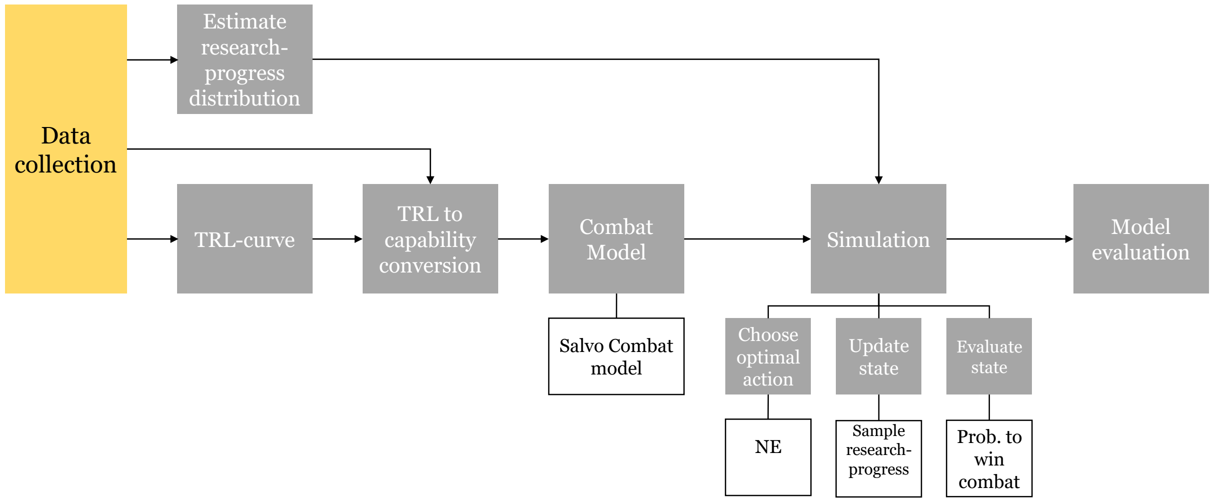


Figure 6: Data collection in method overview

In order to conduct the analysis for this study, both qualitative and quantitative data sets are required. Qualitative data is collected through a literature search to identify relevant technology areas, which are then used to construct a technology portfolio. Quantitative data is collected through citation data of articles in various scientific fields.

4.2.1 Technology Identification

The first fundamental step to achieve the purpose of this study is to identify relevant technology areas to construct a technology portfolio for the players' investments in the game. This is accomplished through an extensive literature search on autonomous functions in the military domain and technological trends in the defense industry.

Since data in this domain is highly sensitive and often classified, data collection has primarily been inferred from studies conducted by government-funded agencies such as the Swedish Defense Research Agency (FOI) and Defense Advanced Research Projects Agency (DARPA). A primary source of information on technology identification is a report called "Militärteknik 2045" published by FOI (FOI, 2022), which aims to predict military technology trends until 2045.

Table 3: List of 14 identified technologies, constituting the technology portfolio

n	**Technology area**
1	AI & Machine Learning
2	Collaborative systems
3	Communications & Networking
4	Control systems & Algorithms
5	Cyber security
6	Edge computing
7	Energy Management
8	Ethics & Regulations
9	Human Machine Interaction
10	Localization & Mapping
11	Sensor Fusion
12	Sensor Technologies
13	Simulation & Modeling
14	Tech. enabling mobility

In total, 14 technology areas have been identified that together constitute the technology portfolio, listed in table 3. The technology areas have been formulated to cover as much of the field of TA as possible while minimizing the overlap between individual technologies.

4.2.2 Citation Data for Research Progress Distributions

The second step in answering the purpose of this study involves estimating the probability distribution that predicts the expected research progress for a given technology, given a certain investment. To achieve this, citation data for articles and conference papers were obtained from Elsevier's Scopus, a database of academic literature. The process involved entering several queries to find relevant articles for each technology area in the technology portfolio (listed in table 3), in order to create citation distributions. To narrow down the search results and extract relevant data for each technology, filters were applied including document type (articles and conference papers only) and publication year. The Scopus database has a limit of 20,000 data points, which required varying the publication years across the technology areas to obtain as many data points as possible below this limit. The years have been varied to be after 2000 and before 2020, in order to maintain relevant data sets. Table 4 presents the filtration used for each technology area, and appendix B lists the specific queries used.

Table 4: Presentation of the filtration used to collect citation data for research area $n = 1 \dots N$. The data was filtered on document type and the years published

Research area, n	**Document**	**Year**	**Data points,** d_n
AI & Machine Learning	Conference Paper & Article	2001-2019	9635
Collaborative Systems	Conference Paper & Article	2006-2019	19844
Communications & Networking	Conference Paper & Article	2001-2019	8946
Control Systems & Algorithms	Conference Paper & Article	2006-2019	17071
Cyber Security	Conference Paper & Article	2013-2018	19748
Edge Computing	Conference Paper & Article	2001-2019	5359
Energy Management	Conference Paper & Article	2001-2019	6199
Ethics & Regulations	Conference Paper & Article	2006-2018	16323
Human Machine Interaction	Conference Paper & Article	2006-2018	16577
Localization & Mapping	Conference Paper & Article	2006-2018	14071
Sensor Fusion	Conference Paper & Article	2011-2018	19430
Sensor Technology	Conference Paper & Article	2001-2019	8661
Simulations & Modeling	Conference Paper & Article	2011-2017	16985
Technology enabling mobility	Conference Paper & Article	2001-2019	4291

4.2.3 Survey for Conversion Matrix C and Starting State X_0

To accurately describe the impact of a technology on a parameter in the combat model, and obtain the current degree of maturity for each technology, a qualitative survey was conducted. Experts in the field were asked to rate the extent to which each technology affects each military capability, representing the parameters in the combat model. They were also asked to place each technology on the TRL curve to determine the starting state $\boldsymbol{X}_0$ and assess where it is located in its current technological paradigm. The survey segmented the TRL curve into 5 different maturity levels, each assigned a numerical value from 1–5. The methodology used to design the survey is based on the *Impact Critically Assessment Scale* adapted from the Canadian Department of National Defense (Olivier, 2011), where the impact of a risk is rated on a 3 -grade

Table 5: Numerical mapping of the rating of a technology's impact on a parameter in the combat model, based on recommendations from Jouannet (2023)

Rating	Numerical value
N/A	0
Minor	1
Major	3
Critical	9

scale from Minor to Critical. Each rating is represented by a numerical value in order to quantitatively parameterize the technologies to battle parameters. Table 5 shows the numerical mapping used.

The exponential scale on the numerical values in table 5 is based on recommendations from Jouannet (2023), stating that the benefit of implementing exponential differences between the lowest level and the highest is to more accurately capture real-world dependencies than a linear scale would have done. The survey used to collect the data needed to conduct the parameterization of technologies to battle parameters is presented in appendix C.

To collect data of high quality, Jouannet (2023) emphasizes the need for high pedagogy level when describing the technologies and the capabilities they will induce, and also to clearly specify the scenario in which they are intended to be applied. This is done by clearly stating the purpose of the survey in the description.

4.3 Parameter Estimation

With the collected data, the parameters inherent to the different models used in the study are to be estimated. In this section, the methods for the parameterization of the *research progress distribution*, the *TRL curve*, and the *Conversation matrix, C*, are described.

4.3.1 Parameterization of Research Progress Distributions

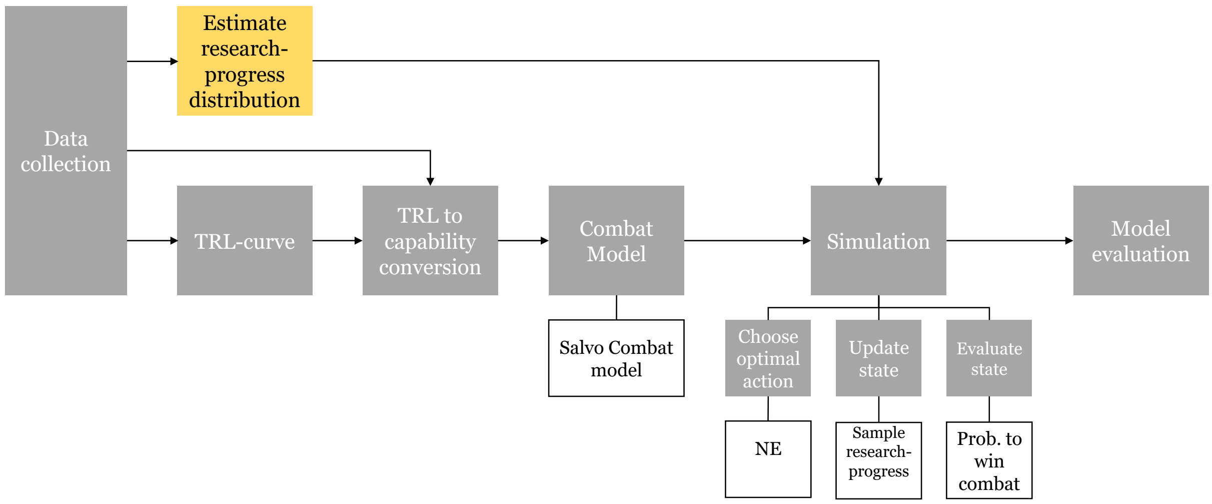


Figure 7: Parameterization of research progress distributions in method overview

The first step in estimating the research progress distribution is to standardize the citation data for each research area listed in table 4 to factor out the dependency between the chance of a publication being cited and the field it belongs to (Radicchi et al., 2008). This is achieved by introducing a *relative indicator* defined in Radicchi et al. (2008) by the ratio $c_f = c/c_0$ between the number of citations received c and the mean number of citations received by articles receiving at least one citation and published in the same field in the same year c_0. For the purposes of this thesis, slight changes have been made to the construction of c_0 which is now defined as the median number of citations received. This was done in order to restrict the value of the standard deviation σ during the estimation. The method developed by Radicchi et al. (2008) shows that the distribution of the relative indicator c_f follows a log-normal distribution, defined by the function:

$$f_\xi(c_f) = \frac{1}{\sigma_\xi c_f \sqrt{2\pi}} e^{-[\ln(c_f)-\mu_\xi]^2/2\sigma_\xi^2} \tag{4.1}$$

where $c_f \sim \ln\mathcal{N}(\mu_\xi, \sigma_\xi)$. The next step in order to estimate the research progress distribution is to determine the parameters μ_ξ and σ_ξ, which was done by a Maximum-Likelihood estimation described in section 3.2.2, for each research-area $n = 1 \ldots N$. The Maximum-Likelihood estimation was done with an implementation in Python using optimization methods from the scipy-package (Virtanen et al., 2020), solving (4.2) below for all values of $c_f \in [.1, 10]$,

$$\mu_{\xi_n}, \sigma_{\xi_n} = \underset{\mu,\sigma}{\operatorname{argmax}}\{\ln(L(\mu,\sigma))\} = \underset{\mu,\sigma}{\operatorname{argmax}}\Big\{\sum_{i=1}^{d_n} \ln\big(f_\xi(c_{f_i}|\mu,\sigma)\big)\Big\} \quad \forall n = 1 \ldots N, \tag{4.2}$$

where L is the log-likelihood function and d_n is the number of citation data points for research-area $n = 1 \ldots N$. This results in the construction of the vector $\boldsymbol{f}_\xi = \{f_{\xi_n}\}_{n=1}^{N}$ containing each technology-specific log-normal distribution with mean μ_{ξ_n} and standard deviation σ_{ξ_n}, for $n = 1 \ldots N$. The fit of the estimated curves is then evaluated using a KS-test, described in section 3.2.4, and a χ^2-test, described in section 3.2.5.

4.3.2 Parameterization of the TRL Curve

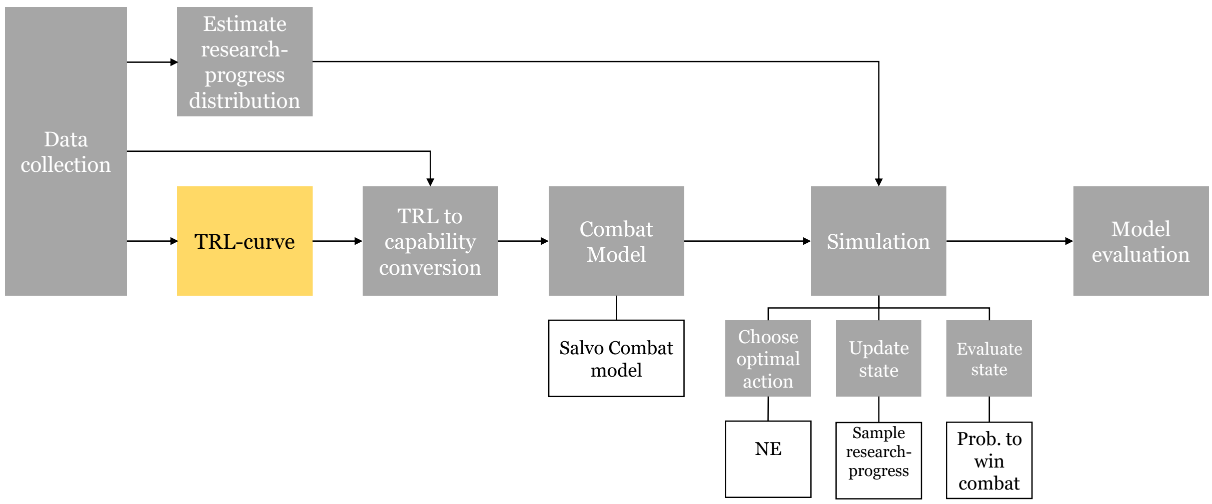


Figure 8: Parameterization of the TRL curve in method overview

The curve that describes a technology's readiness in its current paradigm is a generalized logistic sigmoid function, which is described in section 3.5 and presented in (3.15). In the context of technology readiness, the parameters a and b in (3.15) are assigned the values of K and D, respectively, which describe the slope and shift of the curve. For a given paradigm, the TRL curve is described by:

$$g(x) = \frac{1}{1 + \exp\left(\frac{-x}{K} + D\right)} \tag{4.3}$$

where x is the cumulative progress made up until that time. To enable the model to predict technologies transitioning to a new paradigm, the TRL curve in (4.3), is extended to a sum of two sigmoid functions described in (4.4) below. The choice of designing the TRL curve with two paradigms was to enable technologies which currently were considered mature, to evolve. A transition from one technological paradigm to another corresponds to crossing a saddle point, which is an area of the TRL curve that results in very small gradient values. In other words, major investments is needed in order for a player to make that happen. The reason of only enabling the TRL curve with one paradigm-transition is to maintain the level of abstraction at an intuitive level.

$$g_p(x) = \frac{1}{1 + \exp\left(\frac{-x}{K} + D\right)} + \frac{1}{1 + \exp\left(\frac{-x}{K} + 3D\right)}. \tag{4.4}$$

To prevent the number of parameters in the model from growing too large, one TRL curve is assumed for all technologies, with parameter values $K = 0.5$ and $D = 5$. The values of K and D are assigned with values in order to harmonize with the budget parameter $d = [1, 1]$ and the magnitude of a sample of research progress ξ, such that one technological paradigm is roughly equivalent to a cumulative progress value of $x = 5$, resulting in the appearance visible in figure 9.

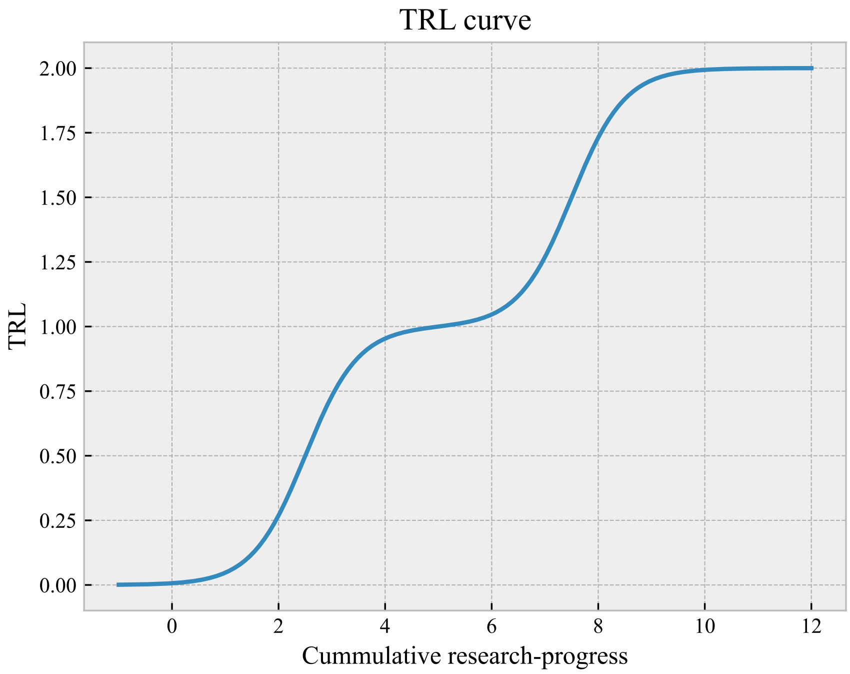


Figure 9: TRL curve used for all studied technologies. The curve describes how the usefulness (TRL) of technology improves throughout the two paradigms as research progresses.

4.3.3 Parameterization of the Conversion Matrix C

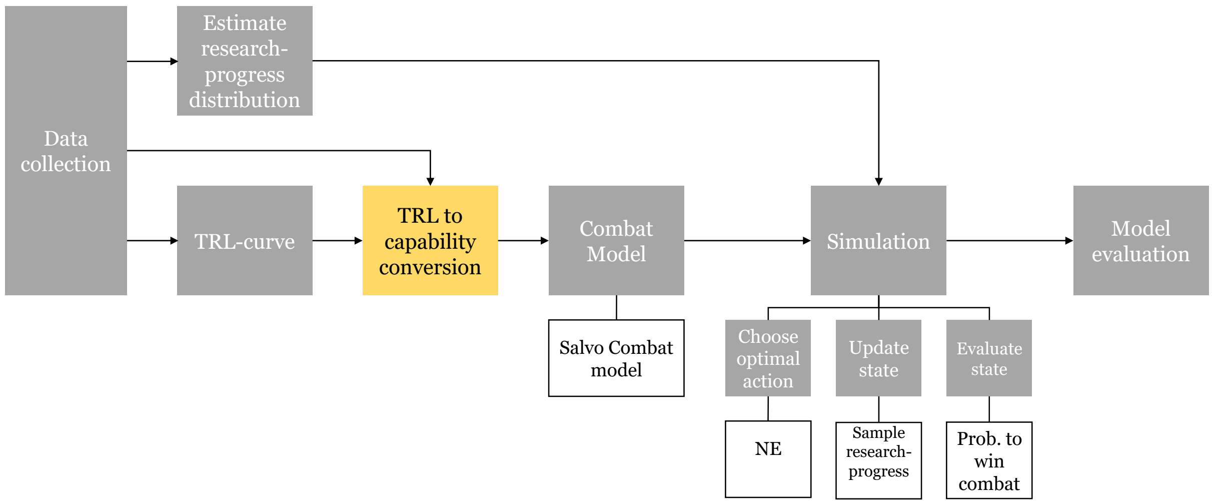


Figure 10: parameterization of the conversation matrix C in method overview

With the data collected through the survey described in section 4.2.3, the conversion matrix $C \in \mathbb{R}^{N\times\Theta}$ is constructed by taking the mode of the set of answers for each question, regarding the degree of impact each technology (see table 3) has on every battle parameter (see table 6). Where mode is the most common answer. The starting state $\boldsymbol{X}_0 \in \mathbb{R}^{N\times 1}$ is obtained in the same way as the conversion matrix, by taking the mode of the set of answers for each question regarding the starting state for each technology.

