Optimizing Air-to-Air Missile Guidance using Reinforcement Learning

M.P. van Hoorn

Technische Universiteit Delft

TU Delft Delft University of Technology

Challenge the future

Optimizing Air-to-Air Missile Guidance using Reinforcement Learning

by

M.P. van Hoorn

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on Tuesday March 26, 2019 at 2:00 PM.

Supervisor:	Dr. ir. S. Hartjes	
Thesis committee:	Dr. A. Gangoli Rao,	TU Delft
	Dr. ir. E. van Kampen,	TU Delft
	Ir. J. Dominicus,	NLR

This thesis is confidential and cannot be made public until March 26, 2021

Acknowledgments

This report encompasses the work I have performed the majority of the past year to obtain my MSc degree in Aerospace Engineering. This marks the end of my time as a student in Delft which has been a tremendous learning experience, not only as an academic but also as a person. I would like to use this opportunity to thank the people who have made this possible.

First of all I would like to thank my supervisor Sander Hartjes for his continued guidance and support throughout this project. Both on the topic of trajectory optimization as well as the process as a whole.

Secondly I would like to thank Jacco Dominicus for his continued input on the project. His knowledge on all aspects of defence operations and missile guidance has helped a great deal in developing the work presented. Furthermore I would like to thank the NLR for hosting me during this project, my colleagues at the NLR for input on the project.

Finally I would like to thank my friends and family for their continued support throughout my time as a student. A special thanks goes out to my parents for making my time as a student possible and supporting me in all my endeavors.

M.P. van Hoorn
Delft, March 2019

ABSTRACT

In this report research is presented regarding air-to-air missile guidance. The performance of an air-to-air missile is largely determined by the guidance law employed, where the guidance law determines the appropriate control action to take to successfully intercept a target. The goal of the research is to develop a novel guidance law which approximates performance and behaviour obtained using trajectory optimization and. Furthermore, contrary to trajectory optimization, it should be feasible as a real-time feedback guidance law. Performance is defined primarily by the ability of the guidance law to hit a target over a large domain (e.g. maximum ranges, hit probabilities). Secondarily it is defined in this research as the time of flight where a faster intercept is beneficial.

Based on a literature study an approach based on Reinforcement Learning (RL) is chosen. Specifically the deep deterministic policy gradient algorithm incorporating hindsight experience replay and learning from demonstrations. This method is based on an agent interacting with an environment during a finite time episode and is rewarded at every discrete timestep using a defined reward function. Based on this generated experience, the specified RL algorithm attempts to improve its behaviour aiming to maximize cumulative reward over an episode. In this research the training environment consists of a simulated air-to-air engagement featuring a missile controlled by the agent and a stationary target, where a single engagement is an episode. The agent learns to hit the stationary target optimizing for minimum time of flight. The resulting agent is implemented in a framework where a predicted intercept point algorithm and a terminal guidance law is added. The resulting combination is able to effectively engage stationary, moving and maneuvering targets. A traditional guidance law is implemented and is used as a baseline for performance evaluation. This guidance law is Proportional Navigation (PN) with lofting.

Comparing maximum range obtained whilst engaging a stationary target, it is found that the RL agent based guidance law attains approximately 90% of performance obtained using optimal control at evaluated altitudes and off-boresight angles. When PN with lofting is used only between 60% and 90% of this maximum range is obtained where especially at lower altitudes performance suffers. The maximum range obtained using the RL agent based guidance law does show outliers towards a lower range. This highlights a drawback of the method where it can not guarantee an appropriate control solution for each possible state. When comparing time of flight when engaging a stationary target, PN with lofting is outperformed over large parts of the evaluated domain by the RL agent based guidance law, whilst times obtained through optimal control are approached (< 5% slower in majority of the domain).

Using a turn and run maneuver where the target initially flies directly at the missile, turns 180 deg away from the missile, and flies away from the missile, the maximum ranges are again found at several altitudes and off-boresight angles. In this scenario the RL agent based guidance achieves between 75% and 90% of maximum range obtained using optimal control. Whereas PN guidance with lofting achieves only between 25% and 75% of maximum range obtained using optimal control.

Finally a target performing random maneuvers is employed to determine hit probabilities from varying ranges and initial target headings. It is established that the RL agent based guidance law has a significantly larger envelope in which a 90% hit probability is achieved if compared to PN guidance with lofting. The no escape zone attained using RL agent based guidance is furthermore significantly larger than that of PN with lofting.

It is concluded that the developed guidance law based on RL is viable as a missile guidance law and shows great performance potential. It is able to effectively approximate performance and behaviour obtained using optimal control over the majority of the evaluated domain whilst being implemented in a real-time feedback manner. Potential for improvement is still present mainly in the training routine. The main recommendations made are to focus on improving training, improving accuracy of physical modelling, and removing the need for predicted intercept points and a terminal guidance law.

Contents

Nomenclature

Abbreviations

AAM	Air to Air Missile
BTT	Bank To Turn
BVR	Beyond Visual Range
DLZ	Dymamic Launch Zones
DoF	Degrees of Freedom
LOS	Line Of Sight
NEZ	No Escape Zone
PIP	Predicted Intercept Point
PN	Proportional Navigation
PP	Pure Pursuit
RL	Reinforcement Learning
SP	Singular Perturbation
STT	Skid To Turn

Latin Symbols

(x, y, z)	Cartesian coordinates of missile in East North Up reference frame	km
$\mathscr{H}$	Hamiltonian	-
$\mathscr{L}$	Lagrange or running cost	-
$\mathcal{N}$	Noise process for RL off-policy exploration	-
$\mathbf{B}$	Boundary conditions for optimal control	-
$\mathbf{C}$	Path constraints for optimal control	-
$\mathbf{f}$	Dynamic constraints for optimal control	-
$\mathbf{L}$	Linkage constraints for optimal control	-
$\mathbf{V_r}$	Relative Velocity Vector	m/s
$\mathbf{x}$	State vector	-
A	Aspect ratio	-
a	Action taken in RL environment	-
$a_{i_b,tgt}$	Target accelerations in targets body reference frame with $i = x, y, z$	m/s^2
C	Lateral aerodynamic force acting on the missile	N
C_L, C_D	Lift and drag coefficients of the missile	-
$C_{D,0}$	Zero-lift drag coefficient of the missile	-
e	Oswald factor	-
g	Goal for RL methodology	–
g_a	Gravitational acceleration	$9.82 m/s^2$
I_{sp}	Specific impulse of missile motor	Ns
J	Cost functional	-
L, D	Lift and drag forces acting on the missile	N
m	Missile mass	kg
N	Navigation Constant	-
Q	Value function	-
q	Dynamic pressure	Pa
r	Reward obtained from RL environment	-
$R_{max,2}$	Maximum no escape range	km

R_{max}	Maximum aerodynamic launch range	km
$R_{min,2}$	Minimum no escape range	km
R_{min}	Minimum launch range	km
R_{tr}	Turn and run range	km
s	State or observation in RL environment	-
S_{ref}	Reference surface area	m^2
T	Missile thrust	N
t	Time	s
t_{go}	Estimated time until target intercept	s
V	Velocity of missile in East North Up reference frame	m/s
Greek Symbols		
α	Mean reverting property of Ornstein-Uhlenbeck process	-
β	Mean of Ornstein-Uhlenbeck process	-
γ	Elevation angle of velocity vector in East North Up reference frame	deg
γ_d	Discount factor	-
μ	Deterministic RL policy	-
Φ	Mayer cost	-
π	Stochastic RL policy	-
$\boldsymbol{\nu},\boldsymbol{\lambda},\boldsymbol{\mu}$	Lagrange multipliers	-
ψ	Azimuth angle of velocity vector in East North Up reference frame	deg
ρ	Atmospheric density	kg/m^3
σ	Volatility of Ornstein-Uhlenbeck process	-
τ	Polyak averaging coefficient	-
θ	Parmaterisation of function estimator	$-$
θ_{el},θ_{azi}	Lines of sight to target with respect to body reference frames	deg
Subscripts		
b	Parameter defined in body reference frame	
f	At terminal time	
i	At initial time $t = 0$	
t	At time t	
tgt	Parameter refers to target	
Superscript		
μ	Denotes usage of the specific policy μ	
π	Denotes usage of the specific policy π	
Q	Denotes usage of the specific value function Q	

1

Introduction

Most countries around the world employ an air force of some sort due to the recognized importance of air superiority during a military campaign [1]. An important factor in achieving air superiority is the Air-to-Air Missile (AAM). Since the introduction of aircraft on the battlefield, weapons have trended from close range machine-guns to advanced beyond visual range AAM [2]. This advancement in weapon technology is reflected in kill ratios achieved by the U.S. Air Force during conflicts. During the Vietnam war the kill ratio was 2:1 where mainly guns and 1st generation AAMs where deployed. During the Desert Storm operation a kill ratio of 11:1 was achieved, where mainly modern AAMs were deployed. One of the enabling technological developments has been in the guidance of such missiles.

1.1. Problem Formulation

One of the most important performance characteristic of an AAM is the effective engagement range. If this range is larger than the effective engagement range of an adversary, this adversary can be engaged without the adversary being able to engage. This is illustrated in figure 1.1. Since the friendly aircraft (blue) has a larger detection and engagement range, the enemy aircraft (red) can be engaged without being able to fire back. Logically this will result in an advantage in an air-to-air engagement. If then sufficient amount of such engagements are won, air superiority can be achieved.