4.4 Constructing the Game

The core idea of this study is to construct a game in which two opposing players aim to optimize their respective probability of winning in battle by placing investments in different technologies relevant to military capabilities A flowchart of the game can be seen below in figure 11:

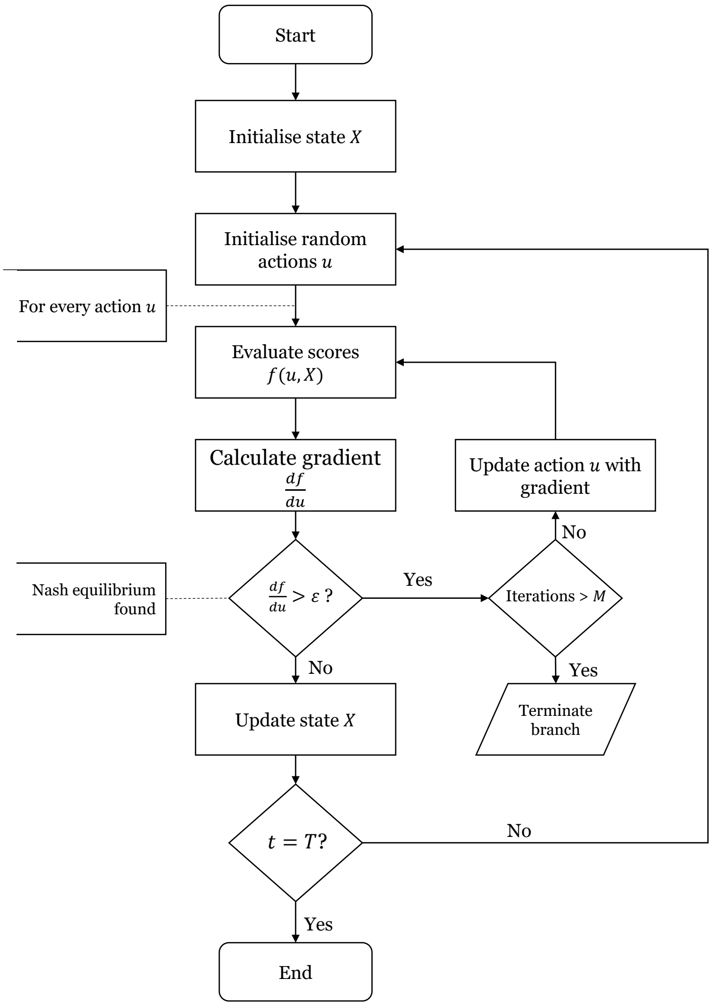


Figure 11: Flowchart of the algorithm constructing the game. Note that the inner loop can be run in parallel for multiple random actions and that running the algorithm results in a tree of states, actions, and scores.

The general concept of the game is as follows: the initial state of the game, $\boldsymbol{X}_0 \in \mathbb{R}^{N\times 2}$, is the current real-world state of cumulative research progress, for all technologies $n = 1 \ldots N$. From this state, a number of random actions $I_\xi = n(\mathcal{U}_i^\xi)$ (representing investments) are generated and constitutes the set of random actions $\mathcal{U}_i^\xi$ generated from node i. The score is then evaluated for each of these actions. A gradient-based iterative optimization method called LSS, described in section 3.8.1, is then applied to vary the actions of both players until the new combined action of both players constitutes a (local) Nash equilibrium (NE). This process is expected to yield several (local) NE, creating the set $\mathcal{U}_i^{NE}$ containing actions that constitutes (local) NE from node i. The filtration function $F_u : \mathcal{U}_i^{NE} \to \mathcal{U}_i^*$, which performs K-means (see algorithm 1) on the principal components of the vectors in $\mathcal{U}_i^{NE}$ is then applied to these actions to construct the set of actions $\mathcal{U}_i^*$. This set contains a smaller number of actions that are representative of the set $\mathcal{U}_i^{NE}$ as a whole. Each of the actions are then used to independently update the state and thus construct several possible future states, in which the search for new actions is performed again. As each state gives rise to several new states in the immediately subsequent time step, the number of states in each time step grows exponentially and and states can naturally be arranged as nodes in a non-recombining tree. This process of generating random actions, having them converge to (local) Nash equilibria using LSS, creating a representative subset and updating the current state with the each of the actions in the subset, (thus yielding several possible states in the following state,) repeats until the time horizon T of this game has been reached, and a large number of possible trajectories have been generated and can be analyzed. This process is visualized in figure 12 below, and explained in detail i the following sections.

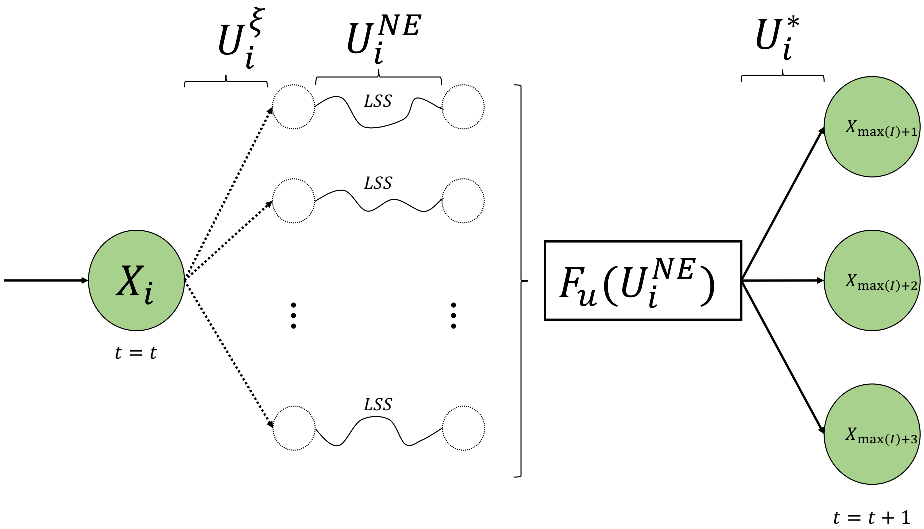


Figure 12: Visualization of the general process of the game and how a state $\boldsymbol{X}_i$ is updated, using the set of random actions $\mathcal{U}_i^\xi$, the set of actions that constitute a Nash equilibrium $\mathcal{U}_i^{NE}$, and the set of actions actually taken $\mathcal{U}_i^*$

4.4.1 Combat Model

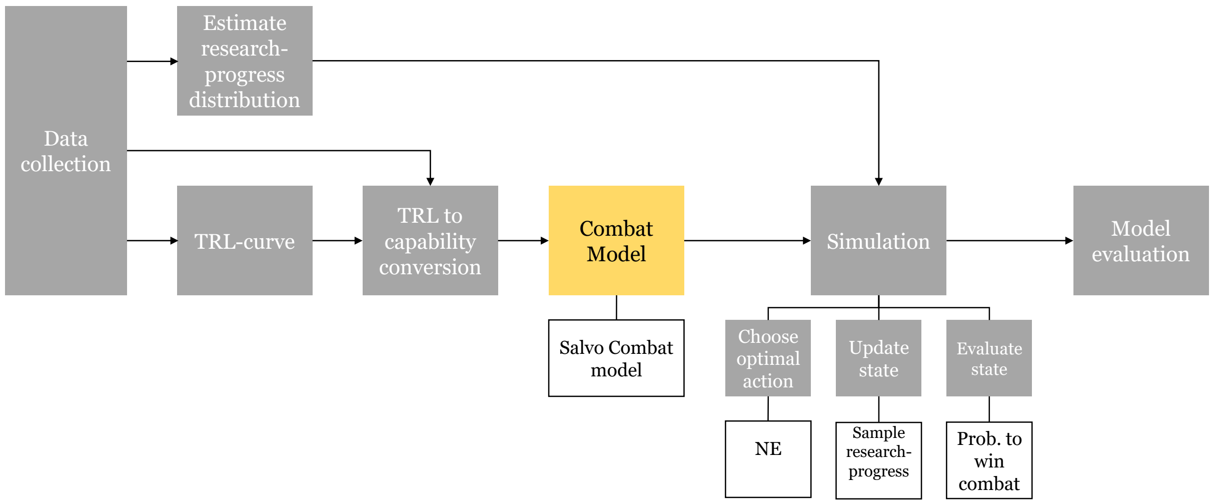


Figure 13: Combat model in method overview

The combat model used in the game is based on a stochastic Salvo combat model, which has been extended to include factors related to intelligence gathering (Armstrong, 2005). The model is used to measure the relative combat capabilities of the two sides and will be used as part of the objective function of the game. The objective function calculates the expected value of the probability that player A has remaining ships or that player B is completely wiped out after firing one Salvo each. Either one of these scenarios are results in a *win* for player A in the simulated battle. The results of the battle depends on the battle parameters, which are expressed in $\boldsymbol{\theta}$. The function arguments $\boldsymbol{\theta}$, referred to as *battle parameters*, can be divided into 8 categories per player:

Table 6: Presentation of $\boldsymbol{\theta}$, referred to as *battle parameters*, used in the extended stochastic Salvo combat model, listing their symbols and description

$\boldsymbol{\theta}$	Description
A, B	Beginning force strength (number of units)
ϕ, ψ	Intelligence gathering
n_α, n_β	Number of offensive actions per unit
p_α, p_β	Probability of offensive actions being accurate
n_y, n_z	Number of defensive actions per unit
p_y, p_z	Probability of defensive actions being successful
u, v	Enemy Losses suffered per successful action
w, x	Staying power per unit

Since this study focuses on modeling the impact of autonomous functions, several parameters in table 6 are set to fixed values and will not be influenced by technological advancements throughout the game. These include the number of units available (A and B), which is assumed to be a function of industrial capacity rather than technology. Additionally, the parameters u, v, w, and x are treated as static, as they reflect the ability of units to withstand damage and continue operating, which is assumed to be determined by factors such as armor, size, and build quality rather than autonomous functions.

The stochastic Salvo combat model accounts for battlefield initiative (Armstrong, 2014), but requires it to be known before evaluating the function. An addition has been made to the model in this study to reflect the stochastic nature of who takes the battlefield initiative. We assume that the efficiency of the players' intelligence gathering in any given state is normally distributed, $\mathcal{N}(\phi, \sigma)$ and $\mathcal{N}(\psi, \sigma)$, respectively, where σ is a known variable. This assumption matches those made regarding other parameters of the model and is intuitive as this performance would be made up of a large number of tasks, each with an unknown distribution of performance on a day-to-day basis. Thus, according to theorem 1, their sum is approximately or precisely normally distributed. Consequently, the distribution of player A's advantage is:

$$\mathcal{N}(\phi - \psi, \sqrt{2}\sigma). \tag{4.5}$$

Whenever two evenly matched sides engage in combat, it is possible that either side will have a significant advantage in terms of the initiative. This possibility is non-negligible, and the probability of either side being favored grows as the difference in ability increases. This can be modelled using the joint probability density function (4.5). Let this probability distribution function be denoted by h and c be a constant scaling factor. The probabilities of player A having a significant advantage, neither player having one, and player B having one can be expressed as:

Neither player favored	$p_0 = \int_{-\sigma \cdot c}^{\sigma \cdot c} h(x)dx$
Player A favored	$p_A = \int_{\sigma \cdot c}^{\infty} h(x)dx$
Player B favored	$p_B = \int_{-\infty}^{-\sigma \cdot c} h(x)dx.$

where p_0 is the probability that neither player has an initial advantage, p_A is the probability that player A has an initial advantage, and p_B is the probability that player B has an initial advantage. If either player has the initiative in a battle the order in which the sides fire their salvos is different and thus the outcome of the battle may change.
When neither player has a initiative they fire their salvos simultaneously and their distributions of surviving units can be calculated separately, as seen in algorithm 3 below.

Algorithm 3: G_0

Input : Parameters: $\boldsymbol{\theta}, R$, Function: Λ
$[\mu_{A_1}, \sigma_{A_1}] \leftarrow \Lambda(R \cdot \boldsymbol{\theta})$
$[\mu_{B_1}, \sigma_{B_1}] \leftarrow \Lambda(\boldsymbol{\theta})$
Output : $\mu_{A_1}, \sigma_{A_1}, \mu_{B_1}, \sigma_{B_1}$

Figure 14: No initiative, both sides fire simultaneously

When player A has the initiative during a battle, player A is assumed to fire a salvo first, likely resulting in some losses for player B before they can fire back. This means a distribution of survivors on the B-side has to be constructed and sampled before the same is done for player A. This is described by algorithm 4 below.

Algorithm 4: G_A

Input : Parameters: $\boldsymbol{\theta}, R$, Function: Λ
$[\mu_{B_1}, \sigma_{B_1}] \leftarrow \Lambda(\boldsymbol{\theta})$
$\mathrm{B}_1 \sim \mathcal{N}(\mu_{B_1}, \sigma_{B_1})$ `//Draw random sample from the distribution`
$\theta_{0,1} \leftarrow B_1$
$[\mu_{A_1}, \sigma_{A_1}] \leftarrow \Lambda(R \cdot \boldsymbol{\theta})$
Output : $\mu_{A_1}, \sigma_{A_1}, \mu_{B_1}, \sigma_{B_1}$

Figure 15: Player A has initiative and fires first

In the final case, when player B holds the initiative they will fire first and player A will likely take some casualties before firing back. This is described in algorithm 5 below.

Algorithm 5: G_B

Input : Parameters: $\boldsymbol{\theta}$, R, Function: Λ
$[\mu_{A_1}, \sigma_{A_1}] \leftarrow \Lambda(R \cdot \boldsymbol{\theta})$
$\mathrm{A}_1 \sim \mathcal{N}(\mu_{A_1}, \sigma_{A_1})$ `//Draw random sample from the distribution`
$\theta_{0,0} \leftarrow A_1$
$[\mu_{B_1}, \sigma_{B_1}] \leftarrow \Lambda(\boldsymbol{\theta})$
Output : $\mu_{A_1}, \sigma_{A_1}, \mu_{B_1}, \sigma_{B_1}$

Figure 16: Player B has initiative and fires first

These algorithms G_0, G_A and G_B output the parameters of a normal distribution of surviving units on either side. They have been written as algorithms rather than functions as they differ mostly in the order of execution and are cumbersome to understand in the form of functions. They are however intended to be implemented as functions and will henceforth be used as such. To calculate the probability of player A winning, the function S is constructed:

$$S(\mu_A, \sigma_A, \mu_b, \sigma_b) = 1 - P(A_1 \leq 0, B_1 > 0), \tag{4.6}$$

where $A_1 \sim \mathcal{N}(\mu_A, \sigma_A$ and $B_1 \sim \mathcal{N}(\mu_B, \sigma_B)$. The expected values of the battle function can then be calculated using a Monte-Carlo simulation, over the weighted average of the three cases, resulting in the objective function:

$$f(\boldsymbol{\theta}) = \mathbb{E}\left[\sum_{j \in Q} p_j \cdot S(G_j(\boldsymbol{\theta}))\right], \tag{4.7}$$

where $Q = \{0, A, B\}$.

4.4.2 Choosing the Optimal Action

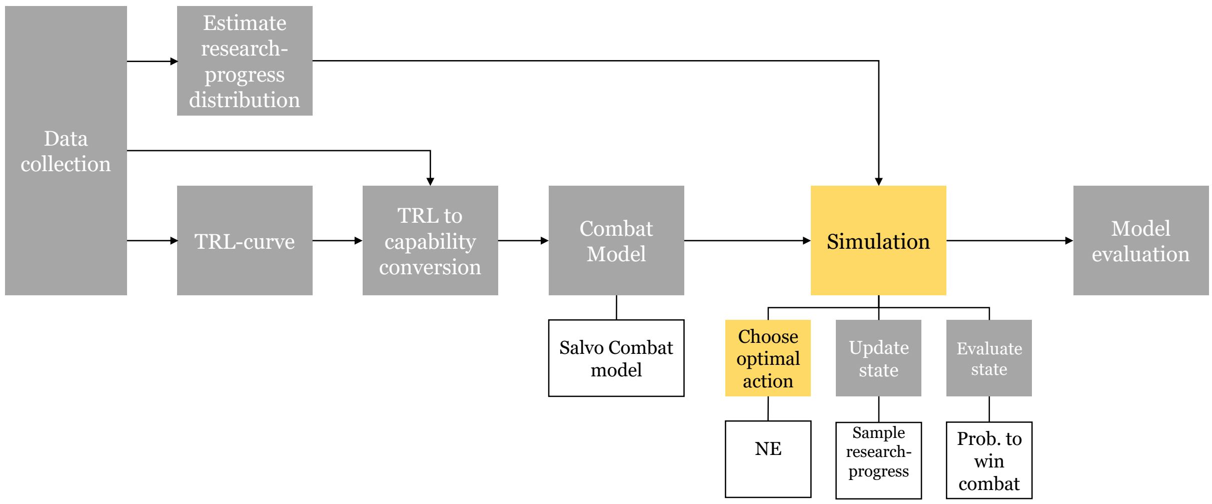


Figure 17: Choosing the optimal action in method overview

In a zero-sum game, such as the proposed game, there is no natural way to characterize a common solution as optimal, according to the definition of a zero-sum game. What is beneficial for one player must necessarily be detrimental for the other. Therefore, the problem of finding an optimal action is transformed into finding the total set of (local) Nash equilibrium (NE) available from the current state that the players are in. In other words, the goal is to identify the set of actions that form a (local) NE, where neither player would benefit from changing their action within the (local) permissible action space.

In their work, Mazumdar et al. (2019) proposed a method for finding locally asymptotic equilibrium (LASE) points in continuous zero-sum games. They showed that all differential NE of such games is LASE points. This method, called LSS, is described in further detail in section 3.8.1. LSS guarantees convergence to a (local) NE, assuming such a point exists. It achieves this by introducing a new process that acts as a limiting ordinary differential equation, steering the gradient trajectory away from stationary points that are not (local) NE.

In every state $\boldsymbol{X}_i \in \mathbb{R}^{N\times 2}$, that the game reaches, each player tries to optimize their technology investments $\boldsymbol{u}_i$ in order to improve their expected outcomes. The agents attempt to maximize the expected value of the next state by varying their actions taken, starting from random points sampled from a sector of a hypercube of dimension N. The samples are bounded so that the initial investment in each individual technology is greater than 0 and their sum is smaller than 1. As the next state is stochastic, it is not possible to exactly evaluate the value of any action. The expected value is computed using Monte-Carlo simulations over both the technology research progress and the outcome of the combat model.

So, from each state $\boldsymbol{X}_i$, a set $\mathcal{U}_i^{\xi}$ of random actions $\boldsymbol{u}_{i,j}^{\xi}$ are initiated from node i with index j, where the LSS algorithm tries to find a (local) NE in the vicinity of each random action. This will give rise to a set $\mathcal{U}_i^{NE}$, containing actions $\boldsymbol{u}_{i,j}^{NE}$ that constitute different NE between the players from node i with index j, and each will serve as a potential action for the players. If the model were to update the state with every $\boldsymbol{u}_{i,j}^{\mathrm{NE}} \in \mathcal{U}_i^{NE}$, the number of states would increase exponentially by a factor equivalent to the cardinality of the number of random initiated actions, $n(\mathcal{U}_i^{\xi})$. This means that the number of states grows to an unmanageable and uncomputable number (with the computational means available in this study) for a relatively low number of random actions $n(\mathcal{U}_i^{\xi})$, meaning that some filtration $F_u(\mathcal{U}_i^{NE})$ is needed in order to limit the growth of the number of states for each time step.

The filtration function $F_u : \mathcal{U}_i^{NE} \rightarrow \mathcal{U}_i^{*}$ performs PCA on the set $\mathcal{U}_i^{NE}$ for each i, and projects each action $\boldsymbol{u}_{i,j}^{NE} \in \mathcal{U}_i^{NE}$ onto the first principal components that capture more, or equal to, 80% of the variance of the actions. This is done in order to reduce the number of dimensions of the matrix $\boldsymbol{u}_{i,j}^{NE}$. After that, the downsized actions in $\mathcal{U}_i^{NE}$ are clustered using the K-means method described in section 3.6.2. The number of clusters is determined to an optimal amount according to the Silhouette method described in appendix A. The resulting centers of the clusters are projected back on the regular action space, and constitute the actions $\boldsymbol{u}_{i,j}^{*} \in \mathcal{U}_i^{*}$ that the player takes from node i with index j.

4.4.3 Updating the State

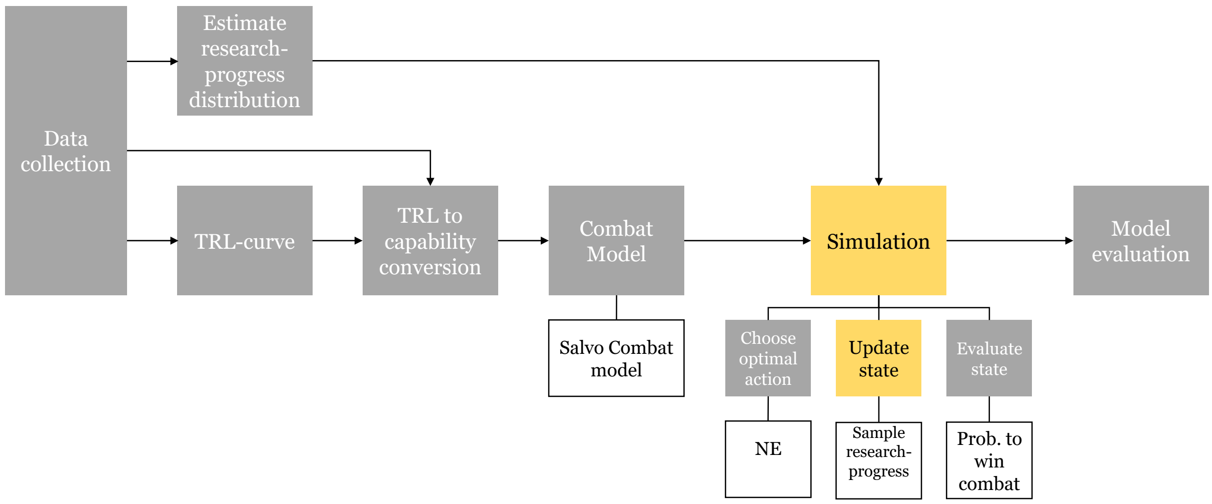


Figure 18: Updating the state in method overview

The knowledge gained from investing in technology could be reasonably assumed to be approximately proportional to the effort and capital expended, as well as some stochastic factor to scale this progress. Radicchi et al. (2008) uses the total number of citations as a metric for scientific impact and argues that it is a good indicator of the importance of results and scientific progress. Given this assumption, it can be shown that the scaled number of citations follows a log-normal distribution such that $\ln(c/c_0) \sim \mathcal{N}(\mu_\xi, \sigma_\xi)$, where c is the number of citations a given article will have, c_0 is the average number of citations for articles in a particular field, and σ_ξ is the standard deviation of articles in the field (Radicchi et al., 2008).