The effective engagement range is the range at which a missile will have a reasonable probability of hitting the target. Furthermore, a so called no-escape zone exist in which its is almost certain the target will be hit no matter what maneuvers the adversary performs. These ranges are dependant on several factors which can be split into two main categories, namely the physical design and capabilities of the missile and the guidance & control of the missile. The research conducted in this thesis focuses on improving the performance of the missile guidance.

The goal of a missile guidance algorithm is to close the distance to a target using information about the current states whilst taking into account changing target states due to its maneuvers. As stated, the future states of the target are unknown thus the guidance algorithm will have to make assumptions to account for this and correct during the evolution of the engagement. Based on these assumptions and changing missile and target states, the algorithm

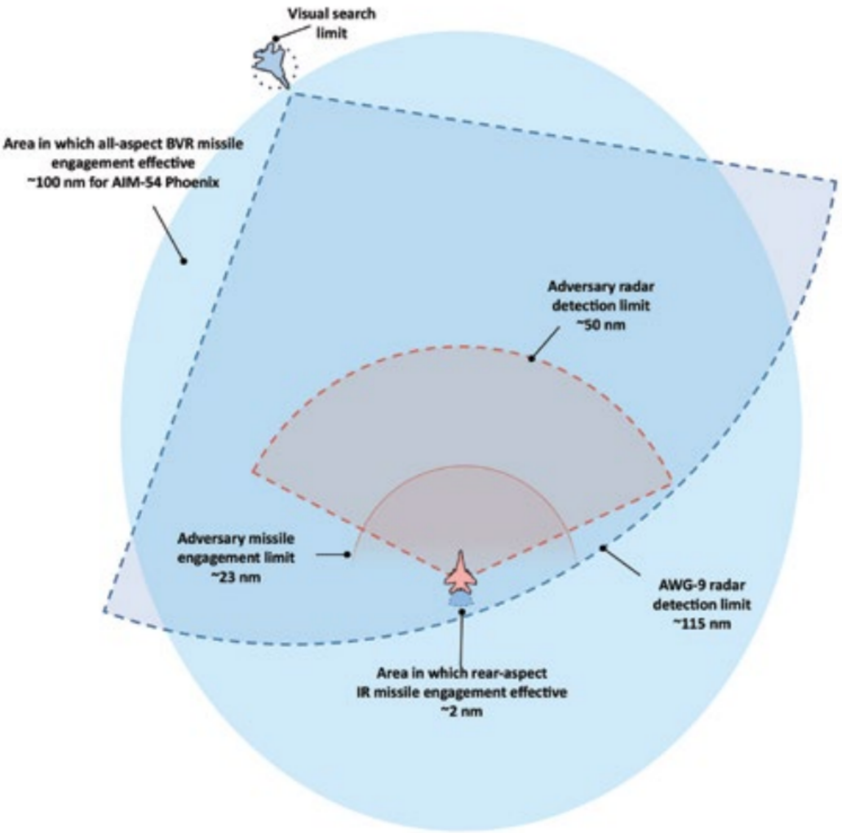

Visual search limit
Area in which all-aspect BVR missile engagement effective ~100 nm for AIM-54 Phoenix
Adversary radar detection limit ~50 nm
Adversary missile engagement limit ~23 nm
AWG-9 radar detection limit ~115 nm
Area in which rear-aspect IR missile engagement effective ~2 nm

Figure 1.1: Visualization of an engagement in which the advantage of a larger effective engagement range is shown [2]

needs to steer the missile on an intercept course.

The performance of the missile guidance and thus the missile is greatly dependant on the guidance law. Generally two major phases are identified in terms of missile guidance, namely the midcourse guidance and terminal guidance. In the midcourse phase, which is the phase between launch and the start of the terminal phase, emphasis lies on closing the majority of the range to the target in an as optimal manner possible. Optimality might be defined as, for example, the shortest time of flight or maximal terminal energy. The terminal phase consist of the last fraction of the flight where target maneuvers have a relatively large impact on the required missile acceleration. Thus it is beneficial if the missile has as much as possible energy left after the midcourse phase.

The guidance problem can be solved using optimal control methods, also often referred to as trajectory optimization. Such methods can compute the trajectories which achieves a defined goal whilst optimizing for a specified performance criterion. In this case this goal could for example be to intercept the target whilst minimizing time of flight is the performance criterion. The drawback of such methods is that they are computationally expensive to solve and require information regarding future target states. Due to these requirements, it is not feasible to implement optimal control as a real-time feedback algorithm. However, if optimal trajectories are compared to traditional methods of missile guidance it is found that especially the range performance of traditional missile guidance laws is inferior to the optimal performance [3] [4] and thus potential performance improvements can be achieved using knowledge from optimal trajectories.

The goal of this study is to develop a missile guidance law for air-to-air missiles which approaches the performance obtained using optimal trajectories whilst limiting computational cost to enable real-time, feedback implementation.

1.2. Past Research

A great amount of research has been performed in the field of missile guidance. At the dawn of the AAM, simple missile guidance laws were employed. Examples of such traditional guidance laws are Pure Pursuit (PP) and Proportional Navigation (PN). Currently many missile guidance laws are still based on PN [5].

The PP guidance law commands an acceleration to direct the missile directly at the target. By doing so, the guidance law steers for the current target position. The PN guidance law commands an acceleration proportional to the Line-Of-Sight (LOS) rate, where the LOS vector is the vector between the missile and target. Mathematically the guidance laws can be given as:

$$a_M = N\mathbf{V_r}\dot{\theta} \tag{1.1}$$

where a_M is the demanded acceleration perpendicular to the LOS, N is the navigation constant, $\mathbf{V_r}$ is the relative velocity vector and $\dot{\theta}$ is the LOS rate vector of the target with respect to the missile.

Due to the lack of information regarding future target behaviour an estimate has to be made regarding an intercept point. This estimate is reflected in these guidance laws by N, PP assumes the intercept point at the current target position and thus $N = 1$. PN assumes the Predicted Intercept Point (PIP) to lie on the current heading of the target at some point in the future meaning $N > 1$. The resulting trajectories for a simple engagement are visualized in figure 1.2. It can be seen that these assumptions regarding future target positions and the PIP influence the guidance greatly.

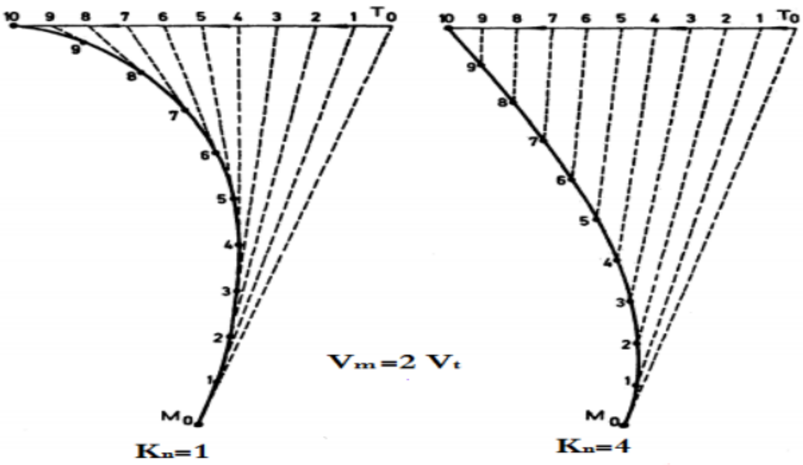


Figure 1.2: Planar engagement showing the working principle of PP and PN guidance where navigation constants used are $N = 1$ and $N = 4$ for the left and right scenario respectively [6]

Most importantly, these missile guidance laws attempt to close the distance to the target only accounting for LOS rate. An example of a potential performance increase left unexploited is the fact that the atmosphere is thinner at higher altitudes and thus less drag is experienced at higher altitude. Thus if optimality is defined as reaching the target as fast as possible, climbing to a higher altitude would be beneficial. Traditional guidance laws do no directly incorporate such behaviour.

A simple, partial solution employed is to loft the missile, meaning the missile gains altitude due to conditions at the launch or alterations to the PN algorithm. Either a elevation angle is given to the missile at launch or a bias is added to the LOS rate observation of the target. An example of this is given in figure 1.3 where the missile is given an elevation angle to artificially loft it.

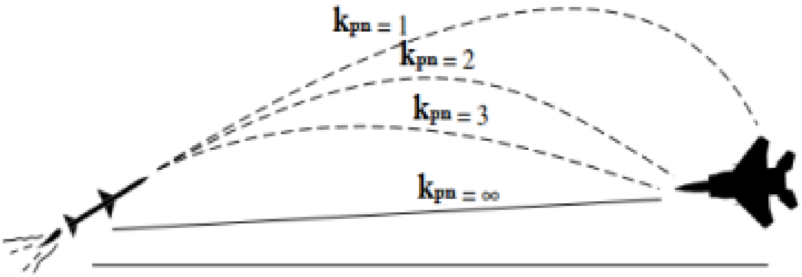


Figure 1.3: Effect of lofting a missile whilst employing different navigations constants [6]

An optimal trajectory could be obtained using optimal control theory. Optimal control theory aims to find a control function that changes the state of a system from an initial condition to a free or fixed final condition, whilst optimizing for a specified criterion, and is subject to system dynamics, and any number of specified constraints such as boundary and path constraints. The obtained state history is the optimal trajectory [7]. However, the obtained control function is an open loop solution and it requires information regarding future target states. Another drawback is the computational time associated with it preventing online updating of the optimal trajectory dependant on the evolution of uncertainties such as target states.

Logically optimal trajectories outperform existing guidance laws if it is provided with information regarding future target states which is shown in research [8]. However, it is also stated that real time implementation is not yet a possibility. This gap in knowledge is the focal point of this dissertation. Several methods have been explored in literature which can be divided into two main categories.

The first category attempts to simplify the physical problem using Singular Perturbation (SP). In the case of missile guidance, the SP method is applied to the problem defined using optimal control leading to an approximate solution of the optimal control problem [4] [9]. These systems usually incorporate high-order dynamic equations where often small parasitic parameters are present increasing the order and stiffness of this system. Stiffness due to the presence of slow and fast phenomena gives rise to time scales. By suppression of a small parameter and thus reducing the order of the system, a singular perturbed system is obtained. The SP approach provides a method to effectively analyze systems by separately solving for each timescale. However the solution obtained using SP methods will be approximate due to the neglecting of dynamics in several timescales. The method still requires discretization in the time domain if the problem cannot be simplified enough. This in term leads to the same issues encountered with the trajectory optimization (requirement of future target states, computational time).

The second category encompasses all methods using machine learning. Several studies have been performed where different machine learning methodologies have been employed. Supervised learning using data generated using optimal control is described in [10] where part of the guidance law is replaced by a neural network. A drawback of this method is the requirement of learning data which can be computationally expensive to generate. Another popular method is the use of an adaptive critic methodology. The adaptive critic methodologies show promise in optimality of results however are limited in the definition of cost function and require many separate neural networks to be trained [11] [12]. Similar research applied to different aerial vehicles show the application of deep Reinforcement Learning (RL) methodologies. These methodologies show promise due to them being independent from the model and ability to learn behaviour optimizing cumulative rewards (e.g. time to complete a trajectory). An example is the application of deep RL to autonomously landing UAVs [13]. Because of the promise of being able to optimize cumulative rewards and its ability to theoretically learn the optimal behaviour independent of the model, it is proposed to pursue a guidance law based on RL.

1.3. Research Questions, Aims, and Objectives

As is established, the main objective of this thesis is to develop a new guidance law using knowledge from identified optimal trajectories to increase the guidance laws performance. This main objective can be split into two sub problems, the first of which is the development of an environment in which an air-to-air engagement can be simulated. Secondly, the implementation and performance evaluation of existing guidance laws and a novel guidance concept. Therefore, the following research question is proposed with several sub-questions:

1. **What improvements to the performance of air-to-air missile guidance systems can be made based on the identification of optimal trajectories?**

   1.1. What method can be used to quantify the performance of a guidance system in an air-to-air engagement?

      1.1.1. What methods can be used to simulate an air-to-air engagement?

      1.1.2. What traditional performance characteristics can be used to describe the performance of guidance laws?

      1.1.3. How can optimal trajectories be used to describe the performance of guidance laws?

   1.2. What changes can be made to missile guidance laws using knowledge from optimal trajectories to improve the performance of the air-to-air missile when compared to existing guidance laws?

      1.2.1. What are the performance characteristics of traditional air-to-air missile guidance laws?

      1.2.2. Which guidance law can be implemented for air-to-air missiles emulating behaviour observed in optimal trajectories?

      1.2.3. How does the performance of this guidance laws compare to traditional air-to-air missile guidance laws?

As is set out in the previous section a methodology based on deep RL is selected. Machine learning has been previously applied to missile guidance however the application of deep RL to missile guidance is not encountered in literature and therefore is a novel concept. Furthermore, using optimal trajectories as a measure of guidance performance is a novel approach. By doing so, it can be estimated what gains can still be made in terms of missile guidance laws by having a theoretical maximum limit on performance. Secondly, the improvement of missile guidance offers a cost effective means of gaining missile performance which in turn will lead to a more favorable air-to-air engagement.

1.4. REPORT STRUCTURE

This report reflects the process used to develop a guidance law which attempts to approach optimal performance. Already covered are the problem formulation, a summary of the literature review previously performed [14], and the research goal and questions. Chapter 2 then aims to establish context regarding AAM and describe the modeling methodology used. Chapter 3 introduces the concept of optimal control and the implementation used in this study. Following this, the methodology employed to develop the proposed guidance law is set out in chapter 4. The resulting guidance law is then used in chapter 5 to evaluate and discuss its performance by comparing it to traditional guidance laws and optimal control. Based on these results conclusions are drawn and recommendations are made in chapter 6.

2

Air-to-Air Missiles

This chapter aims to provide context for the presented research and set out the experimental environment used to simulate an air-to-air engagement. Specifically introducing the AAM, its goals, and how missile guidance fits into the overall concept. First, the goal of the AAM is reiterated and further elaborated upon in section 2.1. Then in section 2.2 the subsystems of the AAM are set out and the role which the guidance law fulfills within this is explained. Section 2.3 introduces a method of representing a missiles performance and the influence of target maneuvers on this performance. Finally, the method used to transfer the real problem to a model which can be used to simulate engagements is introduced in section 2.4.

2.1. Goals

A general introduction to the problem has been provided in section 1.1. This section aims to expand on this. In a military campaign, the battlefield might contain several friendlies and hostiles, both in the air and on the ground. As is apparent from the name, an AAM is launched from an airborne platform and aims to eliminate a hostile airborne platform. The AAM aims to eliminate an adversary, or target, from the battlefield by impacting the target or detonating its warhead close enough to the target to destroy it. Within this framework in which a target can be detected and is targeted, the AAM aim to destroy the designated target.

The AAM of course has limitations in terms of its ability to hit a target. These limitations stem from the design of the missile. Examples could include the ability of the missile to detect the target, the missiles finite amount of energy limiting range, or the missiles limited lateral acceleration leading to the target outmaneuvering it. Thus the goal of an AAM is to hit and destroy a target and the design of the missile determines how effective it is at achieving this.

2.2. AAM Design

An important aspect of missile performance is the airframe itself. Whilst not a deep dive into missile design will be presented here, a theoretical basis is established and several developments and potential limitations present in current missile design are given. It is important to consider these during the development of the novel guidance law.

As with many aerospace vehicles, the missile can be divided into several subsystems. Generally the following subsystems are described in missile design:

- Sensor & Datalink
- State Estimation
- Guidance System
- Control Systems
- Airframe & Propulsion
- Warhead & Fuze
- Missile Power Supply

Each of these subsystems has a function critical to achieve the missiles set out goal. Each subsystem is shortly discussed, and considerations and limitations regarding the guidance law are set out.

2.2.1. SENSORS & DATALINK

The sensors and datalink are discussed together since they have the same function however they fulfill this function differently. They both aim to provide relevant information regarding the current state. This information includes information regarding the missile itself and the target.

Using sensors such as an inertial measurement unit and GPS information regarding the position of the missile can be obtained. To obtain information regarding the position of the target, either information is obtained through a datalink or through onboard sensors. The datalink might be established with the aircraft which launches the missile or another friendly entity. This entity can then relay information it has obtained which might not be available to the missile due to differences in sensor suites.

AAM usually employ a infrared sensor system or radar sensor system to acquire information regarding its target although other methods exist. Infrared systems have the advantage of not alerting the target that its being fired upon since it is a passive system. Infrared sensor can however be fooled by for example flares and are limited in effective range due to other disturbances such as the sun or glare.

Radar systems can either be active or semi-active systems. In both cases the target is illuminated however this is done respectively by the missile or the aircraft. When a radar system is used, the target can be tracked at greater ranges and more information can be extracted such as the range to the target. It does however alert the target that it is being locked by a radar. As is stated in chapter 1.1 mainly increase in engagement range are pursuit in this study, therefore a missile incorporating a radar based seeker system is assumed and range information is available.

2.2.2. STATE ESTIMATION

The state estimation subsystem aims to extract useful information from the raw sensor and datalink data. Effectively translating raw data into states and variances associated with these states using for example Kalman filters. An effective state estimation subsystem provides state information which enables the guidance and control subsystems to work effectively. However, as stated, the state estimation will include variance and thus is not perfect in a practical scenario.

2.2.3. Guidance System

Guidance aims to determine a control command which will lead to reaching a target. This is done based on the observations it is provided of the current environment. Often the method of determining this acceleration is referred to as a guidance law. As is established in section 1.2, many guidance laws exist. An effective guidance law can increase the capabilities of a missile, increasing for example engagement range or terminal energy.

2.2.4. Control System

The control systems aims to translate the required accelerations as determined by the guidance law into control commands. Thus for example deflecting a control surface. An effective control systems reproduces the required acceleration command as accurately as possible. In a realistic scenario the actual acceleration will differ from the required acceleration due to physical effects such as the inertia of the control surface, environmental disturbances, and inaccuracies of the control system.