Starting from the point in time when the players have chosen their respective actions – representing technology investments – contained in the set $\mathcal{U}_i^*$, the state $\boldsymbol{X}_{\max(I)+j} \in \mathbb{R}^{N\times 2}$ is updated accordingly:

$$\boldsymbol{X}_{\max(I)+j} = \boldsymbol{X}_i + \boldsymbol{u}_{i,j}^* \cdot \min(\boldsymbol{\xi}_i, 5), \quad \forall j = 1 \ldots n(\mathcal{U}_i^*) \tag{4.8}$$

where $\boldsymbol{\xi}_i \in \mathbb{R}^{N\times 2}$ is a matrix containing uncorrelated, random samples drawn from each technology-specific log-normal distribution f_{ξ_n} describing research progress, $n = 1 \ldots N$, and $i \in I_t$ where $t = 1 \ldots T$. The $\min$ operator is introduced to reduce the frequency of extreme values and achieve more realistic and robust calculations.

4.4.4 Evaluating the State

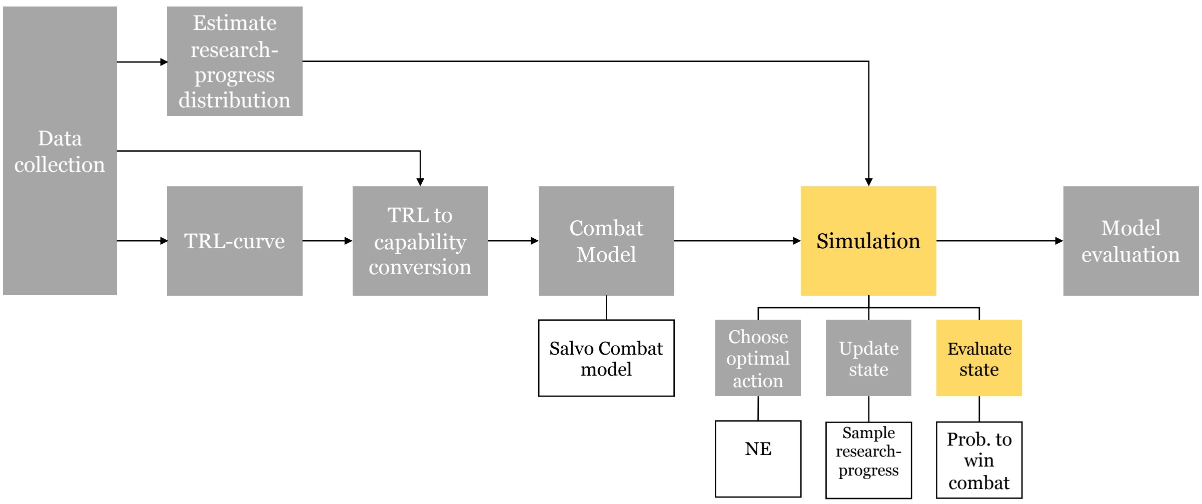


Figure 19: Evaluating the state in method overview

The degree of maturity of each technology available to the players is described by the TRL function in (4.4), and is computed by $g_p(\boldsymbol{X}_i)$. The maturity level of each player's technology portfolio ultimately determines how well each player performs on the battlefield, which will be simulated using the Salvo combat model described in section 4.4.1. To perform this simulation, the technology readiness levels $g_p(\boldsymbol{X}_i)$ for each player must be converted to parameters in the Salvo combat model $\boldsymbol{\theta}_i$, representing relevant military capabilities. $\boldsymbol{\theta}_i$ is defined by:

$$\boldsymbol{\theta}_i = \{\boldsymbol{\theta}_k^A, \boldsymbol{\theta}_k^B\}_{k=1}^{7}, \quad \begin{cases} \boldsymbol{\theta}_k = \boldsymbol{\kappa}_k, & k \in \{1,2,4,6,7\} \\ \boldsymbol{\theta}_k = \frac{1}{1+\exp(-\boldsymbol{\kappa}_k/I_k + D_k)}, & k \in \{3,5\} \end{cases}, \tag{4.9}$$

where $\boldsymbol{\kappa}$ is defined by:

$$\boldsymbol{\kappa}_i = \{\boldsymbol{\kappa}_k^A, \boldsymbol{\kappa}_k^B\}_{k=1}^7 = (1 + \omega \cdot C \cdot g_p(\boldsymbol{X}_{-i} + \boldsymbol{u}^*_{-i,j} \cdot \boldsymbol{\xi}_{-i}) \cdot \boldsymbol{\theta}^0), \tag{4.10}$$

and C is the conversion matrix described in section 4.3.3, ω is a scalar, $\boldsymbol{\theta}^0$ is the baseline battle parameters and $-i$ is the parent node to i. The state is evaluated by inserting $\boldsymbol{\theta}_i$ in the objective function in (4.7), which calculates the expected value of the probability of player A surviving one Salvo in the simulated battle.

4.4.5 Mathematical Formulation

The mathematical formulation of the game is a minimax optimization problem (see section 3.7.1). The objective function describes the expected probability of player A surviving one Salvo in the simulated battle; thus player A wants to maximize with respect to the decision variable $\boldsymbol{u}_A$, and player B wants to minimize with respect to the decision variable $\boldsymbol{u}_B$, where $\boldsymbol{u} = (\boldsymbol{u}_A, \boldsymbol{u}_B)$. The model is designed to optimize actions for a single time step in the future, one at a time. As a result, it produces the following optimization model:

$$\max_{u_A} \min_{u_B} f(\boldsymbol{\theta}) = \max_{u_A} \min_{u_B} \mathbb{E}\left[\sum_{j \in Q} p_j \cdot S(G_j(\boldsymbol{\theta}))\right] \tag{4.11}$$

s.t.

$$\boldsymbol{\theta} = \{\boldsymbol{\theta}_k^A, \boldsymbol{\theta}_k^B\}_{k=1}^8, \begin{cases} \boldsymbol{\theta}_k = \boldsymbol{\kappa}_k, & k \in \{1,2,4,6,7,8\} \\ \boldsymbol{\theta}_k = \frac{1}{1+\exp(-\boldsymbol{\kappa}_k/K_k + D_k)}, & k \in \{3,5\} \end{cases}$$

$$\boldsymbol{\kappa} = \{\boldsymbol{\kappa}_k^A, \boldsymbol{\kappa}_k^B\}_{k=1}^8 = \omega \cdot C \cdot g_p(\boldsymbol{X} + \hat{\boldsymbol{u}} \cdot \boldsymbol{\xi}) + \boldsymbol{\theta}^0$$

$$\hat{\boldsymbol{u}} = d \cdot \frac{\exp(\boldsymbol{u})}{||\exp(\boldsymbol{u})||_1}$$

$$\boldsymbol{\xi} \sim \ln \mathcal{N}(\mu_\xi, \sigma_\xi)$$

$$p_0 = \int_{-\sigma_h \cdot c}^{\sigma_h \cdot c} h(x)dx$$

$$p_A = \int_{\sigma_h \cdot c}^{\infty} h(x)dx$$

$$p_B = \int_{-\infty}^{-\sigma_h \cdot c} h(x)dx$$

$$h(x) = \frac{1}{\sigma_h \cdot \sqrt{2\pi}} \cdot \exp(-\frac{1}{2} \cdot (\frac{\boldsymbol{\theta}_1^A - \boldsymbol{\theta}_1^B}{\sigma_h})^2)$$

$$\boldsymbol{u} = \{\boldsymbol{u}_n^A, \boldsymbol{u}_n^B\}_{n=1}^N, \quad u_n^p \geq 0 \qquad\qquad n = 1 \dots N, p \in \{A, B\}$$

The objective function in the above system of equations (4.11), describes the expected probability that player A *wins* in a simulated battle and is estimated using a Salvo combat model: This is done by running the battle function in its different cases of initiative $G_j(\theta), j \in Q$ where $Q = \{0, A, B\}$ (see section 6.2.3). The results from these simulated battles are evaluated using the function S in (4.6), which yields the probability of player A *winning* in each case of initiative. A weighted average is then computed using the probability of each case of initiative p_j, $j \in Q$, this weighted average is a sample from an unknown distribution of the probability of player A winning a battle given the current state $\boldsymbol{X}$ and the current action $\boldsymbol{u}$. A large number of these samples are drawn and the mean value calculated. This is the value of the objective function which the two player will try to influence in their favor by altering their respective technology investments $\boldsymbol{u}_A$ and $\boldsymbol{u}_B$.

R is a matrix that flips the positions of the first and second column in any two-column matrix and is used to mirror $\boldsymbol{\theta}$ when calculating the distribution of surviving units for player B. $\boldsymbol{\theta} \in \mathbb{R}^{\Theta \times 2}$ is a matrix of the Salvo combat model parameters described in table 6, determining the player's capabilities on the battlefield. $\boldsymbol{\kappa} \in \mathbb{R}^{\Theta \times 2}$ is a matrix that maps the TRL of the players' technology portfolio to battle parameters $\boldsymbol{\theta}$ in the combat model, by multiplying the players' current TRL, described by function (4.4), with the conversion matrix $C \in \mathbb{R}^{\Theta \times N}$ (see section 4.3.3), scalar ω, and the baseline battle parameters $\boldsymbol{\theta}^0 \in \mathbb{R}^{\Theta \times 2}$ used by Armstrong (2005), and adding 1. For the battle parameters $\boldsymbol{\theta}_k$, $k \in \{3, 5\}$, which refers to probabilities, a sigmoid function is used to map the parameters to an interval between 0 and 1. Here, the parameters K_k and D_k are assigned values such that $\boldsymbol{\theta}_k = 0.68$, $k \in \{3, 5\}$ in the initial state (Armstrong, 2005), see table 7. $\hat{\boldsymbol{u}}_{i-1} \in \mathbb{R}^{N \times 2}$ is the action of investments, normalized and scaled by the budget parameter d. The expression for $\hat{\boldsymbol{u}}_{i-1}$ ensures that $0 \leq \sum_{n=1}^N \hat{u}_{n,p} \leq d$, $p \in \{A, B\}$ where $\hat{u}_{n,p}$ is an element in the the action matrix $\hat{\boldsymbol{u}}_{i-1}$, $n = 1 \dots N$, $p \in \{A, B\}$. $\boldsymbol{\xi} \in \mathbb{R}^{N \times 2}$ is a matrix of random samples drawn from the technology-specific log-normal distribution described in section 4.3.1. p_j, $j \in Q$ describes the probability that a player has an initial advantage over the other, where $j = 0$ denotes that neither player has the initial advantage, where $h(x)$ is the joint PDF of player A being advantageous.

Table 7: Presentation of discretionary parameters in the game, listing their symbol, assigned value, and description

Symbol	Assigned value	Description
K	0.5	Parameter determining the slope of the TRL curve
D	5	Parameter determining the shift of the TRL curve
d	$[1, 1]$	Parameter determining the players' budget
ω	0.4	Scaling the importance of research versus base values
K_3	0.5	Slope of the sigmoid-curve that maps $\boldsymbol{\theta_3}$ to the interval $[0, 1]$
D_3	2	Shift of the sigmoid-curve that maps $\boldsymbol{\theta_3}$ to the interval $[0, 1]$
K_5	0.5	Slope of the sigmoid-curve that maps $\boldsymbol{\theta_5}$ to the interval $[0, 1]$
D_5	1.5	Shift of the sigmoid-curve that maps $\boldsymbol{\theta_5}$ to the interval $[0, 1]$
σ_h	$\sqrt{2}$	Standard deviation of the joint PDF $h(x)$
c	1.6	Constant that shifts of the probability of initial advantage
$\boldsymbol{\theta}^0$	$\begin{bmatrix} 6 & 6 \\ -2 & -2 \\ -6.87 & -6.87 \\ -1 & -1 \\ -2.47 & -2.47 \\ -0.42 & -0.42 \\ 0.5 & 0.5 \\ 1 & 1 \end{bmatrix}$	Baseline battle parameters scaled in order for $\boldsymbol{\theta}$ to match the battle parameters used by Armstrong (2005), in the initial state

4.5 Running the Game

To run the constructed game described in section 4.4, parameters that determine the dynamics of the run are to be assigned. These parameters include the time horizon T (determining the depth of the simulation), the number of random actions I_ξ that are to be initiated from each node i, the action step-length (representing the players' investment budget in each time step), the maximum number of optimization iterations M, and the maximum number of actions I^*_{max} taken from each node i. Values assigned to these parameters are presented in table 8 below:

Table 8: Presentation of hyperparameters that determines the dynamics of a run of the game, listing their assigned value and description

Symbol	Assigned value	Description
T	5	time horizon, where each time step is equivalent to 5 actual years
I_ξ	75	Number of random actions initiated from each node i
d	$[1, 1]$	Total step-length of action
M	200	Maximum number of iterations in the optimization
I^*_{max}	6	Maximum number of actions allowed to take from each node i
a_i	$500 \cdot e^{\frac{-i}{50}} + 100$	Sequence of exponentially decaying values controlling step size of action
b_i	100	Step size of v
v_0	100	Initial value of v
ϵ_1	10^{-3}	convergence criterion for maximum absolute value in gradient vector ω
ϵ_2	0.5	convergence criterion of maximum absolute value of step $[a_n g_x, a_n g_y]$
k	12	Required number of consecutive converged iterations

T is assigned such that the run is to simulate 25 years ahead. By increasing T, the depth of the spanning tree (see figure 5) increases, and consequently, the number of nodes that is to be examined during the entire run, which ultimately increases the duration of the run. The number of random actions $n(\mathcal{U}_i^\xi)$ determines the rate of exploration of the solution space in the game but increases the amount of optimization that needs to be performed, and a reasonable balance between the two is expected to be achieved with a value of $I_\xi = 75$. The step-length d determines the amount of capital each player has to invest in each action and is only used to scale the actions to match the parameterization of the TRL curve, described in section 4.3.2. The number of optimization iterations determines the number of gradient steps performed in each optimization round, where an increased value of M results in longer run-time. The maximum number of actions allowed to take from each node is used to limit the growth of nodes during the game.

The parameters a_i, b_i, ϵ_1, ϵ_2 and k are used when finding (local) Nash equilibrium (NE) through LSS (see section 3.8.1). a_i is used to scale the step taken in the action and b_i is used with the same purpose for v. ϵ_1, ϵ_2 and k are used to check for convergence. LSS promises convergence to a (local) NE as shown by Mazumdar et al. (2019), but does not give a criterion for when to stop iterating. The ones outlined above have thus been developed as part of this study. To account for this uncertainty and the fact that LSS by design is expected to first reach and converge to undesirable solutions before finding an actual LASE-point multiple iterations are required to fulfill both criteria before a solution is considered satisfactory.

The game is run on a high-end PC designed for machine learning and rendering workloads with the following specifications listed in table 9:

Table 9: Specifications of PC used for simulation workload

Component	**make & model**	**Specifications**
CPU	AMD 5800x	8 cores @ 3.8 GHz (max boost 4.7 GHz)
GPU	Nvidia 3080Ti	10240 cuda-cores @ 1.67 GHz
Memory	*unknown*	64Gb DDR4 @ 1.79 GHz

The speed of execution has been measured across CPU and GPU with multi-process and single-process execution. CPU processing with the Python multiprocessing library was found to be the most efficient. This may be due to inefficient implementation and inexperience with the PyTorch library and coding for GPU execution.

5 Results and Analysis

This chapter presents results that have been achieved during various parts of the study, starting with the calibration of parameters for the research progress distributions, and a goodness-of-fit analysis. The results of running the game are analyzed in several different ways and measurements taken during the optimization process are inspected and interpreted.

5.1 Research Progress Distribution Parameters

The results from the parameterization of the research progress distribution seem to confirm the findings of Radicchi et al. (2008), that citation data follow, and can be well-fitted to, a log-normal probability distribution, and thus be successfully describe the probability of receiving a particular scientific impact. The resulting log-normal curves are plotted in figure 20, where an interesting observation is that the fitted curve for every research area is quite similar, and seem to follow one universal line. This result coincides with the findings of Radicchi et al. (2008), that widely scattered distributions of citations for publications in different scientific disciplines are rescaled on the same universal curve when the relative indicator c_f is used.

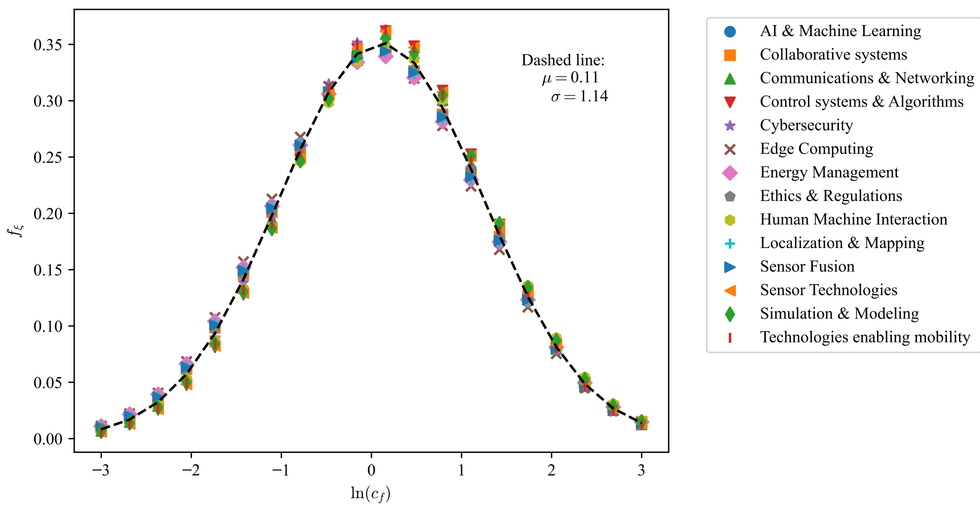


Figure 20: Rescaled probability distribution f_ξ of the relative indicator $c_f = c/c_0$, for research-areas $n = 1 \dots N$. The dashed line is a log-normal fit with $\mu = 0.11, \sigma = 1.14$.

In table 10, the estimated parameters μ_ξ and σ_ξ, and results from the goodness of fit-tests are presented. To evaluate the goodness of the fit of each estimated log-normal curve for research-area $n = 1 \dots N$, a KS-test (see section 3.2.4) and a χ^2-test (see section 3.2.5) were both performed. The null hypothesis for both tests was that the observed citation data follows a log-normal distribution. If we study table 10, we see that the p-value for both the KS-test and the χ^2-test was greater than the significance level $\alpha = 0.05$ for all research-areas $n = 1 \dots N$. This implies failure to reject the null hypothesis, and consequently, that there is not enough evidence to conclude that the observed data significantly deviates from the log-normal distribution at the 5% significance level. Visually, the goodness of the fit can be seen in appendix D.1, which depicts the histogram of the observed citation data and the corresponding fitted log-normal curve, for research-area $n = 1 \dots N$.