2.2.5. Airframe & Propulsion

The airframe and propulsion subsystems have several functions. First of all, the airframe houses all the other subsystems and provides structure to the missile. Furthermore, the airframe determines the aerodynamic characteristics of the missile. Due to the missile only having a finite energy at its disposal, the aerodynamic efficiency greatly affects its performance.

Two main categories of airframes exist, namely the Skid-To-Turn (STT) and Bank-To-Turn (BTT) missiles. As is suggested in the name, the STT missile induces an angle of attack in the direction of turning without adjusting roll whilst the BTT missile would roll first and then pitch itself. The BTT can achieve higher aerodynamic efficiency due to requiring high lifting capability in only one direction. However a delay in maneuvering is introduced due to the requirement of rolling. This distinction in maneuvering is however mostly a control issue and will not significantly change guidance law design in the midcourse phase.

The propulsive system of the missile provides energy to the missile. Traditionally solid rocket motors are used due to their simplicity however no throttle control is possible with these systems. Therefore the velocity will increase until either the propellant runs out or the experienced drag exceeds thrust.

Being able to control thrust can lead to a more efficient use of propellant leading to increased range. A practical example of this is the Meteor missile currently in development which uses a variable flow ramjet propulsion system and claims the largest No Escape Zone (NEZ) of any currently existing AAM [15]. In figure 2.1 the velocity profiles of a conventional AAM and the Meteor are quantitatively plotted. As can be seen, the Meteor is able to sustain a higher velocity which subsequently translates to higher closing velocities and ranges.

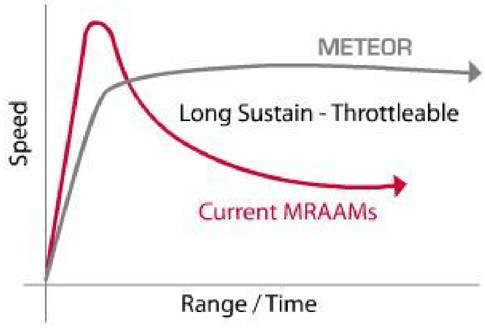


Figure 2.1: Comparison of velocity profiles obtained for medium range AAMs and throttle-able missile such as the Meteor [16]

Throttle control would require the guidance law to determine a required longitudinal acceleration or throttle setting. Furthermore, air-breathing engines could introduce constraint such as angle-of-attack constraints or a maximum velocity.

2.3. AAM Performance

The simplest indicator of AAM performance is the success or failure criteria, where the launch is successful if the target is hit. A secondary performance measures is the time of flight. A lower time of flight would mean the target has less time to take action which could mean engaging a friendly or potentially outrunning the missile. These are however dependant on the initial states of the system and the target maneuvers thus an infinite amount of scenarios exist. To encapsulate many of these scenarios the Dynamic Launch Zones (DLZ) are defined.

2.3.1. Dynamic Launch Zones

The DLZ can be defined for a specific missile and shows the dependence of successful termination on the launch states of the missile, the initial states of the target and the targets maneuvers [16]. A simplified scenario is presented in figure 2.2 where R_{max} or otherwise known as R_{aero} is the maximum aerodynamic range of the missile, $R_{max,2}$ and $R_{min,2}$ are the maximum an minimum range respectively at which theoretically no or little change of escape is possible for the target, and R_{min} is the minimum range at which launch is possible. The maximum range stems from the finite amount of energy available to the missile resulting in the missile not being able to reach the target. Between $R_{max,2}$ and R_{max} the missile has a reasonable chance of hitting the target dependant on the targets actions and its effect on the energy expenditure of the missile. The same holds for the zone between $R_{min,2}$ and R_{min} however here a hit is not guaranteed due to the missile inability to effectively maneuver shortly after launch. Below R_{min} the missile poses a threat to the shooter, which is why the missile will not be launched in this zone. The NEZ is the zone in which the missile can effectively close the distance to the target and outmaneuver the target in the terminal phase. In this zone, the missile has a near guaranteed chance to hit.

As is stated, the DLZ depend on the initial states of both shooter and target, and the trajectory of the target. For example, the altitude of the engagement has a large influence on

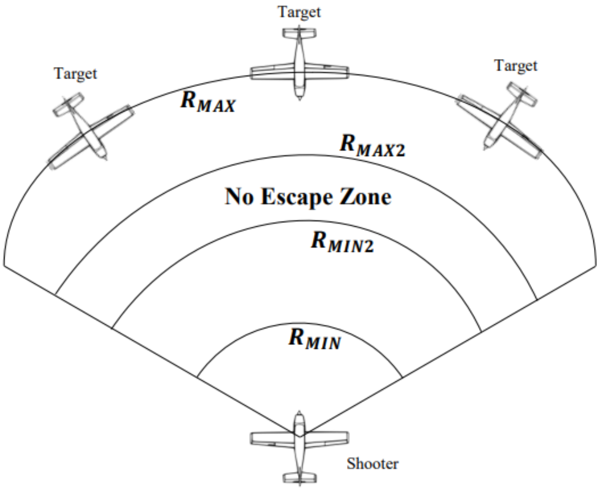


Figure 2.2: Visualization of the DLZ [16]

the DLZ due to missile being able to travel more efficiently in a thinner atmosphere. The DLZ are therefore dependant on a large amount of states and can only be effectively presented using several assumptions and limited scenarios.

Regarding the trajectory of the target, it is logical that the targets direction of travel has an influence on the DLZ. A common addition to the scenario presented in figure 2.2 is the R_{tr} which is the turn-and-run range. For this range it is assumed that the target initially travels in the direction of the target and once the missile is launched it turns with a specific amount of g depending on altitude. Once it is heading away from the missile, the target stops turning and accelerates. Thus at a range lower than R_{tr} the target cannot simply outrun the missile.

2.3.2. TARGET INFORMATION

The DLZ presented in the previous section attempt to generalize the performance of the missile. However, in a realistic scenario an adversary has an infinite amount of possible actions it can take. As is stated, a large unknown is present in the target maneuvers and thus its future states. Therefore the missile is operating based on assumption and is steering to a PIP. The point to which is steered will vary when the targets state evolves and on the guidance method employed.

As an example, assume a target which is flying towards the missile and a simple PIP algorithm which uses the current closing velocity and range to extrapolate the targets position linearly. The PIP will lie somewhere between the missile and the target at this point in time. The target now performs a 180 deg turn in the horizontal plane and heads away from the missile. The PIP will travel along a curve in the horizontal plane until it lies beyond the target. The missile will therefore maneuver in the horizontal plane and spend energy on this maneuver.

This example emphasizes the importance of how this uncertainty is handled. If this is done effectively it will contribute to enlarging the NEZ due to the missile wasting as little energy as possible steering to an incorrect terminal state.

2.4. MODELLING OF AN AIR-TO-AIR ENGAGEMENT

To develop and quantify performance of guidance laws, a simulation environment is required since physical testing is logically not feasible for this research. Modelling of the physical behaviour of both missile and target is in this case the modelling of the state evaluation based on the forces and moments acting on the vehicle. For this, equations of motion are used in combination with a model providing aerodynamic, thrust, and gravitational forces acting on the vehicle.

2.4.1. MODELLING ASSUMPTIONS

The physical problem is simplified in several ways. The subsystems of the missile are greatly simplified since the focus lies on the guidance law. By doing so the performance of guidance laws are compared and not its compatibility with specific implementations of other subsystems.

The missile is modelled as a 3-Degrees of Freedom (DoF) point mass where the aerodynamic angles of attack are neglected. Due to neglecting angles of attack the states of the missile are reduced leading to a more efficient implementation. The state vector of the missile will then consist of three states defining position, three states defining velocity, and one state defining mass or fuel mass left. Neglecting states does mean the model is less accurate, in [17] an in depth comparison between 3-DoF and 6-DoF models is set out where it is concluded that an significant difference in result is present between both. However, it is argued that guidance laws are compared within the same simulation environment thus the absolute performance might be inaccurate however a relative comparison between guidance laws should be valid.

The curvature and rotation of the Earth are neglected since its effect is limited and it is common to exclude these effects in missile trajectory modelling [18] [19]. Since the range is relatively small and the Coriolis acceleration being small compared to the aerodynamic and propulsive forces, this is a valid assumption. Gravity will be assumed constant due to the relatively low altitudes encountered in air-to-air combat leading to low variations in gravitational acceleration.

To determine atmospheric conditions it is assumed that the International Standard Atmosphere is valid and no wind field or disturbances are present. Since the performance will be quantified by comparing guidance laws, random disturbances would introduce variance in the results.

2.4.2. REFERENCE FRAMES

The origin of the absolute coordinate system used for modelling will lie at the surface of the Earth. This coordinate system is defined as East North Up for the (x, y, z) directions. It should be noted that the East and North directions are arbitrary since they have no practical effect on the model and are just defined for completeness. The body reference frame is then defined as (x_b, y_b, z_b) where the x_b-axis is along the missile velocity vector or body axis of the vehicle since angles of attack are neglected, the y_b-axis is pointing left of the vehicle, and the z_b-axis then completes the right handed system.

2.4.3. EQUATIONS OF MOTION

The missile state can then be represented using the state vector $\mathbf{x} = (x, y, z, V, \gamma, \psi, m)$ where x, y, z are the Cartesian position coordinates of the missile, V, γ, ψ are the magnitude and direction of the velocity respectively, and m is the mass of the missile. Using the assumptions given, the state evolution of the missile can be described as

$$\begin{aligned}
\dot{x} &= V\cos\gamma\cos\psi \\
\dot{y} &= V\cos\gamma\sin\psi \\
\dot{z} &= V\sin\gamma \\
\dot{V} &= \frac{T-D}{m} - g_a\sin\gamma \\
\dot{\gamma} &= \frac{L}{mV} - \frac{g\cos\gamma}{V} \\
\dot{\psi} &= \frac{C}{mV\cos\gamma} \\
\dot{m} &= -\frac{I_{sp}}{Tg}
\end{aligned} \tag{2.1}$$

where x, y, z are Cartesian position coordinates, V is velocity, γ is elevation angle with respect to the horizon or $x-y$-plane, ψ is the heading angle or the angle with respect to the $x-z$-plane, T is thrust, D is drag, g_a is the gravitational acceleration, m is mass of the missile, L and C are the lift and side forces acting on the vehicle respectively, and follow from the commanded lateral accelerations multiplied by the mass of the missile.

2.4.4. MODELLING OF FORCES

Three types of forces are acting on the missile and have to be modeled. These are the aerodynamic, propulsive and gravitational forces. These models are purposefully kept simple yet incorporate major effects which are present in a real environment and dictate optimal behaviour. An example of such an effect is the lessening of the atmospheric density with altitude.

As is stated before the gravity will be assumed constant and the calculation is incorporated in the equations of motion.

Secondly the aerodynamic forces have to modelled. To do so, first the ISA is used to calculate the density of the atmosphere ρ at the altitude of the missile. Using the atmospheric density and velocity of the missile the dynamic pressure q can be calculated as

$$q = \frac{1}{2}\rho V^2 \tag{2.2}$$

The total lift coefficient is then determined as

$$C_L = \frac{\sqrt{L^2 + C^2}}{qS_{ref}} \tag{2.3}$$

where S_{ref} is the reference area of the missile. L and C follow directly from the current mass of the missile multiplied by the commanded accelerations in the body frame-of-reference z and y directions respectively. These control commands are constrained to combine for a maximum of 40 g. The drag coefficient is then determined as

$$C_D = C_{D,0} + \frac{C_L^2}{\pi Ae} \tag{2.4}$$

where $C_{D,0}$ is the zero-lift drag coefficient, A is the aspect ratio of the missile, and e is the Oswald factor. This drag coefficient can then subsequently be used to calculate the drag as

$$D = C_D qS \tag{2.5}$$

The thrust is assumed constant and will be either a positive number or zero if the fuel mass is equal to zero. The missile is assumed to have a constant specific impulse and a specified total impulse available. Using these assumptions, no further modelling is required. The thrust is then found as

$$T = \frac{I_{sp}}{\dot{m}g} \tag{2.6}$$

Parameters used for the missile are as found in table 2.1. The parameters chosen are arbitrary since they are the same for each guidance method and the methodology is model-free meaning it can be employed on any model. They are however chosen to obtain range performance approximately equal to that achieved by existing missiles.

Table 2.1: Missile model parameters

Parameter	**Value**	**Unit**
S_{ref}	0.2	m^2
A	2	-
e	0.6	-
$C_{D,0}$	0.05	-
T_{max}	35	kN
J	183	kN/s
m_{empty}	100	kg
m_{fuel}	60	kg

2.4.5. Target States

The target can be modelled using the same states as the missile only disregarding the mass state. Thus the target state is represented by the vector $\mathbf{x_{tgt}} = (x_{tgt}, y_{tgt}, z_{tgt}, V_{tgt}, \gamma_{tgt}, \psi_{tgt})$. Due to the target dynamics not being a main interest of this study, its modelling is simplified. Its lateral acceleration is limited based on altitude where at sea-level it can sustain a 9 G maneuver which is reduced linearly based on altitude to 2 G at 11 km altitude. The longitudinal acceleration is limited to 5 m/s in both directions and is purely a control variable, the effect of drag is not considered. Furthermore, gravity is neglected for the target. Thus the equations of motion then become

$$
\begin{aligned}
\dot{x}_{tgt} &= V_{tgt}\,\cos\gamma_{tgt}\,\cos\phi_{tgt}\\
\dot{y}_{tgt} &= V_{tgt}\,\cos\gamma_{tgt}\,\sin\phi_{tgt}\\
\dot{z}_{tgt} &= V_{tgt}\sin\gamma_{tgt}\\
\dot{V}_{tgt} &= a_{x,tgt}\\
\dot{\gamma}_{tgt} &= \frac{a_{z,tgt}}{V_{tgt}}\\
\dot{\psi}_{tgt} &= \frac{a_{y,tgt}}{V_{tgt}\,\cos\gamma_{tgt}}
\end{aligned}
\tag{2.7}
$$

where $a_{x_b,tgt}, a_{y_b,tgt}, a_{z_b,tgt}$ are the accelerations in the targets body x, y, z-directions. As can be seen these equations are simplified, however since the targets dynamics are not of main interest a pragmatic and simplified model is chosen.

2.4.6. Target Behaviour

Three modes of target behaviour are used in this study, namely:

- Stationary Target: Target is stationary and thus velocity and accelerations are zero.
- Turn and Run: The target initially flies directly at the missile, at $t = 0$ the target turns 180 deg and flies away from the missile.
- Random Maneuvers: The target performs a chain of randomly selected straights and turns with randomly determined accelerations and durations.

The latter two methods are explained further below.

Turn and Run Maneuver

The turn and run scenario is initialized with the target flying directly at the missiles initial position in the horizontal plane, thus the horizontal line-of-sight the target has with respect to the missile is equal to zero. The target then performs a 180 deg turn and after accelerates up to its maximum speed with a specified acceleration. The turn is performed with an acceleration linearly dependant on altitude with a maximum of 9 G at sea-level down to 2 G at 11 km altitude. The maximum longitudinal acceleration is set to 5 m/s and the velocity is limited to a maximum of 250 m/s.

Random Maneuvers

The random behaviour is a chain of turn phases and longitudinal acceleration phases. Each phase has a time duration associated with it determined by a truncated normal distribution with mean 20 s, standard deviation 10 s, and upper and lower bounds [10, 40] s. After this time, the next phase is initialized. At the start of each phase, the type of maneuver is determined by a random 50/50 choice between longitudinal acceleration or turning.

If the maneuver is determined to be a turn, an acceleration and roll angle is randomly determined. The acceleration is sampled from an uniform distribution with bounds

$[g_{max}/4, g_{max}]$ where g_{max} is determined based on altitude in the same manner as is done for the turn-and-run maneuver. The roll angle is sampled from a normal distribution with mean 90 deg and standard deviation 15 deg to mainly have horizontal maneuvers, where a random 50/50 choice determines the sign of the roll angle. Using the roll angle and the absolute acceleration both $\dot{\gamma}_{tgt}$ and $\dot{\psi}_{tgt}$ can be determined. The target velocity derivative $\dot{V}_{tgt}$ is then set to zero. Based on these numbers the target state evolution vector $\dot{s}_{tgt}$ can be determined for each state during the turning phase.

If the maneuver is determined to be a straight, a longitudinal acceleration is sampled from a uniform distribution with bounds $[-5,5]$ m/s to determine $\dot{V}_{tgt}$. Then $\dot{\gamma}_t = \dot{\psi}_t = 0$ and the target state evolution vector $\dot{s}_{tgt}$ can be determined.

The target is limited in altitude and velocity to ensure the target maintains realistic altitudes and velocities. Velocity is simply limited to be within $[150, 300]$ m/s by setting $\dot{V}_{tgt}$ to zero when the bound is violated. The altitude is constraint between $[1, 11]$ km and when the target approaches these limits, $\dot{\mathbf{s}}_{tgt}$ is partly overwritten to steer the target away from these bounds.

Using this process a chain of random target maneuvers is obtained. Resulting trajectories from this process are visualized in figure 2.3.

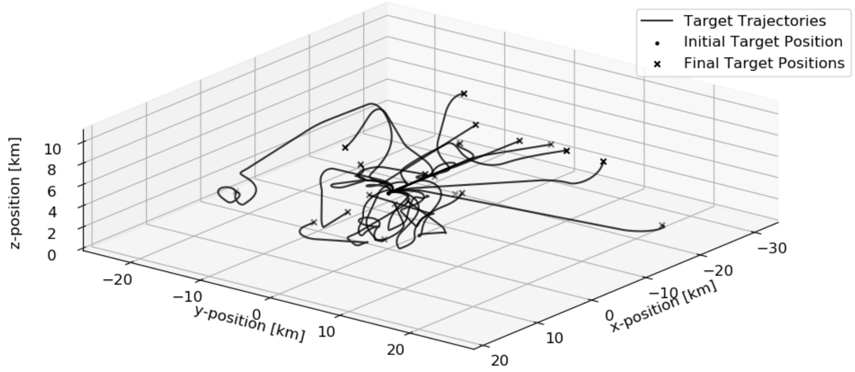


Figure 2.3: Twenty resulting target trajectories using the implemented random maneuver algorithm

2.4.7. Reference Guidance Law

As is stated in [5] many current missile still employ a form of PN. Therefore the PN guidance law is also implemented to be able to compare the newly developed guidance law to it. A navigation constant of 5 is used and a simple lofting algorithm is employed to enhance the performance of the PN guidance. The lofting algorithm adds a bias to the acceleration in the vertical direction if the time-to-go is larger than 10 seconds. This bias is for the first 5 seconds 10 g after which the bias is 1 g to sustain the lofting. This bias is added until the time-to-go is smaller than 10 seconds after which pure PN is used.