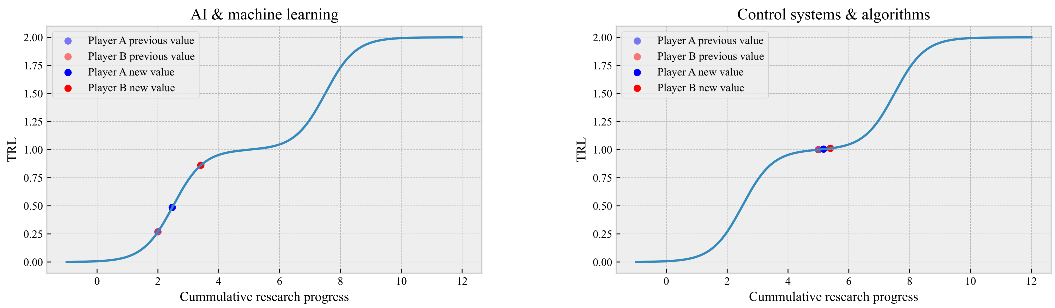
Table 10: Presentation of the results for the parameterization of the research progress distribution f_ξ, listing the mean value μ_ξ, standard deviation σ_ξ, the average number of citations c_0, the maximum number of citations, p-value for a KS- and χ^2-test with significance level $\alpha = 0.05$, for research-area $n = 1 \dots N$

Research area, n	μ_ξ	σ_ξ	c_0	c_{max}	p (KS)	p (χ^2)
AI & Machine Learning	0.07	1.16	12.65	107	0.70	0.33
Collaborative systems	0.10	1.16	13.16	70	0.24	0.27
Communications & Networking	0.13	1.11	10.94	146	0.43	0.46
Control systems & Algorithms	0.17	1.10	9.10	60	0.43	0.48
Cyber security	0.11	1.10	10.7	60	0.24	0.39
Edge Computing	0.03	1.16	19.59	207	0.70	0.41
Energy Management	0.07	1.17	14.46	99	0.43	0.35
Ethics & Regulations	0.09	1.15	17.38	118	0.70	0.50
Human Machine Interaction	0.17	1.14	11.37	60	0.43	0.42
Localization & Mapping	0.06	1.16	13.09	99	0.70	0.34
Sensor Fusion	0.08	1.16	13.12	70	0.24	0.27
Sensor Technologies	0.16	1.10	10.13	60	0.43	0.42
Simulation & Modeling	0.17	1.12	11.09	60	0.43	0.43
Technologies enabling mobility	0.14	1.10	10.24	58	0.43	0.47

5.2 Technology Road-map

With the proposed method, a full run of a game results in a graph of states, connected by the actions taken from each state. The set of nodes $i \in I_t$ where $t = T$ represents the total set of states at the time horizon, containing information on the TRL for each technology at the end of the game. Each of these states serves as a possible scenario for future technological development, and by studying this set, conclusions about technological development for each of the identified technologies can be made. Additionally, by studying clusters of similar end-states, different trajectories of investment more common than others can be identified. These trajectories illustrate strategies that the players are more likely to employ, and thereby closer to optimal to pursue during the game.

In appendix D.2, the resulting technological development – starting from the current real-world state of technology readiness, and simulating 5 time steps ahead (representing 25 years) – is presented for every technology in the examined technology portfolio. Two examples of this are also shown in figure 21. By studying these plots, we can get a clear overview of the technological evolution of each technology, in relation to their respective starting state in the TRL curve. This representation also gives an indication of which technologies are pushed toward a new technological paradigm. For example, if we study the TRL curve for *Control Systems & Algorithms* in figure 21b, we see that TRL-value is greater than 1 and located in the second period of the TRL curve – indicating the start of a new paradigm in that technology area.



(a) AI & machine learning

(b) Control systems & algorithms

Figure 21: Two examples of technological development obtained during the game, for technologies *AI & Machine Learning* and *Control Systems & Algorithms*

While the TRL-curves in appendix D.2, and figure 21 give an intuitive overview of the technological evolution during the game, it is more interesting to investigate technological development in more quantitative terms, and compare the technologies and players relative to each other. If we study table 11 below, we can, for example, see how the players have placed a percentage of their total budget in each technology, and the resulting change in TRL. From table 11 and figure 21a, we can see that player B has focused on *AI & Machine Learning* while player A has found an alternative strategy, placing the majority of investments in *Collaborative systems*.

Table 11: The table presents the results encompassing the technological development for technology $n = 1 \dots N$. It includes the starting state $\boldsymbol{X}_0$, which is the same for both players, as well as the end state for both players. Additionally, it provides information on the percentage increase in technology readiness level (TRL) for all technologies for both players. Lastly, the table presents the average investment of the total budget for both players

Research area, n	$\boldsymbol{X}_0$	$\boldsymbol{X}_T, A$	$\boldsymbol{X}_T, B$	$\Delta TRL, A(\%)$	$\Delta TRL, B(\%)$	(%) of budget, A	(%) of budget, B
AI & Machine Learning	2.00	2.88	3.54	43.76	76.90	16.37	32.59
Collaborative systems	2.00	3.06	2.91	53.21	45.67	36.07	16.24
Communications & Networking	4.00	4.55	5.00	13.87	25.05	0.68	1.21
Control systems & Algorithms	5.00	5.81	5.53	16.30	10.64	0.69	0.92
Cyber security	3.00	3.44	3.54	14.68	17.97	0.82	2.02
Edge Computing	2.00	2.67	2.23	33.38	11.35	0.89	1.37
Energy Management	4.00	4.50	4.36	12.52	9.11	0.65	0.84
Ethics & Regulations	2.00	2.51	2.88	25.56	44.20	0.64	11.85
Human Machine Interaction	3.00	3.41	4.00	13.78	33.26	0.95	1.03
Localization & Mapping	3.00	3.71	3.50	23.71	16.64	1.11	1.81
Sensor Fusion	3.00	3.72	3.20	24.17	6.52	4.84	1.94
Sensor Technologies	4.00	4.48	4.63	11.92	15.81	0.87	0.94
Simulation & Modeling	2.00	3.62	2.79	81.05	39.48	25.78	26.30
Technologies enabling mobility	3.00	3.39	3.54	12.97	17.96	9.63	0.93

The player's respective investment strategy is clearly visualized in figure 22, showing the average technology investments by each player during the entire game, which is also shown in table 11 displaying the share of total budget invested in the respective technology. There are several factors that influence the players' choice of investment; firstly, where in the TRL curve the technology is located plays a crucial role in the value of the gradient of the objective function in (4.7). If a technology is located in a steep section of the TRL curve, there will be greater progress per invested unit than a technology located in a stagnated section. This distinction is clearly visualized between figure 21a and 21b, where the starting state of *AI & Machine Learning* is located at a more steep part of the curve than the starting state of *Control systems & Algorithms* – resulting in more investments being placed there.

Secondly, the mapping of technologies to battle parameters done by the conversion matrix C also plays a significant role in the evaluation of the gradient. Conceptually, the value of each element in C reveals the impact that each technology will have on the different battle parameters, and a high value in C will contribute to a larger gradient value for that technology, and subsequently influence how the players invest. Lastly, the objective function has inherent stochasticity which will affect the gradient of the objective function as well. Weighing these factors together ultimately creates the complex behavior we see in the investments of the players. An interesting observation (demonstrating this complex behavior) can be made in figure 22 when studying the relationship between the player's average investments with the share of weights in C; the technology area that was rated as the most impact-full over the collection of battle parameters was *Sensor Technologies*, and as shown in figure 22 and table 11, neither player has invested a significant amount in that technology area. The reason for this can be found in the plot of the TRL curve for *Sensor Technologies* (see appendix D.2, or $\boldsymbol{X}_0$ in table 11) where we can observe that the technology already is at a mature state of the TRL curve in its initial state. So, even though *Sensor Technologies* have a great impact on the battle parameters – which affects the gradient – the players do not see a benefit great enough to place their investments there and instead find other investments that are more favorable.

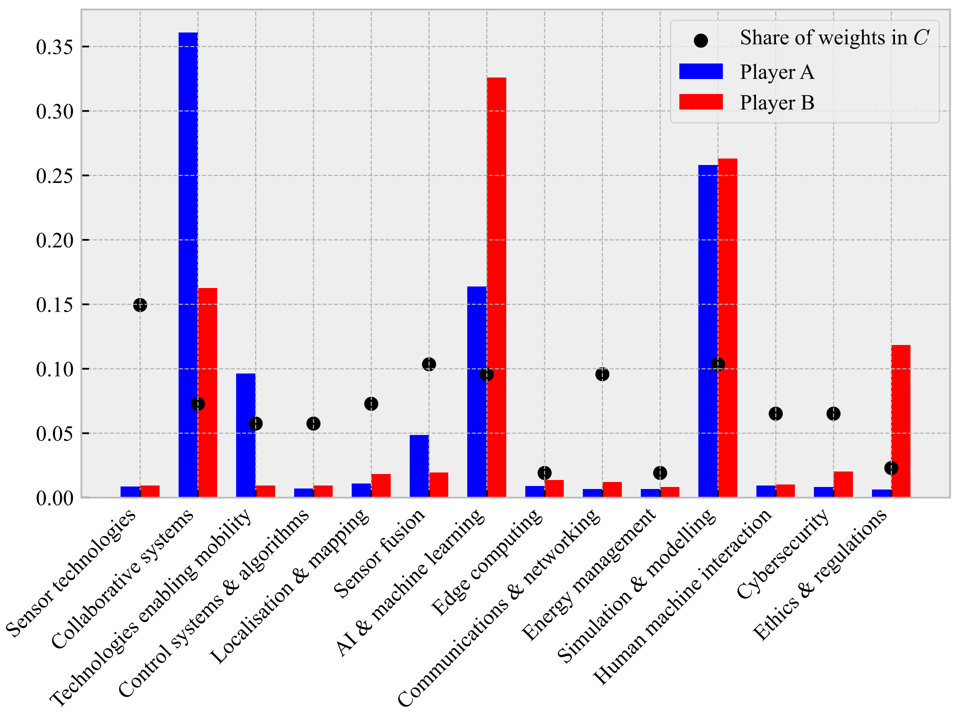


Figure 22: Average technology investments compared with weights in C, the matrix specifying mapping between technologies and battle parameters. A larger share of weights in C would with a naive strategy lead to larger investments.

Another interesting question that is raised when analyzing the set of end-states $i \in I_T$, is if there is some subset of the solution space that is more common as a solution than others. If that would be the case, we would see that the players invest in such a way that they end up in the same area of the solution place more frequently. To evaluate such a behavior, trajectory plots that track how the state evolves during the game for both players have been made, projected on the first 3 principal components. This is presented in figure 23, showing two randomly sampled end-states from the set I_T. In appendix D.3 a larger sample-set of trajectories is presented. What we can see from figure

23 is that the players seem to be drawn towards a certain subset of the solutions space. They seem to follow some universal trajectory with a degree of noise, and *not* travel completely randomly between different parts of the solution space. Instead, the players seem to choose a certain direction that they consider to be the best. We can also see this behavior in figure 25b, which depicts a scatter plot over the total set of states (nodes $i \in I_t$ where $t = 1 \ldots T$) accumulated during the game. Here, we see a clear trend in the evolution of the states moving toward one corner of the 3-dimensional principal component space.

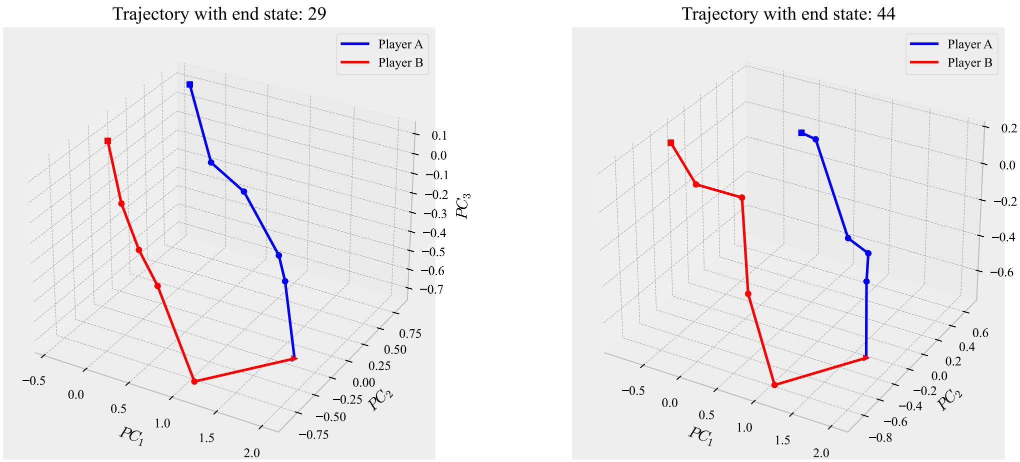


Figure 23: Trajectories for player A and player B in 4 random sampled end-states, projected on the 3 first principal components of states reached by either player. The trajectories are *joined* at the starting state and diverge as the game moves forward

These observations suggests that the model successfully captures a behavior of active, non-random decision-making for the players – despite acting under uncertainty and stochasticity. In fact, the decisions made by the players indicate an underlying attractiveness, pushing them towards a reward that is more beneficial than others.

5.3 Objective Function Values Throughout the Game

Figure 24 shows how objective function values change during the course of the game. The scores all lie in the range $[54\%, 76\%]$ and the simulated battles are therefore highly contested and uncertain throughout the whole game. Noticeably the average across states in the same time step moves very little, indicating the game is fairly constructed with no clear advantage given to either side. If this were not the case we would expect to see the average steadily move in favor of the advantaged player.

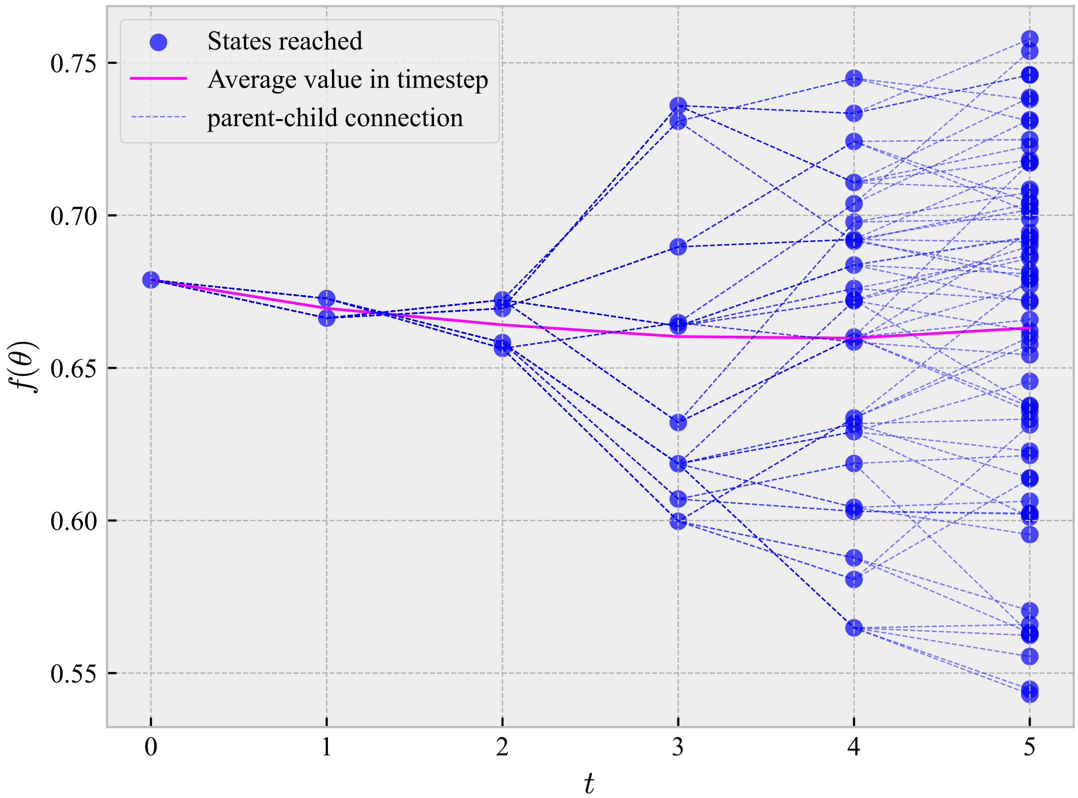
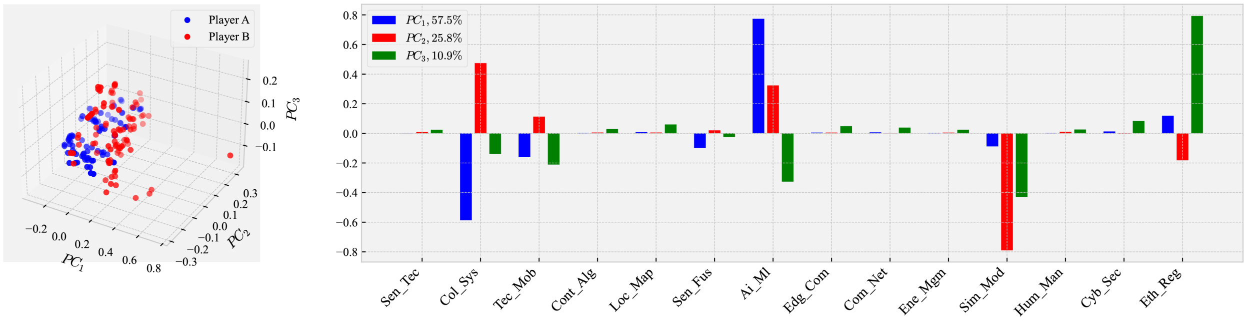


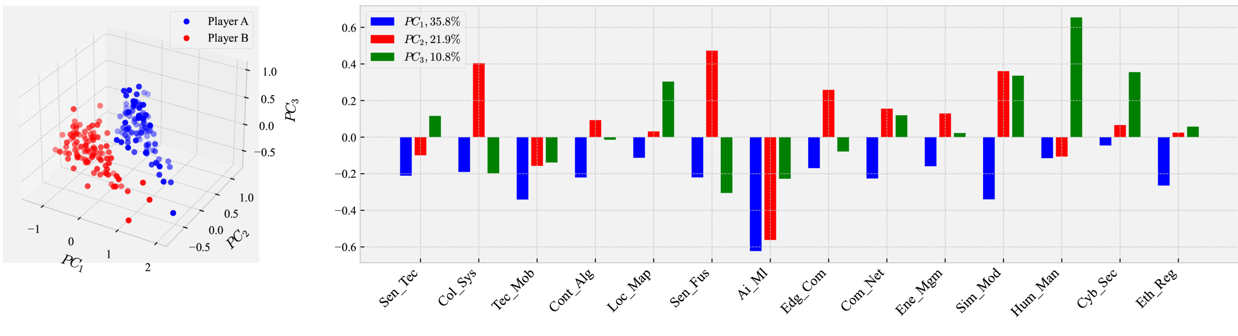
Figure 24: Objective function $f(\theta)$ evaluated at every state reached. Child and parent states are connected to show trajectories throughout the game

5.4 Principal Component and Clustering Analysis

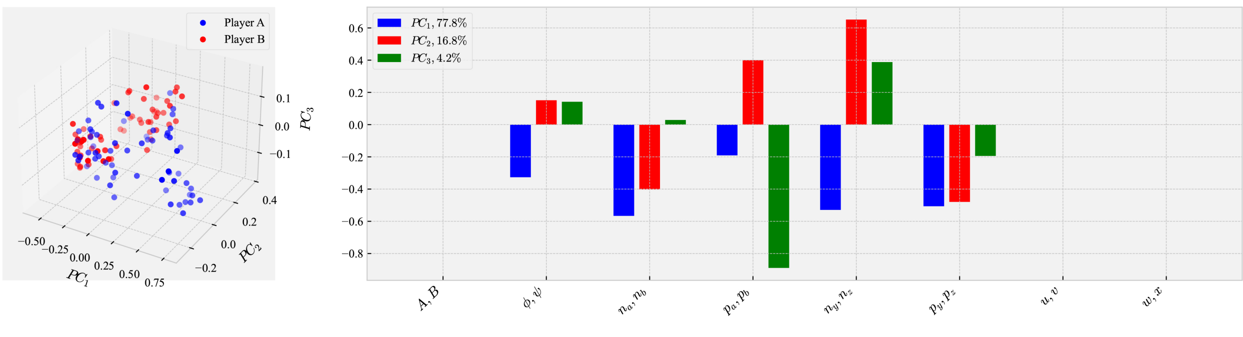
In a high-dimensional space such as the one in the proposed model, visually inspecting and analyzing data is very hard. PCA has therefore been applied to the actions taken by the players, the states they have reached thereby, and the parameters passed to Salvo combat model. By projecting the original data onto the first 3 components the data can be visualized in this new 3-dimensional space as in figure 25 with only minor loss of information, presented in figure 26.



(a) Actions



(b) States



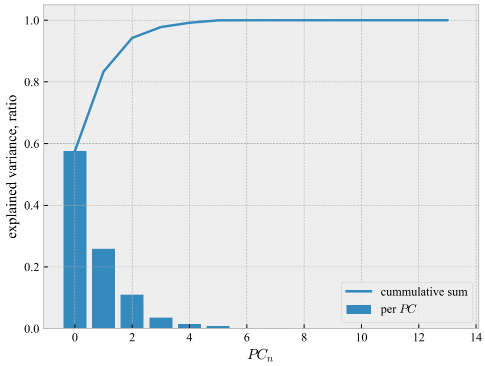
(c) Battle parameters

Figure 25: Actions, states, and battle parameters projected onto their first 3 principal components respectively, as well as the values of these principal components. The 3d plots show degrees of clustering in the different measurements and the principal component values are included for further analysis and discussion.