3
Optimal Control

Optimal control aims to find a control function that changes the state of a system from an initial condition to a free or fixed final condition, whilst optimizing for a specified criterion, subject to system dynamics, and any number of specified boundary and path constraints. The obtained state history is the optimal trajectory [7]. Using this methodology, the theoretical optimum trajectory for a missile intercepting a target can be found, given that the trajectory of this target is known.

Optimal control serves two purposes in this study. First of all it can be used to quantify performance since it should theoretically provide the optimal solution. Secondly it is used in the methodology as described in chapter 4 to generate demonstration trajectories. In this chapter a theoretical background is provided regarding optimal control theory and the used implementation is set out.

3.1. Problem Formulation

A general problem formulation used in optimal control theory is known as the Bolza problem [7]. It is formulated to minimize a cost functional J:

$$J = \Phi(\mathbf{x}(t_o), t_0, \mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \mathscr{L}(\mathbf{x}(t), \mathbf{u}(t), t)\,dt \tag{3.1}$$

where $\mathscr{L}$ is the Lagrange or running cost and Φ is the Mayer or endpoint cost. Subject to the systems dynamics, the boundary conditions, and the path constraints:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{3.2}$$

$$\mathbf{B}(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) = 0 \tag{3.3}$$

$$\mathbf{C}(\mathbf{x}(t), \mathbf{u}(t), t) \leq 0 \tag{3.4}$$

where $\mathbf{f}$ are the dynamic constraints, $\mathbf{B}$ are the boundary conditions, and $\mathbf{C}$ are the path constraints.

By adjoining the constraints to the cost functional using Lagrange multipliers, the augmented cost functional can be defined as given in equation 3.5

$$\begin{aligned} J_a = \; & \Phi(\mathbf{x}(t_o), t_0, \mathbf{x}(t_f), t_f) - \boldsymbol{\nu}^T \mathbf{B}(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) \\ & + \int_{t_0}^{t_f} \left(\mathscr{L}(\mathbf{x}(t), \mathbf{u}(t), t) - \boldsymbol{\lambda}^T(t)(\dot{\mathbf{x}} - \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t)) - \boldsymbol{\mu}^T(t)\mathbf{C}(\mathbf{x}(t), \mathbf{u}(t), t) \right) dt \end{aligned} \tag{3.5}$$

where $\boldsymbol{\nu}$, $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are Lagrange mutlipliers. By using the Pontryagin's Minimum Principle as given in equation 3.6 with $\mathscr{U}$ the set of permissible controls and $\mathscr{H}$ the Hamiltonian, a set off first-order optimality conditions can be determined [20]. These conditions form the Hamiltonian Boundary-Value Problem and are given in equations 3.7 - 3.14

$$\mathbf{u}^*(t) = arg\{ \min_{\mathbf{u}(t) \in \mathscr{U}} \mathscr{H}(\mathbf{x}(t), \mathbf{u}(t), t)\} \tag{3.6}$$

$$\dot{\mathbf{x}} = \frac{\delta \mathscr{H}}{\delta \boldsymbol{\lambda}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{3.7}$$

$$\dot{\boldsymbol{\lambda}} = -\frac{\delta \mathscr{H}}{\delta \mathbf{x}} \tag{3.8}$$

$$\frac{\delta \mathscr{H}}{\delta \mathbf{u}} = 0 \tag{3.9}$$

$$\boldsymbol{\lambda}^T(t_0) = -\frac{\delta \Phi}{\delta \mathbf{x}(t_0)} + \boldsymbol{\nu}^T \frac{\delta \mathbf{B}}{\delta \mathbf{x}(t_0)} \tag{3.10}$$

$$\boldsymbol{\lambda}^T(t_f) = -\frac{\delta \Phi}{\delta \mathbf{x}(t_f)} + \boldsymbol{\nu}^T \frac{\delta \mathbf{B}}{\delta \mathbf{x}(t_f)} \tag{3.11}$$

$$\mathscr{H}(t_0) = \frac{\delta \Phi}{\delta t_0} - \boldsymbol{\nu}^T \frac{\delta \mathbf{B}}{\delta t_0} \tag{3.12}$$

$$\mathscr{H}(t_f) = \frac{\delta \Phi}{\delta t_f} - \boldsymbol{\nu}^T \frac{\delta \mathbf{B}}{\delta t_f} \tag{3.13}$$

$$\mathbf{B}(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) = 0 \tag{3.14}$$

A solution satisfying the first order optimality conditions does not guarantee a local minimum. To confirm the candidate solution is indeed a local minimum, the second-order sufficiency conditions need to be satisfied.

3.2. IMPLEMENTATION

To obtain optimal control solutions for the model as described in section 2.4 the GPOPS software is used [21]. GPOPS stands for Gauss Pseudospectral Optimization Software. It employs a direct collocation method and as is apparent from the name the software uses a pseudospectral method and LGR collocation points.

Pseudospectral methods are employed in global collocation methods due to their ability to efficiently approximate integrals, differential equations and constraint which are all relevant in optimal control [22]. Pseudospectral methods offer high accuracy with relatively little discretisation points thus reducing computational cost. Methods vary in the trial functions used, quadrature on which they are based and the collocation points at which the approximating function is fitted. For example, Legendre-Gauss (LG), Legendre-Gauss-Lobatto (LGL), or Legendre-Gauss-Radau (LGR) methods can be employed [22]. All three are based on Legendre polynomials as trial functions and collocation points based of a Gauss quadrature. They differ in the collocation points used, LG does not include either boundary points, LGR includes only one boundary point, and LGL includes both boundary points.

Furthermore, GPOPS is able to handle multiphase problems which is critical for the problem at hand. This due to the discontinuity in thrust and thus state derivatives between the phases of boosting and coasting, a discrete phase has to be defined. As this research is not specifically focused on the trajectory optimization, only the implementation off the model as described in 2.4 will be set out.

Two different cost functions are used throughout this study. The first and foremost used is as follows:

$$J = t_f \tag{3.15}$$

where t_f is the terminal time at which thus the boundary conditions are met for the last phase. As can be seen, this is purely a Mayer cost functional. The second cost function used in this report is defined to find the maximum range. This is done by defining the cost function as:

$$J = x_f \tag{3.16}$$

where x_f is the x-position of the missile which is part of the state vector.

As is stated, the problem consists of two phases. The first being the boost phase in which fuel is used to produce thrust. The second is the coast phase in which no thrust is produced and the kinetic energy of the missile slowly diminishes. For both phases, the systems dynamics $\dot{\mathbf{x}}$ are described in sections 2.4.3 and 2.4.4. The phases are linked by linkage constraint defined as:

$$\mathbf{L}_{min} \leq \mathbf{L}\big(\mathbf{x}^{(p_l)}(t_f), t_f^{(p_l)}, \mathbf{x}^{(p_r)}(t_0), t_0^{(p_r)}\big) \leq \mathbf{L}_{max} \tag{3.17}$$

where p_l and p_r are the "left" and "right" phases to be linked. For the missile model this is simply defined as follows:

$$0 \leq \mathbf{x}^{(p_r)}(t_0) - \mathbf{x}^{(p_l)}(t_f) \leq 0 \tag{3.18}$$

thus ensuring that across the phase change all states are equal. It should be mentioned that the optimizer aims to fulfill said constraint with a certain accuracy. Thus realistically zero should be replaced by the set criteria for accuracy (e.g. $\pm 10^{-6}$).

Boundary conditions are enforced at both initial and final time. At t_0 the following boundary conditions are enforced:

$$
\begin{aligned}
x_i \leq & x_0 \leq x_i \\
y_i \leq & y_0 \leq y_i \\
z_i \leq & z_0 \leq z_i \\
V_i \leq & V_0 \leq V_i \\
\gamma_i \leq & \gamma_0 \leq \gamma_i \\
\psi_i \leq & \psi_0 \leq \psi_i \\
m_{empty} \leq & m_f \leq m_{full}
\end{aligned}
\tag{3.19}
$$

where the subscript i stands for initial. In this manner the specified initial state is enforced. As can be seen the initial mass is left free between fully fueled (m_{full}) and empty (m_{empty}). Due to the fact that the optimizer will always choose more fuel mass due to the added energy, this choice is made to alleviate potential numerical difficulties. Then at t_f the following boundary conditions are enforced:

$$
\begin{aligned}
x_{tgt} \leq & x_f \leq x_{tgt} \\
y_{tgt} \leq & y_f \leq y_{tgt} \\
z_{tgt} \leq & z_f \leq z_{tgt} \\
V_{min} \leq & V_f \leq V_{max} \\
\gamma_{min} \leq & \gamma_f \leq \gamma_{max} \\
\psi_{min} \leq & \psi_f \leq \psi_{max} \\
m_{empty} \leq & m_f \leq m_{full}
\end{aligned}
\tag{3.20}
$$

where the subscripts min and max indicate the bounds of the problem, and the subscript t indicates target states. These bounds are as given in table 3.1 and are enforced between t_0 and t_f. As can be seen, most states are left free. The most relevant bounds are the bounds on z, V, and m. The vertical coordinate z should be above sea-level, hence the lower bound. However the upper bound is set in place due to convergence issues which are encountered when increased. This is potentially caused by the ISA being discontinuous leading to an ill posed problem which the solver cannot overcome at these altitudes. For V the upper bound is irrelevant since the missile is not physically able to reach this speed, however the lower bound is relevant. It is set to ensure a minimum velocity during flight which if violated in a realistic scenario would lead to a very low change of intercept due to no or a negative closing velocity.