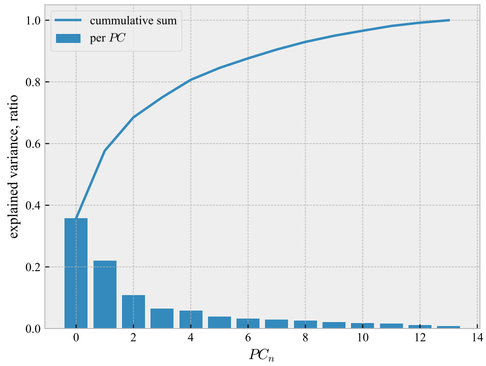
The figure above describes, in order, the actions taken by each player, the resulting states the player enters, and finally the resulting battle parameter values all projected onto their first 3 principal components respectively. A detailed description of what Principal component analysis is and how it works can be found in section 3.6.1. We see various levels of clustering among the different plots where the states in figure 25b very clearly shows one cluster for player A and one for player B. This is expected as they optimize for different outcomes and serves as verification of the combat model. The main difference between the states reached by the two players seems to be along the second principal component PC_2 which contributes about 22% of the total variance. These components indicate there is a negative relationship between research progress in *AI & Machine Learning* versus research progress into *Collaborative systems* and *Sensor fusion*, among others. This relationship can also be seen in the plot of average technology investments (see figure 22), where player A invests much more in *Collaborative systems* and *Sensor fusion*, and player B favors *AI & Machine Learning*.

The plots show clustering among actions and battle parameters as well, although not as clearly. The battle parameters appear to be comprised of three significant clusters, each exhibiting a considerable amount of variance within themselves. Interestingly, only player A is present within the cluster located in the bottom right, centered around the coordinates $(0.5, 0.1, -0.15)$. The presence of these three clusters suggests the existence of multiple distinct strategies, all of which are locally optimal. However, the varying concentrations of each player within these clusters indicate that these locally optimal strategies may not necessarily be optimal for both players.

Figure 26 displays the explained variance of actions and states by the principal components. The first 3 components utilized in constructing the plots shown in Figure 25 account for approximately 94% and 69% of the total variance in actions and states, respectively. The lower explained variance observed for states can be attributed to the stochasticity introduced during state updates, as mentioned in section 6.2.1. This stochasticity introduces distortions in the actions taken by the players and is independent across different technology areas. Consequently, to achieve a perfect explanation, precisely N (in this case, 14) components would be required.



(a) Actions



(b) States

Figure 26: Explained variance by each principal component, including cumulative sum. The PCA is conducted on all actions taken and states reached, by both players

5.5 Action Convergence

LSS, described in section 3.8.1, has been used to find the (local) Nash equilibrium (NE) of the game, that forms the basis of this analysis. This subsection will investigate the results of the optimization itself and interpret various measurements taken during the process.

5.5.1 Inspection of Hessian Matrices

LSS uses Jacobian and Hessian matrices to update the steps taken during the optimization. By studying the dynamics of these matrices, conclusions can be drawn about the dependencies between investments in different technologies for the players. Figure 27 below shows an example of a Hessian matrix during a random sampled iteration of the LSS algorithm:

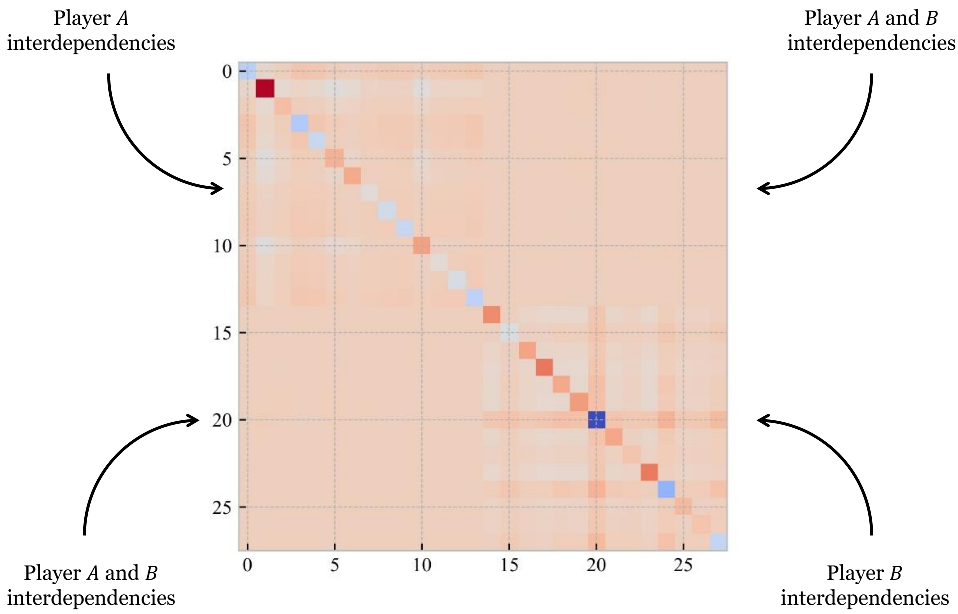


Figure 27: Hessian matrix normalized and plotted as a heat map in randomly sampled iteration of the LSS algorithm.

Looking at the example above (and several others during the development process), it is apparent that the Hessian matrix often takes the form of a block matrix. The two blocks along the diagonal provide information on the interdependencies between technology investments in the players' own portfolios, and the two blocks along the anti-diagonal provide information on the interdependencies between investments in player A:s portfolio and investments in player B:s portfolio. The blocks on the anti-diagonal are smaller than the values on the diagonal by one or more orders of magnitude. This structure suggests that technology investment for one player is highly interdependent among the technologies belonging to the same portfolio, but is less dependent on the investments placed by the other player. Additionally, it appears that the second derivative with respect to a single technology is generally much larger than the second-order partial derivatives with respect to two different technologies. This can be interpreted by studying the diagonal elements in the Hessian matrix.

Furthermore, the observation that the second derivative with respect to a single technology is much larger than the second-order partial derivatives with respect to two different technologies implies that changes in one technology have a more substantial impact on the objective function than changes in two different technologies simultaneously. This observation could have significant implications for decision-making processes related to technology investment or allocation of resources. Incentivizing the players to invest their resources in a more narrow set of technologies. The discussed structure of the observed Hessian matrices can more formally be described by:

$$\begin{aligned} D^2_{x_i,x_i} f(x,y) \gg D^2_{x_i,x_j} f(x,y) \gg D^2_{x_i,y_j} f(x,y) \neq 0, \quad \text{where } i,j \in 1 \dots N, i \neq j \\ D^2_{y_i,y_i} f(x,y) \gg D^2_{y_i,y_j} f(x,y) \gg D^2_{x_i,y_j} f(x,y) \neq 0, \quad \text{where } i,j \in 1 \dots N, i \neq j \end{aligned} \tag{5.1}$$

Note that this behavior is not omnipresent or law but rather an observation and that while the Hessian matrix's anti-diagonal blocks are generally smaller than the diagonal blocks, it is essential to note that these values are still non-zero and have a significant impact on the decision-making process, not least as it is used in LSS to differentiate between extreme values and saddle points. The fact that the anti-diagonal blocks of the Hessian matrix show non-zero values, proves that the model captures the game-theoretic behavior that it is trying to model and that the players are basing their investments on information about the other player's investments. However, the fact that these values are so much smaller than the values along the diagonal, indicates that the method used in the model is not complex enough to capture the actual real-world dependencies we would expect to see.

5.5.2 Inspecting Convergence and Step Sizes

By inspecting the step sizes taken and relating this to the gradient of the objective function, we get a better understanding of the difference between LSS and regular GDA as described in section 3.8. LSS is intended to recognize convergence to a local extreme point meaning a current set of actions is locally optimal for player A or B and in that case take steps away from this point until a (local) Nash equilibrium (NE) is found, these are characterized by being locally optimal for both players. Crucially, GDA does not make this distinction. By inspecting figure 28 below, one can more intuitively understand how this process works. Figure 28 shows the size of the gradient, the step length, and the objective function of the current solution versus the iteration number of LSS.

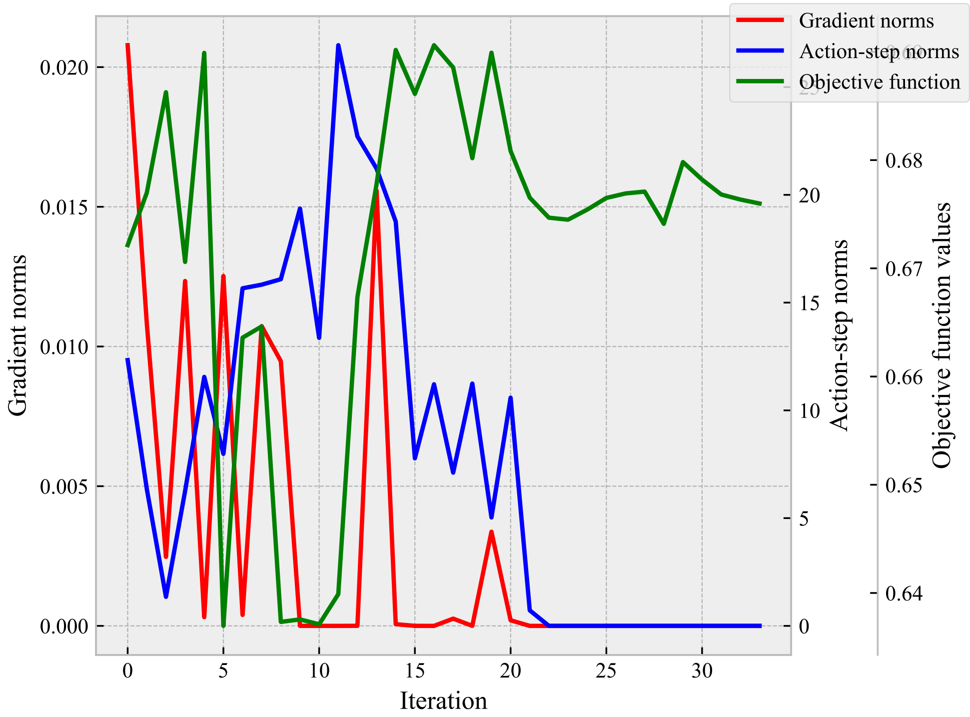


Figure 28: Norm of gradient, norm of action-step and objective function value for every iteration of LSS algorithm until convergence, in a random sampled iteration of the game.

The plot above is by no means smooth and have no natural way of measuring the rate of convergence. This is, however, by design. The spikes in step size indicate the algorithm has identified the current solution as being an extreme point rather than a saddle point and thus works to push the solution away from that particular stationary point. The following spike in the gradient norm indicates we are moving toward a new stationary point which could be either a saddle point (local NE) or another extreme point. The convergence criteria and hyperparameters have been set with the goal of only exiting optimization at true (local) NE but no reliable method has been found to verify this. An attempt has been made by overlaying the objective function value to these plots. As expected the objective function changes more after iterations with a large action step and is more stable when these are smaller. Note that there the objective contains stochastic elements and the number of simulations made during optimization is not large enough to remove this factor completely. This explains the changes in the objective function value even after iterations where the actions taken by the players change very little.

6 Discussion

This chapter presents a comprehensive discussion of the obtained results and the methodology employed to generate them. It explores the impact of various methodological choices and limitations while offering suggestions for future research to advance the development of this research topic. Furthermore, ethical considerations pertaining to the thesis are addressed. Finally, the chapter concludes by presenting the final conclusions.

6.1 Evaluation of Results

The model and its associated parameters as presented in this thesis produce results that seem neither random nor perfectly aligned to the prior knowledge and biases gathered from the queried subject matter experts during the data collection phase. The technology investments made by both players are reasonable and show clear preferences for some technologies over others while still not appearing binary, but rather more granular in nature. These observations indicate that the mathematical modeling built into the game is to some extent successful in capturing the decision-making of two rational entities serving military purposes. Furthermore, the technology investments of the two players show both similarities and clear differences, this is especially clear when looking at the trajectories in figure 23 and the clusters in figure 25. As an asymmetric objective function where the players optimize for different outcomes is used, it is expected they would act slightly differently. This result further serves to verify the construction of the game by confirming expected behavior. Additionally, the objective function values do not drift throughout the game but the game instead seems rather balanced. Suggesting the game is fair and the chosen actions are in fact (local) Nash equilibrium (NE).

The results in section 5.5.1 indicate that the primary driver of technology investments is the benefits a single technology could have on future outcomes of battle, and that complex combinations of investments between many different technologies in the portfolio in a single investment round to be of lesser effect. Additionally, it is evident that the investments of the opponent affect the player's choice of investment even less. Critically however, we see that these more complex relationships are captured by the model such that the principles of portfolio construction and countering one's opponent's action is relevant and impact decision-making. These insights suggest the game-theoretic aspects of the constructed model are captured and affect the decision-making of the two agents.

The results further indicate that each sub-step of the method; including the estimation of the research progress distributions; the modeling of the TRL curve; the conversion of technologies to military capabilities; the simulation of a battle using the Salvo combat model; the optimization to find equilibria points using LSS; and the filtration function that uses PCA and K-means, seem to work as anticipated – producing a rather well-behaved game. Moreover, the results show promising potential to serve as a quantitative aid to those qualitative, more broadly used road-mapping techniques that are present in the field today. The results that are obtained in this study should *not* be considered as neither optimal nor recommendations from an actual decision stand-point, but with further work on mainly how combat is simulated and technologies are evaluated, the model could in the future be used to weigh results against more qualitative work such as in the work by FOI (2022).

To summarize, these insights suggest that the methodology adopted in this study is promising, and that it may be fruitful to approach technology development as a series of investment decisions as one would when constructing a portfolio of financial assets. Further developing of such a framework, until the point of real-world usability would serve to introduce a quantitative perspective to the decision-making progress and thus contribute to better quality decisions.

6.2 Evaluation of Methodology Choices

In this section, we evaluate the different methodology choices and delimitations that have been made throughout the course of this study. Decisions have been made continuously in order to answer the purpose of the thesis, ultimately impacting the outcome of the results.

6.2.1 Modeling Research Progress

When modeling technological development, the correlation and interdependence in technological development between different technology areas have not been considered. This means that for two deeply interconnected technology areas, $T1$, and $T2$, that have a lot of dependencies between each other, a significant technological breakthrough in $T1$ will have no impact on $T2$, whereas in reality a significant development $T1$ would result in some indiscriminate development in $T2$. For example, if we study the case when $T1$ is assigned with the technological research area of microprocessors, and $T2$ with the technology area of AI & Machine Learning. A significant development in $T1$, leading to more improved microprocessors, would result in an enhanced capacity for computational power – enabling new prerequisites within $T2$ and for new projects to be initiated. In more mathematical terms, for each technology $n = 1 \ldots N$ in the technology portfolio (see table 3) there is some correlation coefficient $\rho_{ij} \neq 0, \quad i, j = 1 \ldots N$, indicating dependencies between the technologies. By not considering this correlation, the proposed model gains in simplicity, interpretability, and a reduction of the source of errors, but forfeit in modeling reality.

To mathematically model technological development over time, the concept of TRL and sigmoid functions have been used to construct the TRL function in (4.4). Firstly, the delimitation that all technologies operate under the same TRL curve was made. This is equivalent to the assumption that all technologies require the same amount of resources to induce advancements and the same rate of development. In reality, every technology is likely to operate under its own TRL curve, being different investment-heavy, in need of different amounts of research, and inherent to different levels of risk. If the proposed model were to take these factors into account, it would have a significant impact on the decision-making of the players; an investment in a more investment-heavy technology with a lower rate of change would produce a smaller gradient value than a technology operating under a TRL curve with a steeper slope. By not considering these differences in the evolution of technological development between the technologies, and assuming one universal TRL function, the gradient is not affected by technology-specific factors. Instead, the gradient is affected by the mapping of technologies to battle parameters in the combat model, described in section 4.3.3, and the expected value of the objective function, obtained by Monte-Carlo simulations.

By assuming one universal TRL curve, the amount of parameterization needed is significantly decreased, which in turn reduces the source of errors in the model. The choice of parameters K and D – determining the properties of the universal TRL curve – were made in order for the TRL curve to harmonize with the budget parameter d, and a sample ξ from the research progress distribution, such that one technological paradigm (one period) in the TRL curve is roughly equivalent to cumulative progress of $d \cdot \xi_{max}$, where $\xi_{max} = \min(\xi, 5)$. The parameter choice of K and D determines the slope and shift of the TRL curve, ultimately determining the dynamics of the gradient in the optimization.

Another crucial part in order to model research progress, was to capture the stochastic nature of technological enhancements. This was accomplished by estimating the underlying probability distribution using citation data from the different research areas. The queries used to collect the citation data are accounted for in appendix B, and the filtration is listed in table 4. The methodology used to collect the citation data was to vary the year of publication in order to obtain the largest possible dataset with less than $20,000$ data points, which was the maximum allowed amount to export. Since the estimation is based on the parameter $c_f = c/c_0$, which is a citation count standardized with the median of the number of citations on the year of publication, the varying time interval between the different data sets is believed to have no impact on the parameterization. This is confirmed by figure 20, where every log-normal distribution seems to follow a general distribution.

6.2.2 Constructing the Conversion Matrix C

A crucial part of the game is the mapping of technologies to battle parameters in the combat model, determining the player's military capabilities given their maturity in the technology portfolio. The mapping is described in section 4.2.3. The numerical values used in the mapping affect how important each technology is in order to gain an advantage on the battlefield. This means that these numerical values will control the impact that technology investments have on the battlefield, and which technologies that yield the greatest pay-off. The numerical scale used for this is exponential, implicating that big differences are made between what is defined as *Minor* and *Crucial* and the players will be drawn towards investing in technologies categorized as *Crucial*. The decision to employ an exponential scale for the mapping was based on an interview conducted with Jouannet (2023). Specifically, a non-linear scale was suggested to create a clear demarcation between items ranked as unimportant and those deemed critical. This was deemed necessary in assisting the model to identify realistic decision-making strategies. This decision was not primarily backed by previous academic research but rather by the experience of engineers more familiar with the subject area and the problems that may arise from having less differentiated classifications. Assessing the impact of this decision is complex and a full rerun of the game would be necessary to compare the results. For this reason, alternative choices of scale have not been investigated as part of this thesis. The choice was instead informed by subject matter experts with relevant experience in similar problems.

The method of calculating each element in the conversion matrix C, accounted for in section 4.3.3, was to use the mode of the set of survey answers. This was done, similar to the reasoning of the exponential scale, in order to maintain a clear demarcation between items ranked as unimportant and those deemed critical. An alternative to mode was to use the mean of the set of survey answers. By using the mean would create a smaller difference between the ranking of the technologies, which in turn would affect the decision-making of the players.

6.2.3 Combat Model

In this study, a modified stochastic Salvo combat model is used to evaluate the combat ability of the two players. This value is what is targeted during the optimization and thus has to be computationally efficient enough to be evaluated a large number of times in every iteration and be zero-sum, meaning player A will try to maximize its value, and player B will try to minimize it. In order to fulfill these criteria many assumptions and simplifications are made to allow for implementation, optimization, and evaluation in the limited time available, as well as interoperability with the other parts of the larger model presented in this study. The stochastic combat model outputs distributions for the remaining fighting forces on each side. Using these one can measure the probability of either side "winning" in several different ways. For example, the probability of player A having more than zero units remaining after a single Salvo has been used in this study. Another example would be to measure the relative size of the losses on both sides, called the force exchange ratio presented by Armstrong (2005), or some calculation based on the probability of each side surviving the Salvo.

The outlined ways of evaluating the objective function would likely yield very different values but it is still not known what impact this choice has on partial derivatives with respect to individual battle parameters and by extension the results of the game. Initially, it was planned to simulate multiple salvos being fired and sample a remaining number of units as input for the next iteration. This idea was eventually scrapped as so many safeguards would have to be added to ensure consistent behavior of the objective function that the iterative and dynamic would be excessively compromised.

The stochastic Salvo combat model uses the probability of offensive and defensive robots hitting their targets as a parameter. As such, tasks are key applications of certain technologies studied in this thesis a method of modeling these probabilities using the state of technologies has been developed using sigmoid functions, one for offensive probabilities p_α, p_β and one for defensive probabilities p_y, p_z, which map their inputs to a value between 0 and 1, the natural range of a probability. The functions used for this purpose are identical for the purpose of limiting the number of parameters. The two parameters used are set to give show clear progression in these parameters throughout the game but never reaching 1. Depending on how the parameters for these functions are chosen the dynamics of the simulated battle would likely be different on of the shifting importance of offensive and defensive strength.

The pivotal insight put forth by Armstrong (2014) is the significance of initiative in determining the outcome of a combat scenario. To incorporate this factor into the stochastic Salvo combat model, two additional parameters, namely ϕ, and ψ, have been introduced for each player. These modifications serve to account for the players' abilities to gather and analyze information regarding enemy movements, strengths, and weaknesses from long-range. As a result, they gain an informational advantage, thereby enhancing their prospects of entering the battlefield with the initiative. This logic is encapsulated by constructing a normal distribution (4.5) that represents the disparity in the players' abilities, and subsequently evaluating the cumulative probability of this difference surpassing a critical threshold. The parameter c can be adjusted to modify the sensitivity of the function. This framework primarily relies on the Central Limit Theorem, assuming that the player's ability to gather information comprises numerous systems and actions that follow a normal distribution on a case-by-case basis. This model is considered elegant as it requires only one parameter to be specified, allowing for the analytical construction of a CDF and the ability to alter the sensitivity using just a single parameter.