Table 3.1: Upper and lower bounds used for missile states

Parameter	**Lower bound**	**Upper bound**	**Unit**
x	-1.0×10^6	1.0×10^6	m
y	-1.0×10^6	1.0×10^6	m
z	0	3.0×10^4	m
V	200	2000	m/s
γ	-10π	10π	rad
ψ	-10π	10π	rad
m	100	160	kg

Lastly a path constraint is added to enforce control limits. This path constraint is defined as follows:

$$-40g_a \leq \sqrt{a_{y_b}^2 + a_{z_b}^2} \leq 40g_a \tag{3.21}$$

where a_{y_b} and a_{z_b} are the acceleration commands in the missile body y and z direction respectively.

Lastly, a maximum duration is set for the trajectory of 300 seconds. This is mainly to scope the optimization and RL methodology, however a practical reason is also present due to limited power supply time in missiles.

The optimization is then ran using 32 nodes in each phase, thus in total 64 nodes are used. This number is based on the insignificant change in objective function when the number is increased combined with the acceptable computational time at 32 nodes.

3.2.1. MOVING TARGET

A second implementation is derived from the presented implementation where a target moving in a straight line with a constant velocity is incorporated. This is done using a so called event constraint in which the following is enforced:

$$0 \leq \| [x_{tgt}(t_f), y_{tgt}(t_f), z_{tgt}(t_f)] - [x(t_f), y(t_f), z(t_f)] \| \leq 10 \tag{3.22}$$

where $x_{tgt}, y_{tgt}, z_{tgt}$ are a function of time extrapolating the initial position of the target using a specified velocity vector. This constraint enforces a final distance between missile and target lower than 10 meters. The bounds at t_f then become:

$$\begin{aligned}
x_{min} \leq& x_f \leq x_{max}\\
y_{min} \leq& y_f \leq y_{max}\\
z_{min} \leq& z_f \leq z_{max}\\
V_{min} \leq& V_f \leq V_{max}\\
\gamma_{min} \leq& \gamma_f \leq \gamma_{max}\\
\psi_{min} \leq& \psi_f \leq \psi_{max}\\
m_{empty} \leq& m_f \leq m_{full}
\end{aligned} \tag{3.23}$$

4

Guidance Law using Reinforcement Learning

As is set out in section 2.2.3 the guidance law provides a guidance command based on the current observation of the environment to reach a specified target. This target may be moving and maneuvering and thus the guidance law steers towards an unknown future target state. A method has thus to be developed which determines a control command or action a based on some observation or state s.

In this research it is explicitly chosen to split the determination of an PIP and the determination of the guidance command to reach this PIP. Thus the guidance law is split into two main parts, the determination of the point to which is steered and the determination of the control command to reach this point. This is done for several reasons of which the foremost is the problem of modelling the targets behavior. If a certain behavior is implemented for the target and this described split is not made, the method employed to determine a guidance command would be near-optimal only for the implemented target behavior. A secondary reasons is that this splits the problem into a deterministic problem and a stochastic problem, respectively determining the optimal manner of reaching said PIP and determining this PIP. This in term simplifies implementation and the quantification of performance due to being able to develop and benchmark both implementations separately.

As is stated in section 1.2 a methodology based on deep RL is chosen. This methodology will be employed to determine the guidance command based on the PIP and current state of the missile. The specific methodology chosen is based on the Deep Deterministic Policy Gradient (DDPG) framework. DDPG is chosen due to its compatibility with the specified problem and relative simple implementation making it easy to implement, expand, and scale. The problem at hand features continuous action and state spaces which not all RL methods are compatible with. Furthermore, DDPG takes advantage of the fact that the problem is deterministic leading to more efficient learning [23] [24].

In this chapter the reinforcement learning methods employed to develop an agent which acts as part of the guidance law are set out. First, in section 4.1 the definition of artificial intelligence and machine learning is set out and how RL is a part of this. After this, in section 4.2, the general problem definition associated with RL and several important concepts regarding this report are set out. In section 4.3 the theory behind the used implementation is

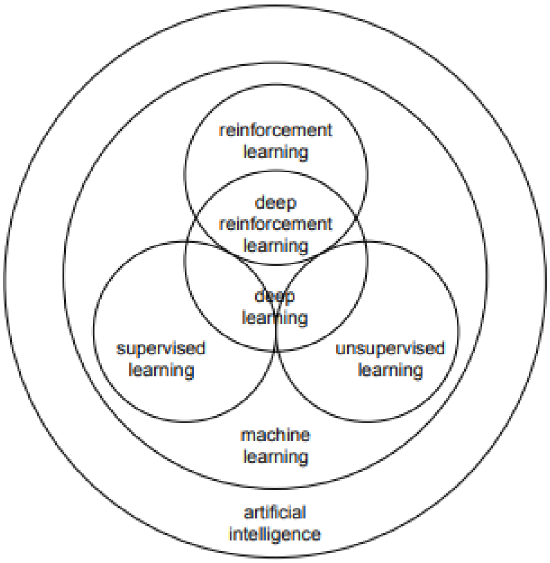


Figure 4.1: Visualization of relevant disciplines encapsulated by artificial intelligence [26]

set out. Finally, section 4.5 covers the actual implementation of the framework combined with the methods of determining the PIP.

4.1. ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Artificial Intelligence (AI) can be defined in many ways. To provide a short introduction, four definitions as given in [25] are given:

1. Definition 1, "acting humanly": creating machines which perform tasks which require intelligence when performed by humans.
2. Definition 2, "thinking humanly": creating machines which can think, learn, and solve problems and thus are intelligent themselves.
3. Definition 3, "thinking rationally": creating machines which can perceive, reason and act
4. Definition 4, "acting rationally": creating machines which act rationally

The last definition, acting rationally, is currently the predominant approach. Machines which show rational behaviour, however do not reason to arrive at this behaviour. Machine Learning (ML) is one of the fields encompassed by AI and can be classified under this definition. An overview of the field of ML is given in figure 4.1

Machine learning is the field of study which attempts to find good predictors based on previous experiences [27]. ML is usually divided into supervised, unsupervised, and reinforcement learning. Supervised and unsupervised learning attempt to extract information from data, where in supervised learning the data is labeled whilst in unsupervised learning the data is not labeled. Examples of the latter are clustering and density estimations. Reinforcement Learning (RL) is usually concerned with sequential decision making and long-term accumulative rewards [26].

4.2. REINFORCEMENT LEARNING

As previously stated, reinforcement learning concerns itself with developing agents which maximize the cumulative reward obtained by taking actions in an environment. This environment is usually a Markov Decision Process (MDP) and can be either a real physical environment or a simulated environment. Behaviour is thus established by rewarding good behaviour and hence reinforcing this behaviour. Several concepts are introduced below which are important throughout this report.

4.2.1. PROBLEM DEFINITION

A standard RL problem consists of an agent interacting with an environment over time. At each time step t the agent determines an action a_t from the action space A based on the observation or state s_t from the state space S. Based on this, the agent receives a scalar reward r_t based on the reward function $r(s_t, a_t)$ and transitions to the next state s_{t+1} according to the state transition probability $P(s_{t+1}|s_t, a_t)$ (thus assuming a stochastic environment). The agent's behavior is determined by the policy $\pi(a_t|s_t)$ which is a mapping from state s_t to action a_t. The problem might be episodic meaning the process continues until the agent reaches a terminal state. The return is defined as

$$R_t = \sum_{i=t}^{T} \gamma_d^{(i-t)} r(s_i, a_i) \tag{4.1}$$

where $\gamma_d \in (0, 1]$ is the discount factor. The discount factor determines if the emphasis lies on short term rewards or long term rewards. The agent aims to maximize this return by choosing the actions to take. Usually RL problems are MDP where the future depends only on the current state and action, specifically not on previous states. Applying a policy to an MDP process defines a Markov chain. The superscript of π is introduced denoting the usage of the policy π over the Markov chain.

Several important concepts are established here, namely the agent, rewards, policies and the MDP. The process described above is visualized in figure 4.2.