6.2.4 Optimization Hyperparameters

In order to identify logical and probable paths of technology development LSS, as presented by Mazumdar et al. (2019), was implemented to find actions that constitute (local) Nash equilibrium (NE). Such points are characterized by being locally optimal for each player and as such neither have an incentive to alter their action. Because this a relatively new technique there are limited examples of it being used in practice and no example of it being used in a problem of similar complexity has been found. This means the choice of hyperparameters (see table 8) used in this study was done so without much guidance from previous work, but instead set with a process of trial and error measuring the rate of convergence and inspecting variables throughout the optimization phase.

Executing a full run of the game is a computationally expensive task. By utilizing the hyperparameters presented in table 8, the simulation requires almost a week to complete on a PC with specifications outlined in table 9. The selection of appropriate hyperparameters is a critical aspect of the process and often involves a trade-off between the speed of execution and the quality of the output.

The first hyperparameter was the time horizon T, determining how many time steps the game will include, which is equivalent to the depth of the simulation. The value of T will influence the number of nodes in the graph of states that is spanned during the game since the number of nodes grows exponentially by the number of time steps. This means that the value of T is limited by the computational power. The second hyperparameter was the number of random actions I_ξ initiated from every node, which governs the exploration of the solution space. A higher value of I_ξ means a greater number of identified equilibrium points using the LSS algorithm, and more potential investment scenarios are identified. However, with each new random action, optimization needs to be applied, which significantly increases computational time. The third hyperparameter was the maximum number of optimization iterations in the LSS algorithm, which governs the likelihood of convergence towards (local) NE points in the solution space. Although LSS guarantees convergence to a (local) NE, it does so by investigating attracting saddle points and moving away from points that are not NE. Therefore, it is crucial that the number of optimization iterations is large enough for LSS to discover a true NE. However, the computational time increases with the number of iterations, as LSS involves computationally intensive approximations of Jacobians and Hessians. I_ξ is set to the highest value possible while keeping the computation time reasonable. The maximum number of Monte-Carlo iterations M affects the dynamics of the game in a similar way to I_ξ; as M increases, the approximation of the expected value of the objective function gets more true, improving the decision-making of the players, but increases the computational load of the game. The last hyperparameter was the maximum number of actions allowed to take from each node, I^*_{max}, which purpose is to limit the computational time of the game. This parameter affects how fast the graph of nodes grows since the graph grows exponentially with this number

6.2.5 Solution Methodology

The developed framework uses an optimization model to find different Nash equilibrium (NE) of technology investments between the players. This optimization model is designed to optimize for one action at a time – iterating until an equilibrium point is found. This means that the actions that the players take during the course of the game are *not* optimal in the sense of reaching an optimal state at $t = T$, where T is the time at the end of the game. Instead, each action that the optimization model produces is optimal based on the state that is currently evaluated, and not for any future state. This means that the players cannot formulate any investment strategies that plan for the future or strategies that are optimal later in the game, which is a property of the model that is not true to reality. However, for the purpose of this study where the goal is to predict technological development, this solution methodology is deemed reasonable since the optimization aims to find as many NE as possible from each state, where each NE is assumed to be equivalent to a possible real-life scenario of technology investments. By clustering groups of identified NE, trajectories of more common strategies can be obtained and analyzed subsequently.

An alternative solution methodology that could have been used to simulate the game between the players, and to identify possible scenarios of technology investments, is deep Q-learning whereby a neural network is fitted to generate optimal investments based on the current technology research of both players as well as the opponent's current investment choices. This network could then be repeatedly called until the two players *agree* on mutually optimal investments and step the simulation forward. This method only requires the neural net to be fitted once, perhaps with the data generated as part of this study used as a starting point. This method would have the added benefit of retaining the networks used to make decisions, The network could then be closely examined to further understand the dynamics of the game. Deep Q-learning and other methods that rely on neural networks are widely used in the field of game theory, especially in complex games where self-play can be leveraged to simultaneously generate training data and fit on that data.

6.2.6 Filter Actions, F_u

As reported in section 4.4.2, the filtration function $F_u : \mathcal{U}_i^{NE} \rightarrow \mathcal{U}_i^*$ was introduced to reduce the set of NE actions and obtaining a manageable set of actions to update the state with. Without the filtration function, the number of nodes in the graph of states that is spanned during the game would grow exponentially with the number of the parameter I_ξ. The first step in the filtration function is a PCA, that reduces the number of dimensions of the actions in order to cluster them efficiently. Here, a threshold of explained variance was set in order to ensure that we retain a sufficient amount of information. This threshold represents the minimum amount of variance that we want to retain in the reduced dataset of actions. By choosing an appropriate threshold, we could ensure that the quality of the dimension reduction was high enough and that not too much information was lost. If the threshold should be set too high, we may retain too many components, which can result in overfitting and poorer clustering results. On the other hand, if the threshold should be set too low, we may lose too much information and the reduced dataset may not be representative of the original dataset. In our case, after analyzing the data and considering our objectives, we decided that a threshold of 80% was an appropriate value since it retained enough information to ensure that the dimension reduction was high-quality, while also reducing the dimensionality of the dataset significantly.

The second part of the filtration function was to cluster the reduced actions using the K-means algorithm (see section 3.6.2). The methodology for clustering a set of NE points may seem counterintuitive since the center of a cluster of NE is unlikely to be a NE itself. However, by letting the centers of the clusters of NE points represent the actions that the players are taking, we could capture a collection of general directions or strategies that the players are more likely to follow. The centers represent a part of the action space that is more probable than other parts, and by increasing the hyperparameter I_ξ, representing the number of random starting points initiated from each node, the approximation of these centers was improved.

6.2.7 Implementation Environment

All the code in this study was written in Python 3.10. This choice was made for the purpose of development ease and speed, as well as access to commonly-used libraries in deep learning applications, such as PyTorch and TensorFlow. The main features of these libraries, including tensor operations and backpropagation, were heavily relied upon for the implementation of this study.

After careful consideration, PyTorch was chosen as the primary library for implementing the game. PyTorch is written in C++ and supports dynamic computational graphs, which was seen as a significant benefit compared to TensorFlow since the exact formulation of the objective function was not yet known at the start of the implementation phase.

The implementation process was generally smooth, but issues arose when utilizing some of the library's more exotic features, such as just-in-time (JIT) compilation and compilation to C++, as part of an attempt to optimize execution. After some minor adjustments, the game was able to run on both CPU and GPU, but it was found that CPU execution was faster. This was likely because the dimensionality was too low to leverage the thousands of CUDA cores available. Another benefit of CPU execution was the ability to parallelize it into multiple processes running simultaneously. A similar solution was attempted with the GPU, but it never worked as intended.

6.3 Future Research

This study proposes a quantitative methodology for allocating resources to research and development applied to the defense industry. There is nothing inherent in the method that requires the defense industry to be used. One could perform the same kind of mapping from technologies to product characteristics and then use these product characteristics to model market share in a particular product category. As long as research and development is a primary driver for market share and products are generic enough to be compared across different manufacturers this would likely be a worthwhile experiment.

The combat model used in this study is by design very simple and interpretable. These are not words commonly associated with real-world combat or warfare. An actor with significantly more resources than were available for this study could possibly build on the ideas of this study to expand the number of battle parameters and make more accurate simulations of real-world combat.
As already mentioned, an interesting approach to this problem would have been to use another solution methodology to find (local) Nash equilibrium (NE). As described in section 6.2.5, deep Q-learning is an interesting alternative to the optimization model presented in this thesis.

6.4 Ethical Aspects

The purpose of this thesis was to develop a quantitative framework for technology road-mapping, applying game-theoretic concepts in order to model rational decision-making for technology investments. The framework has been developed to be applied in the field of tactical autonomy, thus raising several ethical considerations. Tactical autonomy refers to the development of autonomous systems that can perform military tasks, including surveillance, reconnaissance, and combat operations. As such, any technologies developed through this research can potentially be applied for military and defense purposes, which raises several questions about the ethical implications of such applications.

One ethical consideration is the possibility of enhanced development of autonomous technologies purposed for military applications, and the potential consequences of using these technologies in warfare. Autonomous systems have the potential to reduce the risk of harm to military personnel by eliminating the need for human involvement in certain tasks. However, their use can also lead to civilian casualties and human rights violations, particularly if they are used inappropriately or without proper oversight. The view taken by SAAB and the authors of this thesis is that autonomous weapons systems shall only be deployed if they are deemed safe, reliable, and well-suited for their tasks. Getting to that point is a problem of research and development. Limiting this research will likely lead to the deployment of systems that are less safe and less reliable.

Finally, the mathematical formulations used in the proposed model should hopefully generate solutions that lead to potential scenarios of technology development, based on rational decisions made by the players in the game. However, no mathematical model aimed at describing the real world can do so perfectly, and it will never be better than the assumptions used in the model – and there have been a lot of assumptions made during the development of this model. Relying wholeheartedly on the outputs of the model, without applying common sense and the knowledge of field experts should always be avoided, and decision-making based on evidence provided by the model should always be made with the consideration that the complexity of real-world processes does not follow the necessary simplifications encapsulated in the mathematical formulation.

6.5 Conclusions

To address the purpose of this thesis, a quantitative framework has been developed, integrating crucial concepts from game theory and probability theory. In order to accomplish this, several questions need to be answered throughout the development process. Firstly, how can game theory be applied to model technological development quantitatively? The answer to this question can be found by studying Nash equilibrium (NE), and formulating the problem of technological development as an investment problem where two players try to maximize opposing reward functions in a zero-sum environment, where the possible solutions are (local) NE. We show in the proposed method that game-theoretic concepts can be used to model the decision-making of the players in the game, and to generate different scenarios of technological development. Another question was how the probability distribution describing the underlying randomness of research progress, was to be estimated. This was done using a method developed by Radicchi et al. (2008), using citation data to model the probability of scientific impact for different research areas. The results of the estimation show that the citation data successfully was fitted to a log-normal probability distribution at a significance level of 5%. This meant that the stochasticity underlying scientific breakthroughs could be modeled and introduced in the developed model. Thirdly, how to find the equilibria points in sequential, two-player zero-sum games serving as potential solutions of technology investments of the game. The method produced in this study proposes an optimization model using the LSS algorithm developed by Mazumdar et al. (2019) and designed for guaranteed convergence to (local) NE. The results show how the implementation of LSS successfully converges and produces the set of (local) NE, representing possible investments, from each evaluated state. Thus showing that (local) equilibria points can successfully be found in complex games operating in high-dimensional variable spaces.

As presented in the results, the developed framework produces a well-behaved game where the players seem to make reasonable technology investments with clear preferences for some technologies over others. However, the results of technology development should not be considered as any recommendation or be viewed as optimal in any real-world sense, as many assumptions have been made during the development and the model is a significant simplification of reality. Nonetheless, what we can conclude from the results of the technology road-map is that the potential of a quantitative framework, such as the one proposed in this study, exhibits a remarkable level of significance as an aid to the qualitative methods that are used in the field today. With further work, the framework can serve as a valuable tool to introduce a quantitative perspective to the decision-making process of technology resource allocation.

References

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *WIREs Computational Statistics*, *2*(4), 433–459. https://doi.org/https://doi.org/10.1002/wics.101

AFRL, T. A. R. L. (2021). About the tactical autonomy program [[Online; accessed February, 2023]].

Amyx, S. (2017). Ai in the military: Paradigm shift in warfare [[Online; accessed February, 2023]].

Armstrong, M. J. (2014). The salvo combat model with a sequential exchange of fire. *Journal of the Operational Research Society*, *65*(10), 1593–1601.

Armstrong, M. J. (2005). A stochastic salvo model for naval surface combat. *Operations Research*, *53*(5), 830–841.

Armstrong, M. J., & Powell, M. B. (2005). A stochastic salvo model analysis of the battle of the coral sea. *Military Operations Research*, 27–37.

Blom, G., Enger, J., Engblom, G., Grandell, J., & Holst, L. (2016a). *Sannolikhetsteori och statistikteori med tillämpningar* (Vol. 5:14). Studentlitteratur.

Blom, G., Enger, J., Engblom, G., Grandell, J., & Holst, L. (2016b). *Sannolikhetsteori och statistikteori med tillämpningar* (Vol. 5:14). Studentlitteratur.

Carvalho, M., Fleury, A., & Lopes, A. P. (2012). An overview of the literature on technology roadmapping (trm): Contributions and trends.

Center for Strategic and International Studies. (2023). Send in the swarm [[Accessed: March 31, 2023]].

Clark, B., Patt, D., & Schramm, H. (2020). Mosaic warfare exploiting artificial intelligence and autonomous systems to implement decision-centric operations.

Clark, I. G., & Olds, J. R. (2005). The technology roadmapping and investment planning system (trips).

DefenseNews. (2022). Top 100 defense companies [[Online; accessed February, 2023]].

Dick, K., Russell, L., Dosso, Y. S., Kwamena, F., & Green, J. R. (2019). Deep learning for critical infrastructure resilience. *Journal of Infrastructure Systems*, *25*(2), 05019003. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000477

Edwards, D. (2022). 7 roles of artificial intelligence in the defence sector [[Online; accessed February, 2023]].

Eliaçık, E. (2022). Guns and codes: The era of ai-wars begins [[Online; accessed February, 2023]].

Epstein, I. (2022). The evolution of war: How ai has changed military weaponry and technology [[Online; accessed February, 2023]].

FMV. (2022). Teknisk und informerar [[Online; accessed February, 2023]].

FOI. (2022). Militärteknik 2045, ett underlag till försvarsmaktens perspektivstudie [[Online; accessed February, 2023]].

Försvarsdepartementet. (2017). Slutredovisning av försvarsmaktens perspektivstudie 2016-2018.

Garcia, M. L., & Bray, O. H. (1997). Fundamentals of technology roadmapping.

Gentile, G., Shurkin, M., Evans, A. T., Grisé, M., Hvizda, M., & Jensen, R. (2021). A history of the third offset, 2014-2018.

Hagos, D. H., & Rawat, D. B. (2022). Recent advances in artificial intelligence and tactical autonomy: Current status, challenges, and perspectives.

Hughes, J., & Wayne, P. (1995). A salvo model of warships in missile combat used to evaluate their staying power. *Naval Research Logistics (NRL)*, *42*(2), 267–289.

Investopedia. (2023). Economic equilibrium. *Investopedia*. https://www.investopedia.com/terms/e/economic-equilibrium.asp.

Jarne, G., Sanchez-Choliz, J., & Fatas-Villafranca, F. (2007). "s-shaped" curves in economic growth. a theoretical contribution and an application.

Jim Banke, N. A., & Administration, S. (2010). Technology readiness levels demystified.

Jouannet, C. (2023).

Kumar, S. (2020). Silhouette method: Better than elbow method to find optimal clusters. *Towards Data Science*. https://towardsdatascience.com/silhouette-method-better-than-elbow-method-to-find-optimal-clusters-378d62ff6891

Kuo-Pin Yang, Y.-C. C., & Kuo, C.-C. (2010). The relationship between rd investment and firm profitability under a three-stage sigmoid curve model: Evidence from an emerging economy. *Transactions on Engineering Management*, *57*(1), 103–117.

Li, H., Farnia, F., Das, S., & Jadbabaie, A. (2022). On convergence of gradient descent ascent: A tight local analysis.

Loève, M. (1977). *Probability theory i*.

Lucas, T. W., & McGunnigle, J. E. (2003). When is model complexity too much? illustrating the benefits of simple models with hughes' salvo equations. *Naval Research Logistics (NRL)*, *50*(3), 197–217. https://doi.org/https://doi.org/10.1002/nav.10062

Luotsinen, L. J., Oskarsson, D., Svenmarck, P., & Bolin, U. W. (2019). Explainable artificial intelligence: Exploring xai techniques in military deep learning applications.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. M. L. Cam & J. Neyman (Eds.), *Proc. of the fifth berkeley symposium on mathematical statistics and probability* (pp. 281–297). University of California Press.

Mazumdar, E., Jordan, M., & Sastry, S. (2019). On finding local nash equilibria (and only local nash equilibria) in zero-sum games. https://doi.org/10.48550/ARXIV.1901.00838

Mcafee, R. P., & Mcmillan, J. (1996). Competition and game theory. *Journal of Marketing Research*, *33*(3), 263–267. https://doi.org/10.1177/002224379603300301

Myung, I. J. (2001). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, *47*, 90–100. https://doi.org/10.1016/S0022-2496(02)00028-7

Nagarajan, V., & Kolter, J. Z. (2017). Gradient descent gan optimization is locally stable. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 5591–5600.

Najera, J. (2019). The basics of game theory. *Towards Data Science*.

Nash, J. (1951). Non-cooperative games. *Annals of Mathematics*, *54*(2), 286–295.

NewsHour, P. (2023). Drone advances amid war in ukraine could bring fighting robots to front lines [[Accessed: March 31, 2023]].

Olivier, J. P. (2011). Strategic planning and prioritization methodology for major naval surface combatants conceptual design.

Pan, J.-X., & Fang, K.-T. (2002). Maximum likelihood estimation. In *Growth curve models and statistical diagnostics* (pp. 77–158). Springer New York. https://doi.org/10.1007/978-0-387-21812-0_3

Pettinger, T. (2023). Zero-sum games in economics. *Economics Help*. https://www.economicshelp.org/blog/167397/economics/zero-sum-games-in-economics/.

Phaal, R., Farrukh, C. J., & Probert, D. R. (2004). Technology roadmapping—a planning framework for evolution and revolution.

Radicchi, F., Fortunato, S., & Castellano, C. (2008). Universality of citation distributions: Toward an objective measure of scientific impact. *Proceedings of the National Academy of Sciences*, *105*(45), 17268–17272. https://doi.org/10.1073/pnas.0806977105

Ratliff, L. J., Burden, S. A., & Sastry, S. S. (2013). Characterization and computation of local nash equilibria in continuous games. *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 917–924. https://doi.org/10.1109/Allerton.2013.6736623

Rogers, E. M. (2003). *Diffusion of innovations* (5th). Free Press.

Ross, D. (2021). Game theory. In E. N. Zalta (Ed.), *The stanford encyclopedia of philosophy* (Fall 2021). Metaphysics Research Lab, Stanford University.

Saab. (2023a). Emerging technologies.

Saab. (2023b). Vår historia [[Online; accessed February, 2023]].

Shutaywi, M., & Kachouie, N. N. (2021). Silhouette analysis for performance evaluation in machine learning with applications to clustering. *Entrop (Basel, Switzerland)*, *20*(6), 759.

Smirnova, K., Golkar, A., & Vingerhoeds, R. (2018). A game-theoretic framework for concurrent technology roadmap planning using best-response techniques.

Smith II, J. D. (2005). An alternative to technology readiness levels for non-developmental item (ndi) software.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Von Neumann, J., & Morgenstern, O. (1966). *Theory of games and economic behavior* (3rd). Princeton University Press.

Wang, Y., Zhang, G., & Ba, J. (2019). On solving minimax optimization locally: A follow-the-ridge approach. *CoRR*, *abs/1910.07512*. http://arxiv.org/abs/1910.07512

Wirkuttis, H., N. Klein. (2017). Artificial intelligence in cybersecurity.

Yang, K.-P., Chiao, Y.-C., & Kuo, C.-C. (2019). Mosaic warfare: From philosophy to model to solutions. *Int Rob Auto*, *5*(5), 157–166.

Appendix

A The Silhouette Method

The silhouette method is a method to find the optimal number of clusters when classifying a data set into different clusters (Kumar, 2020). The method does so by computing *silhouette coefficients* of each point in the dataset, which is a measure of how much a point is similar to its own cluster compared to other clusters, ranging on the interval $[-1, 1]$. A high value indicates that the point is well-matched to its own cluster and poorly matched to neighboring clusters. If the majority of the points have a high value, then the clustering configuration is appropriate, and vice-versa. The *silhouette score* is defined by the average of all the silhouette coefficients in the data set. The silhouette coefficient $s(i)$ of a point i is defined by:

$$s(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))}, \tag{A.1}$$

where $a(i)$ is the average distance of point i with all other points in the same cluster, $b(i)$, is the average distance of point i with all the points in the closest cluster to its cluster. Then, the silhouette score can be calculated by:

$$S = \frac{1}{I} \sum_{i=1}^{I} s(i), \tag{A.2}$$

for points $i \in I$. By maximizing the silhouette score over the number of clusters, the optimal number of clusters can be obtained.

B Queries for Citation Data

Research area	**Query**
AI & Machine Learning	("artificial intelligence" OR "ai" OR "Machine Learning" OR "ml") AND ("military" AND ("naval" OR "combat" OR "aeronautics" OR "defense")) AND PUBYEAR > 2000 AND PUBYEAR < 2019 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))
Collaborative systems	("multi-agent" OR "multi-agent system" OR "multi-agent systems" OR "swarming" OR "swarms" OR "collaborative systems") AND ("naval" OR "combat" OR "aeronautics" OR "defense"))PUBYEAR > 2005 AND PUBYEAR < 2020 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))
Communications & networking	(("communication" AND "intelligence") OR ("communication" AND "networking") OR ("networking" AND "intelligence")) AND ("naval" OR "combat" OR "aeronautics" OR "defense")) PUBYEAR > 2000 AND PUBYEAR < 2019 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))
Control systems & algorithms	(("Control systems") OR ("algorithms")) AND ("military" AND ("naval" OR "combat" OR "aeronautics" OR "defense")) PUBYEAR > 2005 AND PUBYEAR < 2020 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))
Cybersecurity	(("cyber security")) PUBYEAR > 2005 AND PUBYEAR < 2019 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))
Edge computing	(("edge computing")) PUBYEAR > 2005 AND PUBYEAR < 2019 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))
Energy Management	(("energy" AND "management") OR ("power" AND "efficiency") OR ("energy" AND "efficiency")) AND ("military" AND ("naval" OR "combat" OR "aeronautics" OR "defense")) PUBYEAR > 2000 AND PUBYEAR < 2020 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))

Ethics & Regulations	(("ethics") OR ("regulations") OR ("doctrine")) AND ("military" AND ("naval" OR "combat" OR "aeronautics" OR "defense")) PUBYEAR > 2005 AND PUBYEAR < 2019 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))
Human Machine Interaction	(("human machine interaction") OR ("HMI")) PUBYEAR > 2005 AND PUBYEAR < 2019 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))
Localization & mapping	(("localization and mapping")) PUBYEAR > 2005 AND PUBYEAR < 2019 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))
Sensor Fusion	(("sensor fusion")) PUBYEAR > 2010 AND PUBYEAR < 2019 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))
Sensor Technologies	(("sensor")) AND ("military" AND ("naval" OR "combat" OR "aeronautics" OR "defense")) PUBYEAR > 2000 AND PUBYEAR < 2020 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))
Simulation & modeling	(("simulation" OR "modeling" OR "simulate")) AND ("military" AND ("naval" OR "combat" OR "aeronautics" OR "defense")) PUBYEAR > 2010 AND PUBYEAR < 2018 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))
Technologies enabling mobility	("robotics" OR "robotic" OR "robots" OR "robot") AND ("military" AND ("naval" OR "combat" OR "aeronautics" OR "defense")) PUBYEAR > 2000 AND PUBYEAR < 2020 AND (LIMIT-TO (DOCTYPE , "cp") OR LIMIT-TO (DOCTYPE , "ar"))

C Survey for Parameterization

Parameterisation of "Battle Parameters"

This survey is part of a master thesis that aims to develop a quantitative framework for technology road-mapping within the field of autonomy, using game theory. The framework can then be used to simulate possible scenarios and thus be helpful in Saab's processes to road-map future R&D-projects. The thesis is part of FCAS *Tactical Autonomy*, and is supervised by Marina Rantanen Modéer, Saab Dynamics AB.

Aim of the survey:

Part 1 (question 1-6): An important part of the project is to assess the impact of different autonomous technologies on a battlefield scenario. In order to do this, we will ask you to rate the impact that different technologies have on a specific "Battle parameter" (parameters that will be used to simulate a battle between two military actors) in a combat simulation. A "Battle parameter" in this context can be interpreted as a military capability on the battlefield.

Part 2 (question 7): Next step is to initiate a starting state for each technology, representing how far the development has come in its current technological paradigm. In order to do this, we will use "sigmoid-curves" (also known as "S-curves"), and we will ask you to assess the maturity of each technology by placing the technology on a sigmoid-curve.

We understand that this is a difficult problem affected by a high degree subjectivity, but we ask you to try your best to rate the impact of each technology by making assumptions based on your expertise.

In the next two sections, a description of the technologies, and the "Battle parameters" is presented. Then follows a series of questions.

Technology description

- **Sensor Technologies:** Sensors, such as cameras, lidar, and radar, can be used to provide real-time information about the environment and help machines and systems make decisions.
- **Collaborative systems**: Heterogeneous and homogeneous multi-agent systems involve the coordination of multiple autonomous agents to achieve a common goal, such as searching an area or performing a task. For example swarming systems.
- **Technologies enabling mobility**: Technologies referring to hardware-based autonomous systems that enables transportation from A to B, that can be used in a variety of tactical settings, such as search and rescue, operating in hazardous environments, and offensive/defensive military operations
- **Control systems and algorithms:** Control systems and algorithms are used to enable autonomous systems to make decisions and take actions based on sensor inputs and other data.
- **Localization and mapping**: Autonomous systems need to be able to accurately locate themselves within an environment and map out their surroundings in order to make decisions and navigate effectively.
- **Sensor Fusion:** Sensor fusion technologies are used to combine data from multiple sensors in order to obtain a more accurate and comprehensive understanding of the environment.
- **AI and Machine Learning:** These technologies can be used to enable machines and systems to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and language translation, aswell as to train algorithms to learn patters and relationships in data, rather than being explicitly programmed to perform a specific task.
- **Edge computing:** Edge computing refers to the processing and analysis of data on the edge of the network, closer to where the data is generated, rather than sending it to a central server.
- **Communications and networking:** Reliable and secure communication and networking technologies are crucial for enabling communication and coordination between different tactical units and systems.
- **Energy Management:** Energy management technologies are important for enabling autonomous systems to operate for long periods of time without needing to be recharged or refueled.
- **Simulation and modeling:** Simulation and modeling can be used to test and refine autonomous systems before they are deployed in the field, as well as to train operators and test different scenarios.
- **Human Machine Interaction:** Developing intuitive and effective interfaces between humans and autonomous systems is crucial for enabling effective collaboration and decision-making in tactical settings.
- **Cybersecurity:** As autonomous systems become more prevalent in tactical settings, ensuring their security and resilience against cyber attacks becomes increasingly important.
- **Ethics and Regulations:** As autonomous systems become more widespread, there is a growing need to develop ethical frameworks and regulations to ensure that they are used safely and responsibly.

Description of Battle Parameters

- **Intelligence gathering**: measures the information advantage a player is expected to have over their opponent.
- **Number of offensive actions taken per time and unit**: The rate at which a single unit is able to complete one iteration of OODA-loop (Observe, Orient, Decide, Act) and possibly carrying out an offensive action.
- **Probability of offensive actions being accurate**: The probability of an offensive action succeeding in its goal, assuming no defensive intervention of the opposing player.
- **Number of defensive actions taken per time and unit**: The rate at which a single unit is able to complete one iteration of OODA-loop (Observe, Orient, Decide, Act) and possibly carrying out a defensive action
- **Probability of defensive actions being successful**: The probability of a defensive action to effectively repel or minimise damage suffered from an offensive action taken by the opposing player.

Battle parameter, player 1 & 2	Description
ϕ, ψ	Intelligence gathering
n_α, n_β	Number of offensive actions taken per time and unit
p_α, p_β	Probability of offensive actions being accurate
n_y, n_z	Number of defensive actions taken per time and unit
p_y, p_z	Probability of defensive actions being successful

Parameterisation of Battle parameters

Start from the description of the technologies and the "Battle parameters", and assess to which extent a technological improvement would impact the military capability.

TIP: open a separate window with the description of the technologies and the "Battle parameters" when answering the questions.

1

What impact would a technological improvement have on the capability **"Intelligence gathering"**, for every technology area.

	N/A	Minor	Major	Critical
Sensor Technology	◯	◯	◯	◯
Collaborative systems	◯	◯	◯	◯
Technologies enabling mobility	◯	◯	◯	◯
Control systems and algorithms	◯	◯	◯	◯
Localization and mapping	◯	◯	◯	◯
Sensor Fusion	◯	◯	◯	◯
AI and Machine Learning	◯	◯	◯	◯
Edge computing	◯	◯	◯	◯
Communications and networking	◯	◯	◯	◯
Energy Management	◯	◯	◯	◯
Simulation and modeling	◯	◯	◯	◯
Human Machine Interaction	◯	◯	◯	◯
Cybersecurity	◯	◯	◯	◯
Ethics and Regulations	◯	◯	◯	◯

2

What impact would a technological improvement have on the capability **"Number of offensive actions taken per time and unit"**, for every technology area.

	N/A	Minor	Major	Critical
Sensor Technology	◯	◯	◯	◯
Collaborative systems	◯	◯	◯	◯
Technologies enabling mobility	◯	◯	◯	◯
Control systems and algorithms	◯	◯	◯	◯
Localization and mapping	◯	◯	◯	◯
Sensor Fusion	◯	◯	◯	◯
AI and Machine Learning	◯	◯	◯	◯
Edge computing	◯	◯	◯	◯
Communications and networking	◯	◯	◯	◯
Energy Management	◯	◯	◯	◯
Simulation and modeling	◯	◯	◯	◯
Human Machine Interaction	◯	◯	◯	◯
Cybersecurity	◯	◯	◯	◯
Ethics and Regulations	◯	◯	◯	◯

3

What impact would a technological improvement have on the capability **"Probability of offensive actions being accurate"**, for every technology area.

	N/A	Minor	Major	Critical
Sensor Technology	○	○	○	○
Collaborative systems	○	○	○	○
Technologies enabling mobility	○	○	○	○
Control systems and algorithms	○	○	○	○
Localization and mapping	○	○	○	○
Sensor Fusion	○	○	○	○
AI and Machine Learning	○	○	○	○
Edge computing	○	○	○	○
Communications and networking	○	○	○	○
Energy Management	○	○	○	○
Simulation and modeling	○	○	○	○
Human Machine Interaction	○	○	○	○
Cybersecurity	○	○	○	○
Ethics and Regulations	○	○	○	○

4

What impact would a technological improvement have on the capability **"Number of defensive actions taken per time and unit"**, for every technology area.

	N/A	Minor	Major	Critical
Sensor Technology	◯	◯	◯	◯
Collaborative systems	◯	◯	◯	◯
Technologies enabling mobility	◯	◯	◯	◯
Control systems and algorithms	◯	◯	◯	◯
Localization and mapping	◯	◯	◯	◯
Sensor Fusion	◯	◯	◯	◯
AI and Machine Learning	◯	◯	◯	◯
Edge computing	◯	◯	◯	◯
Communications and networking	◯	◯	◯	◯
Energy Management	◯	◯	◯	◯
Simulation and modeling	◯	◯	◯	◯
Human Machine Interaction	◯	◯	◯	◯
Cybersecurity	◯	◯	◯	◯
Ethics and Regulations	◯	◯	◯	◯

5

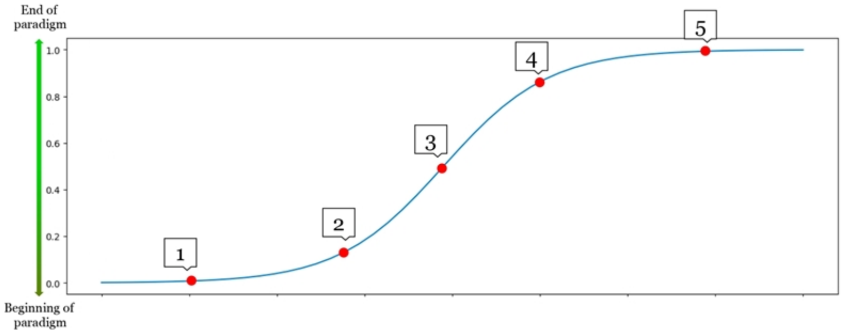
What impact would a technological improvement have on the capability **"Probability of defensive actions being successful"**, for every technology area.

	N/A	Minor	Major	Critical
Sensor Technology	○	○	○	○
Collaborative systems	○	○	○	○
Technologies enabling mobility	○	○	○	○
Control systems and algorithms	○	○	○	○
Localization and mapping	○	○	○	○
Sensor Fusion	○	○	○	○
AI and Machine Learning	○	○	○	○
Edge computing	○	○	○	○
Communications and networking	○	○	○	○
Energy Management	○	○	○	○
Simulation and modeling	○	○	○	○
Human Machine Interaction	○	○	○	○
Cybersecurity	○	○	○	○
Ethics and Regulations	○	○	○	○

6

Where in its current technological paradigm (represented by the sigmoid-curve) would you assess technology *n* to be?

Here, paradigm is defined as a particular phase of technological development that is characterized by a dominant technology. When a paradigm reaches maturity, no progress is expected to be made without a major breakthrough that changes the dominant set of assumptions, concepts, or methods within the field.

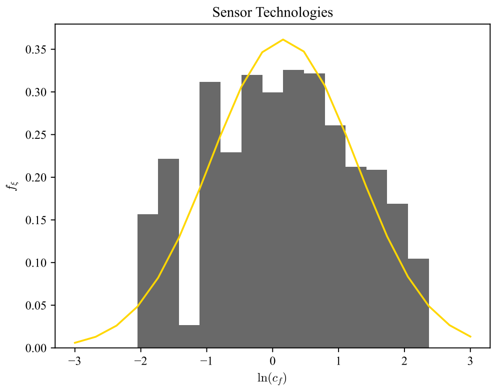


	1	2	3	4	5	Pass
Sensor Technology	◯	◯	◯	◯	◯	◯
Collaborative systems	◯	◯	◯	◯	◯	◯
Technologies enabling mobility	◯	◯	◯	◯	◯	◯
Control systems and algorithms	◯	◯	◯	◯	◯	◯
Localization and mapping	◯	◯	◯	◯	◯	◯
Sensor Fusion	◯	◯	◯	◯	◯	◯
AI and Machine Learning	◯	◯	◯	◯	◯	◯
Edge computing	◯	◯	◯	◯	◯	◯
Communications and networking	◯	◯	◯	◯	◯	◯
Energy Management	◯	◯	◯	◯	◯	◯
Simulation and modeling	◯	◯	◯	◯	◯	◯

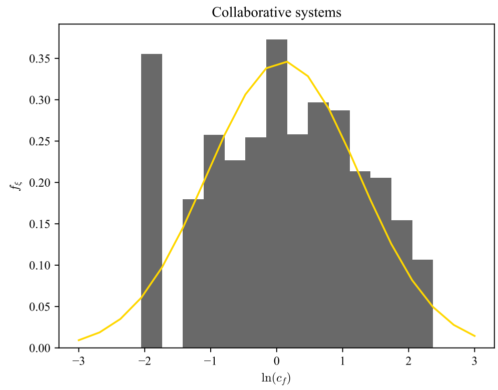
Human Machine Interaction	◯	◯	◯	◯	◯	◯
Cybersecurity	◯	◯	◯	◯	◯	◯
Ethics and Regulations	◯	◯	◯	◯	◯	◯

D Figures of Results

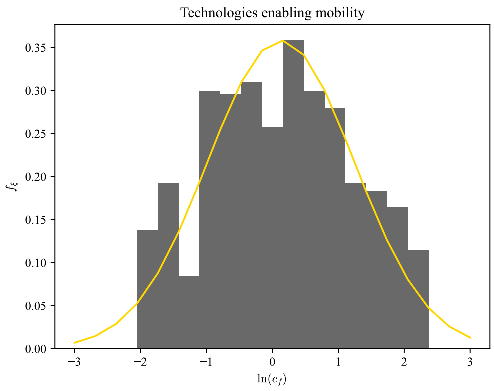
D.1 Research Progress Distributions



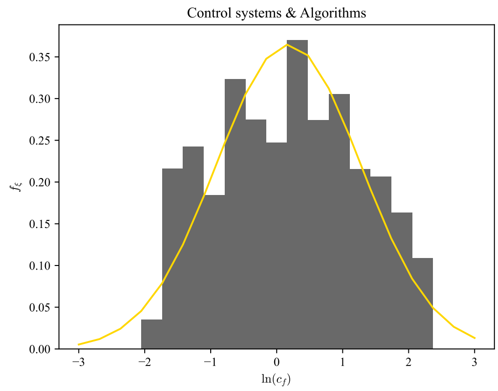
(a) Sensor Technologies



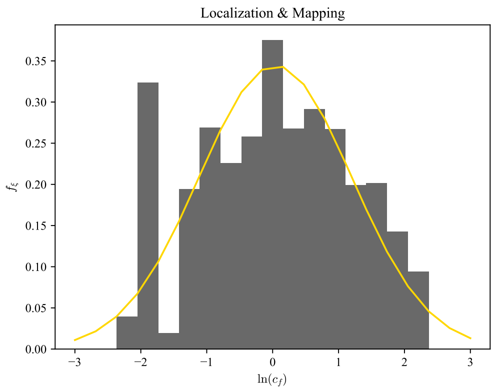
(b) Collaborative systems



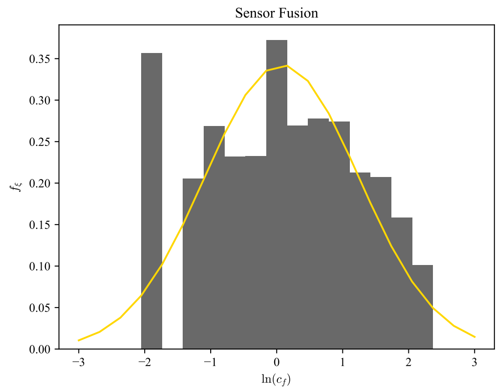
(c) Technologies enabling mobility



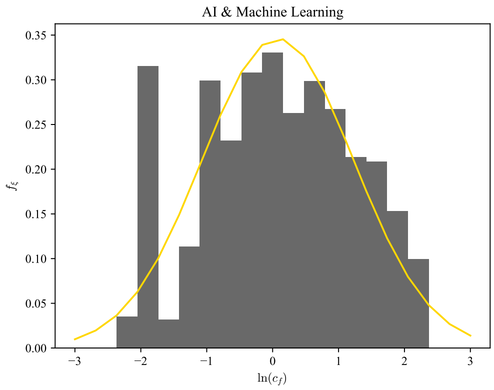
(d) Control systems & Algorithms



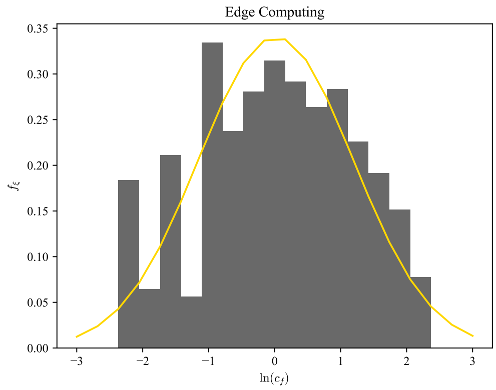
(e) Localization & Mapping



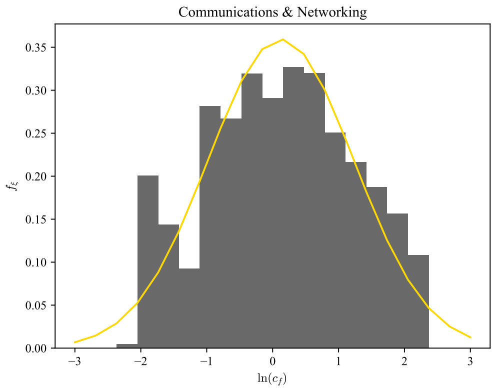
(f) Sensor fusion



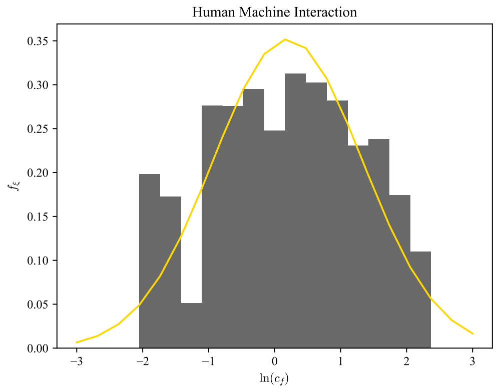
(g) AI & Machine Learning



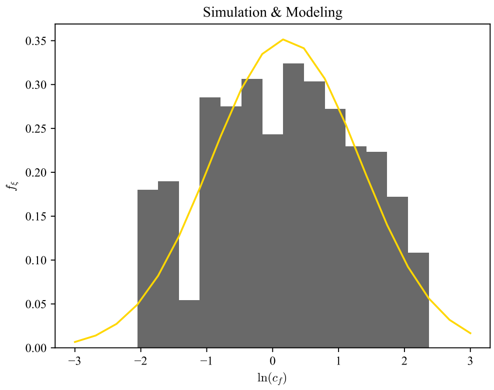
(h) Edge Computing



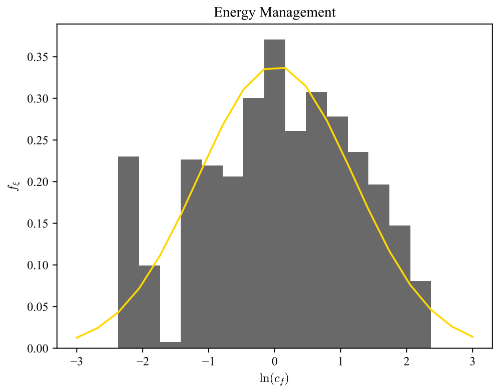
(i) Communications & Networking



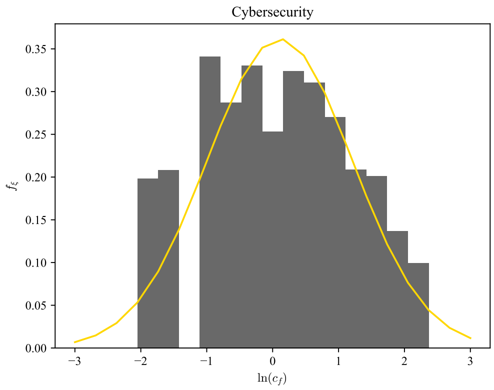
(j) Energy Management



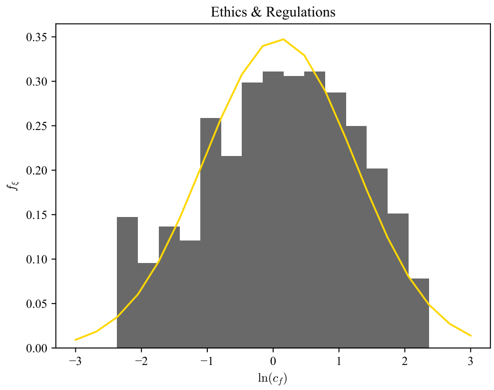
(k) Simulation & Modeling



(l) Human Machine Interaction



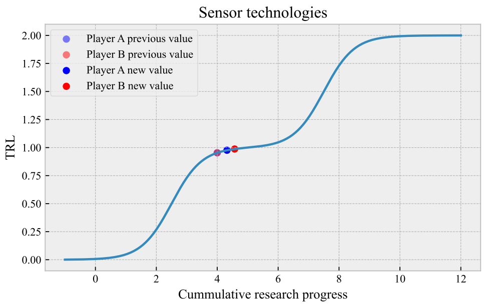
(m) Cyber security



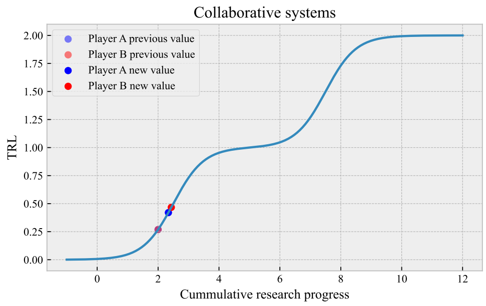
(n) Ethics & Regulations

Figure 29: Presentation of the fit of the estimated log-normal probability distributions describing research progress.

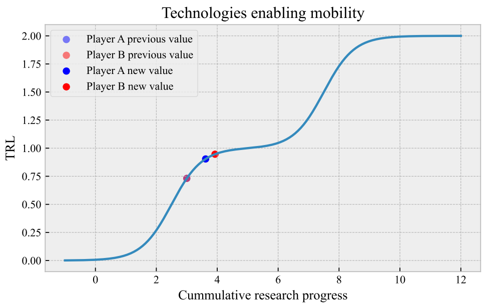
D.2 Technology Development TRL-curves



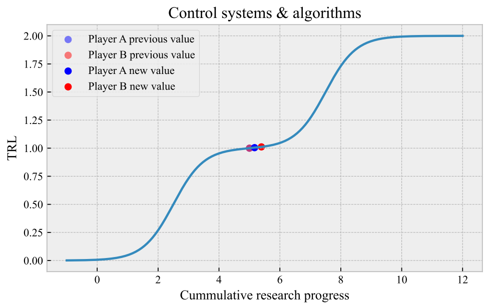
(a) Sensor Technologies



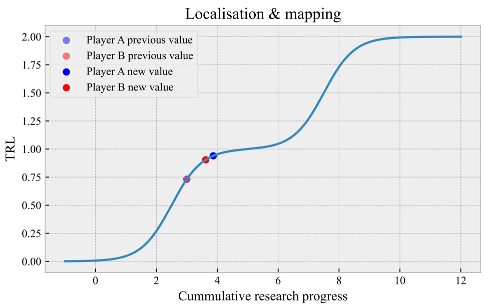
(b) Collaborative systems



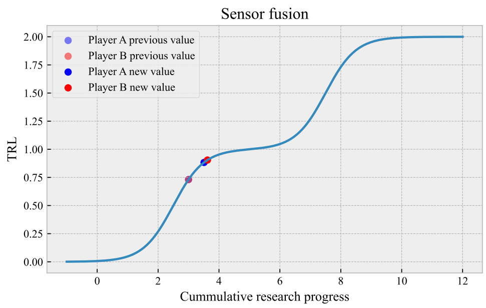
(c) Technologies enabling mobility



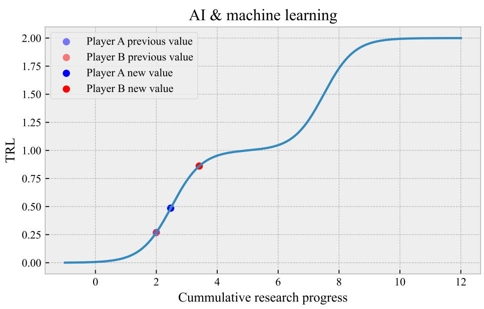
(d) Control systems & Algorithms



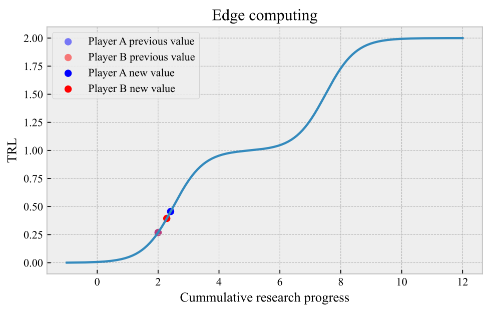
(e) Localization & Mapping



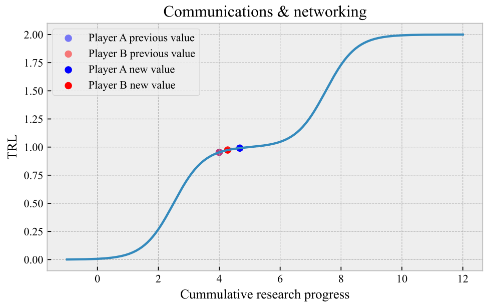
(f) Sensor fusion



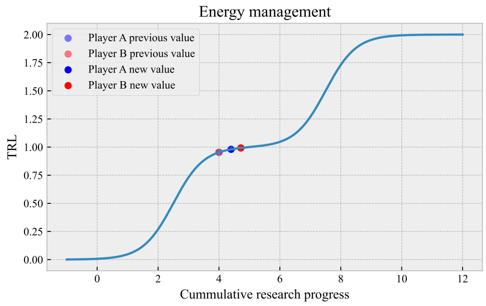
(g) AI & Machine Learning



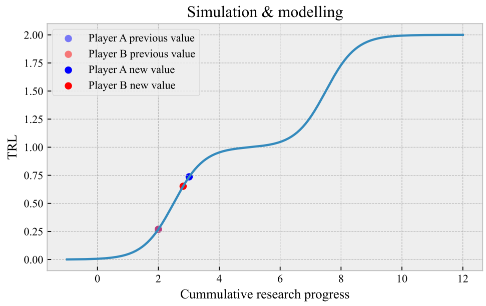
(h) Edge Computing



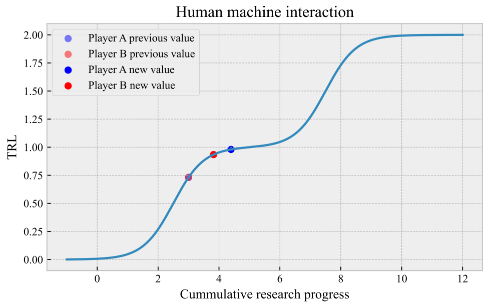
(i) Communications & Networking



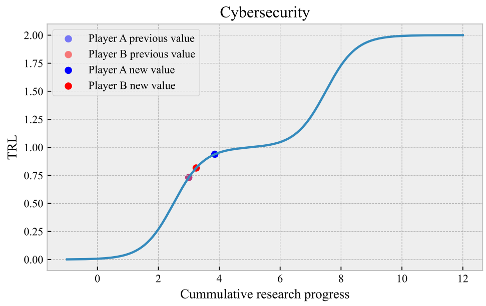
(j) Energy Management



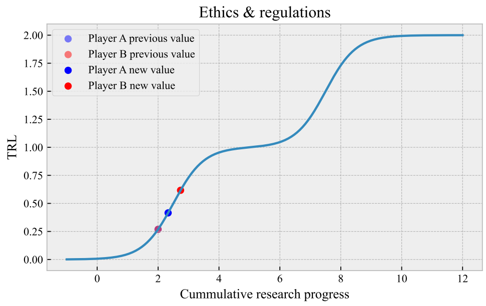
(k) Simulation & Modeling



(l) Human Machine Interaction



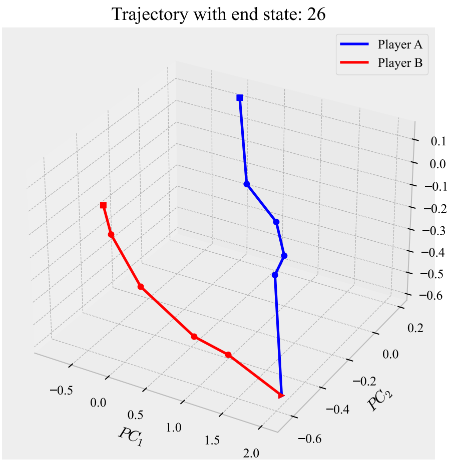
(m) Cyber security



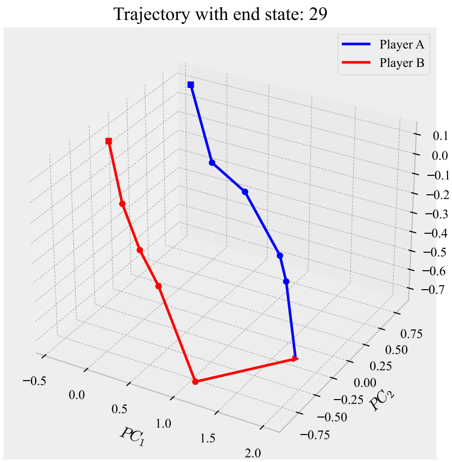
(n) Ethics & Regulations

Figure 30: Presentation of resulting TRL-curves, showing the evolution of the TRL during the game

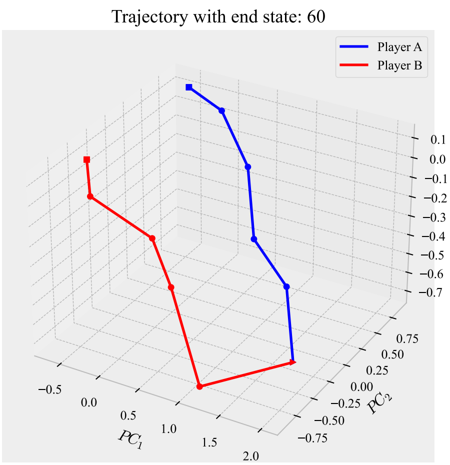
D.3 Trajectories



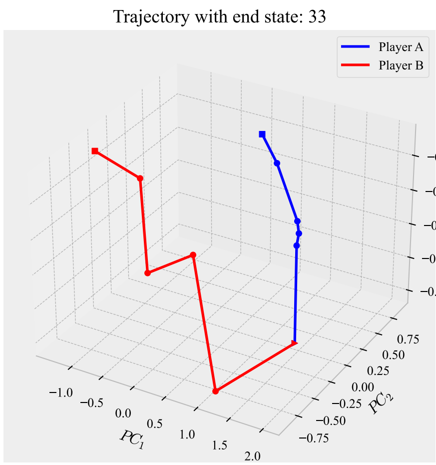
(a)



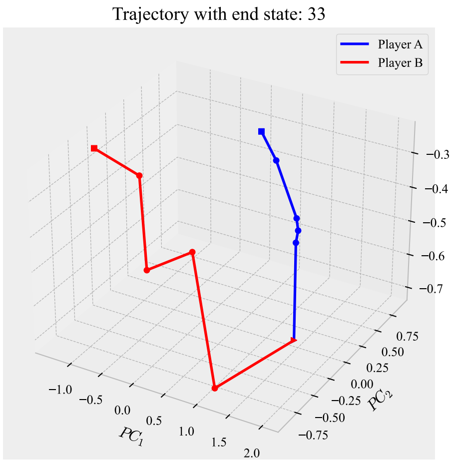
(b)



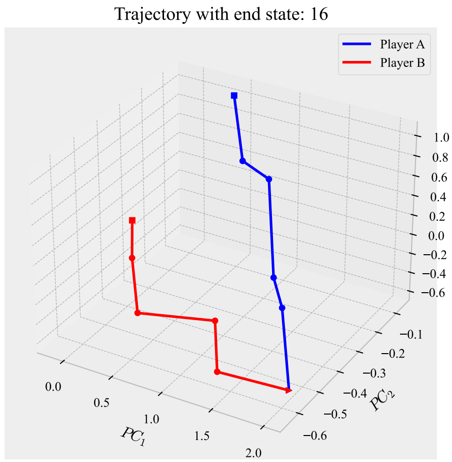
(c)



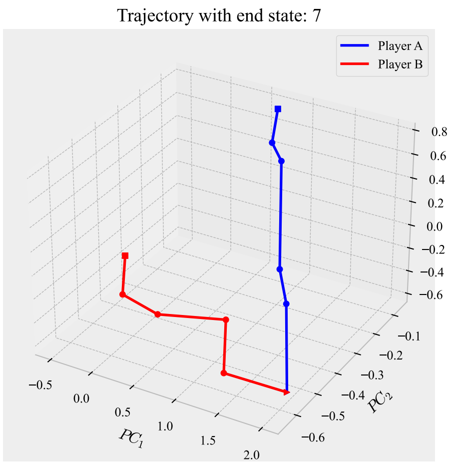
(d)


Trajectory with end state: 33
Player A
Player B
PC_1
PC_2
PC_3

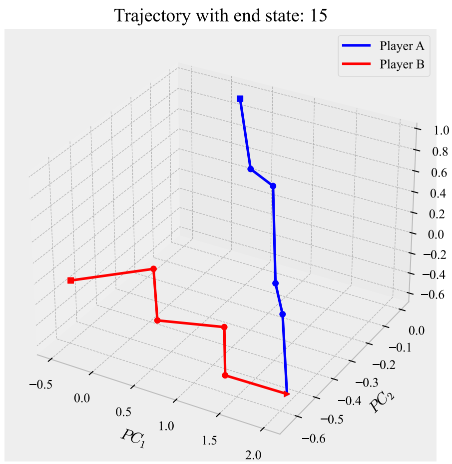
(e)


Trajectory with end state: 16
Player A
Player B
PC_1
PC_2
PC_3

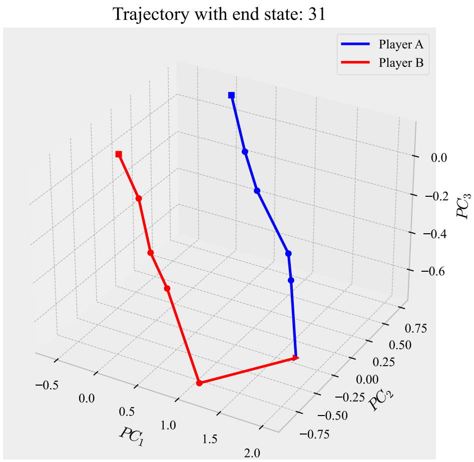
(f)


Trajectory with end state: 7
Player A
Player B
PC_1
PC_2
PC_3

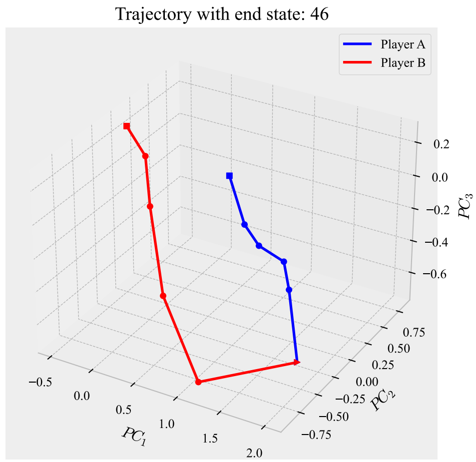
(g)


Trajectory with end state: 15
Player A
Player B
PC_1
PC_2
PC_3

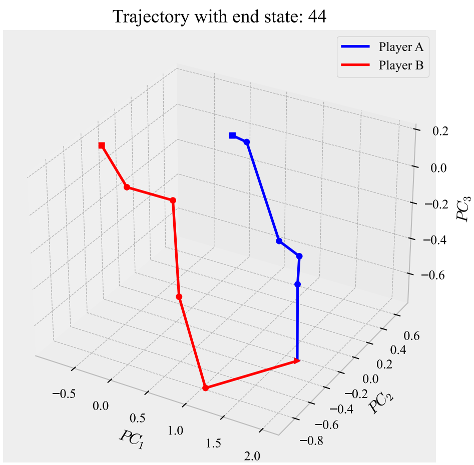
(h)


Trajectory with end state: 31
Player A
Player B
PC_1
PC_2
PC_3

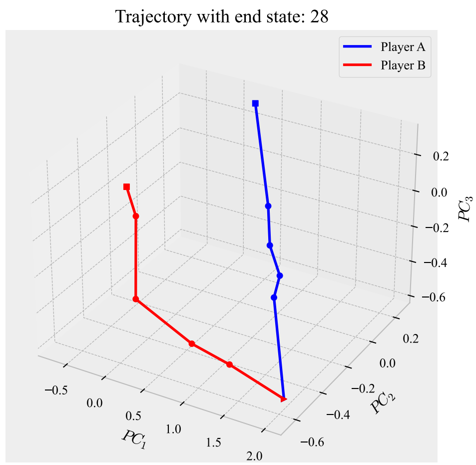
(i)


Trajectory with end state: 46
Player A
Player B
PC_1
PC_2
PC_3

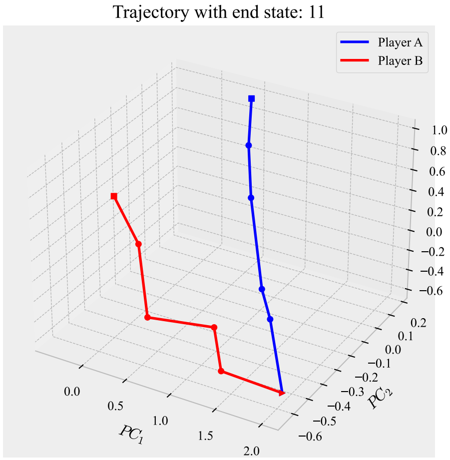
(j)


Trajectory with end state: 44
Player A
Player B
PC_1
PC_2
PC_3

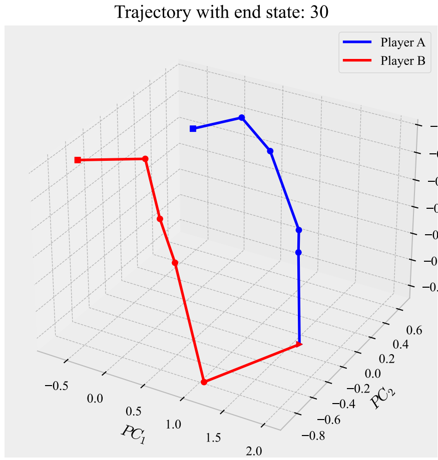
(k)


Trajectory with end state: 28
Player A
Player B
PC_1
PC_2
PC_3

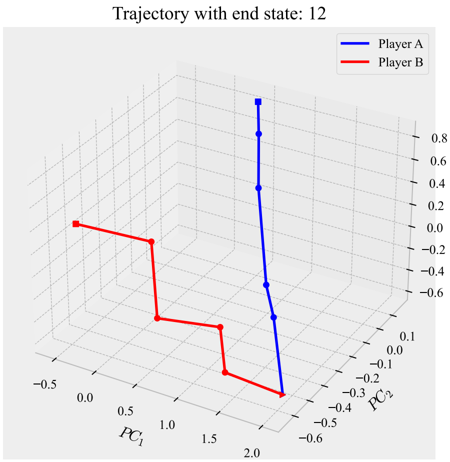
(l)



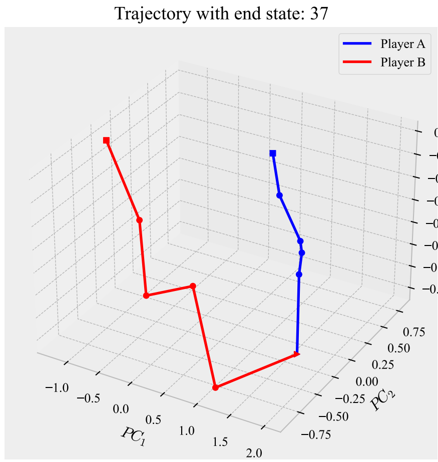
(m)



(n)



(o)



(p)

Figure 31: Presentations of trajectories