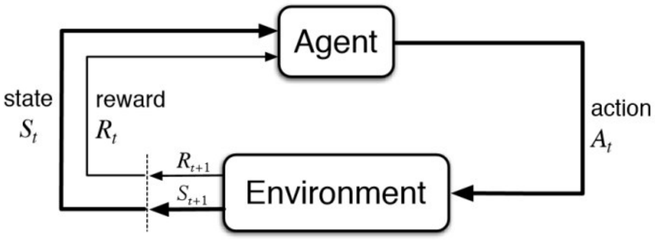


Figure 4.2: Diagram visualizing agent-environment interaction [28]

4.2.2. VALUE FUNCTION

The value function predicts the expected, cumulative, discounted, future reward based on the state or state-action pair. Abstractly it estimates how good the state or state-action pair is. For the latter case, this value function is defined as

$$Q^{\pi}(s) = \mathbb{E}[R_t | s_t, a_t] \tag{4.2}$$

for the expected return for action a_t and state s_t after which the policy π is followed. This equation can be decomposed into the Bellman equation, equation 4.2 then becomes

$$Q^{\pi}(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}\left[r(s_t, a_t) + \gamma \mathbb{E}_{a_{t+1} \sim \pi}[Q^{\pi}(s_{t+1}, a_{t+1})]\right] \tag{4.3}$$

where the environment, E, may also be stochastic. Many RL approaches make use of this relationship [29].

4.2.3. EXPLORATION AND EXPLOITATION

To learn, a RL algorithm needs to generate (new) experiences. Exploiting the existing policy π which might not be optimal and exploring uncertain policies is a dilemma in RL. If no "off-policy" experience is generated, the agent has no method of learning since it is not known what rewards are obtained outside the policies nominal Markov chains. An example of an exploration method is ϵ-greedy where the action determined by the policy is followed with probability $1-\epsilon$, else a random action is chosen within the action space A. This will thus lead to new experience with which the policy might be adapted.

4.2.4. TEMPORAL DIFFERENCE LEARNING

Temporal difference (TD) learning is a reinforcement learning method which learns through bootstrapping. Bootstrapping uses estimates to update a value rather than the exact value. To give an example based on one-step returns, the TD update rules as used in the Q-learning method is given as

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{4.4}$$

where Q is some function estimator used to estimate the value function, α is the learning rate, and $r + \gamma \max_{a'} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$ is called the TD error. As can be seen, the function estimator is updated using the estimated return from the next time step $Q(s_{t+1}, a_{t+1})$ and not an exact value obtained at the end of an episode. Bootstrapping values are common in RL and have benefit such as faster learning, and enabling online and continues learning [26]. It however is not a true gradient decent due to the target depending on an estimate.

4.2.5. POLICY GRADIENT

Policy gradient methods update the policies parameters using the gradient of the expected reward with respect to the policy parameters [28] [30] rather than learning the value function and using it to select the best possible action. Selecting actions is then based on the policy directly without using the value function. This means that the actor is not dependant on a value function to select an action, however the value function is still used to train the policy.

Policy gradient methods are often more stable during training, are more effective in high dimensional action spaces, can handle continuous domains, and the policies may be stochastic or deterministic. A disadvantage is that the learned policy may not the global optimum due to it being a gradient based method. The major benefit being its ability to handle continuous domains.

4.2.6. ACTOR-CRITIC ARCHITECTURE

Actor-critic algorithms consist of two estimators, one of which learns a policy π and the other learns the state-value function Q^{π}. The state-value estimator or critic is updated using for example bootstrapping and the critic is then used to update the parameters of the actor's policy parameters. This architecture accelerates learning and reduces variance [31] [32]. The actor-critic architecture is visualized in figure 4.3. An major advantage of the actor-critic algorithm is that the critic's gradient can be used to train the actor as discussed in section 4.2.5.

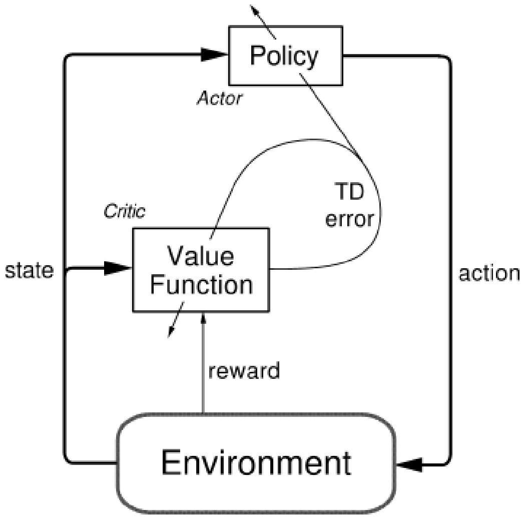


Figure 4.3: Diagram visualizing actor-critic-environment interaction [28]

4.2.7. DEEP REINFORCEMENT LEARNING

Deep learning is the opposite of "shallow" learning and encompasses methods which incorporate one or more hidden layers between the input and output layer. Many deep learning architectures exist however the most well known are based on the Neural Network (NN). A deep neural network maps inputs to outputs using several simple mathematical functions composed in a network structure. In this network so called neurons embody these functions. Each neuron in a layer following the input layer receives a input composed of a weighted sum from the outputs of the previous layer. A mathematical transformation is applied over the neuron such as logistic, tanh, or the rectified linear unit [26].

The structure of the NN allows to compute the error derivatives with respect to the weights connecting the layers. This is called backpropagation and can be used to update the weights of the NN in a very effective manner.

The methods previously discussed rely on function approximators. When a deep NN is used in a RL methodology as a function approximator for any component (for example the policy or value function), it is called a deep RL method.

Due to the NN being a combination of simple mathematical transformations, the amount of operations used when evaluating a NN is small. This is especially relevant in this case since it is to be employed in a feedback manner at a high frequency. A neural network using 3 layers and 512 neurons per layer requires in the order of 1×10^5 FLOPs per evaluation. To put this in a frame of reference, a readily available small computer such as the Raspberry Pi 3 B+ achieves in the order of 1×10^8 FLOPs per second [33] and could theoretically evaluate the mentioned NN at 1×10^3 Hz.

4.3. Deep Deterministic Policy Gradient

Deep Deterministic Policy Gradient is a framework implementing several previously introduced concepts. It is a policy gradient method implementing the actor-critic architecture. The actor and critic are represented by NNs and thus backpropagation can be used to determine the error derivatives with respect to the weights. The critic network is updated through gradients obtained from TD error signals by minimizing the loss given as

$$L(\theta^Q) = \mathbb{E}_{\mu'}\left[\Big(Q(s_t, a_t \mid \theta^Q) - \big(r(s_t, a_t) + \gamma Q(s_{t+1}, \mu(s_{t+1}) \mid \theta^Q)\big)\Big)^2\right] \tag{4.5}$$

where θ^Q represents the parameterisation of Q and μ is used to denote the deterministic policy. To update the actor, first equation 4.3 is rewritten to

$$Q^{\mu}(s_t, a_t) = \mathbb{E}_{r_t, s_{t+1} \sim E}\left[r(s_t, a_t) + \gamma Q^{\mu}(s_{t+1}, \mu(s_{t+1}))\right] \tag{4.6}$$

avoiding the inner expectation due to using the deterministic policy $\mu : S \leftarrow A$. Due to this the expectation only depends on the environment. It is now possible to learn Q^{μ} off-policy using a exploratory policy μ' due to the expectation not explicitly being dependant on the policy [29].

By applying the chain rule to equation 4.6 with respect to the actor parameters θ^{μ}, the policy gradient is obtained as given in equation 4.7.

$$\nabla_{\theta^{\mu}} \mu \approx \mathbb{E}_{\mu'}\left[\nabla_a Q(s,\ a \mid \theta^Q)|_{s=s_t,\ a=\mu(s_t)} \nabla_{\theta^{\mu}} \mu(s \mid \theta^{\mu})_{s=s_t}\right] \tag{4.7}$$

Thus the critic is updated using TD learning and the critic is then used using to update the actor. These updates are performed with an optimizer where in RL the Adam optimizer is popular [34]. However simply implementing this would not result in a stable, converging method where several issues persist. One of them is that the transitions used are generated in a sequential manner due to the episodic nature of the environment. Thus the transitions are not independent and not well distributed. To overcome this a replay buffer is introduced where transitions are stored generated from the environment. Once full, old transitions are discarded and new experience is added. Due to DDPG being an off-policy algorithm, a large replay buffer can be used and a set of uncorrelated transitions can be sampled from it.

Another issue is the instability due to the critic being updated using bootstrapping making it prone to diverge. The concept of target networks is introduced for both the actor and critic network. By having target networks and having the parameterization of both actor and critic "tracking" them rather than directly updating the parameters, the learning can be stabilized. A drawback of this is that learning is slowed down. Mathematically this can be expressed as

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \text{ with } \tau \ll 1 \tag{4.8}$$

where τ is called the polyak averaging coefficient.

The third issue encountered is the potentially different scale of states provided to the algorithm. Vastly different scales can hamper learning and could comprise the algorithms ability to be generalized across many environments. To counteract this batch normalization is used which normalizes each dimension to have unit mean and variance [35].

Lastly, an exploration policy is required to generate new experience. Due to the DDPG algorithm learning off-policy, this is quite simple and can be achieved by adding noise to the actor policy

$$\mu'(s_t) = \mu\left(s_t \mid \theta_t^{\mu}\right) + N \tag{4.9}$$

where N is some form of noise. For physical processes with inertia often an Ornstein-Uhlenbeck process is used due to it generating temporally correlated noise.

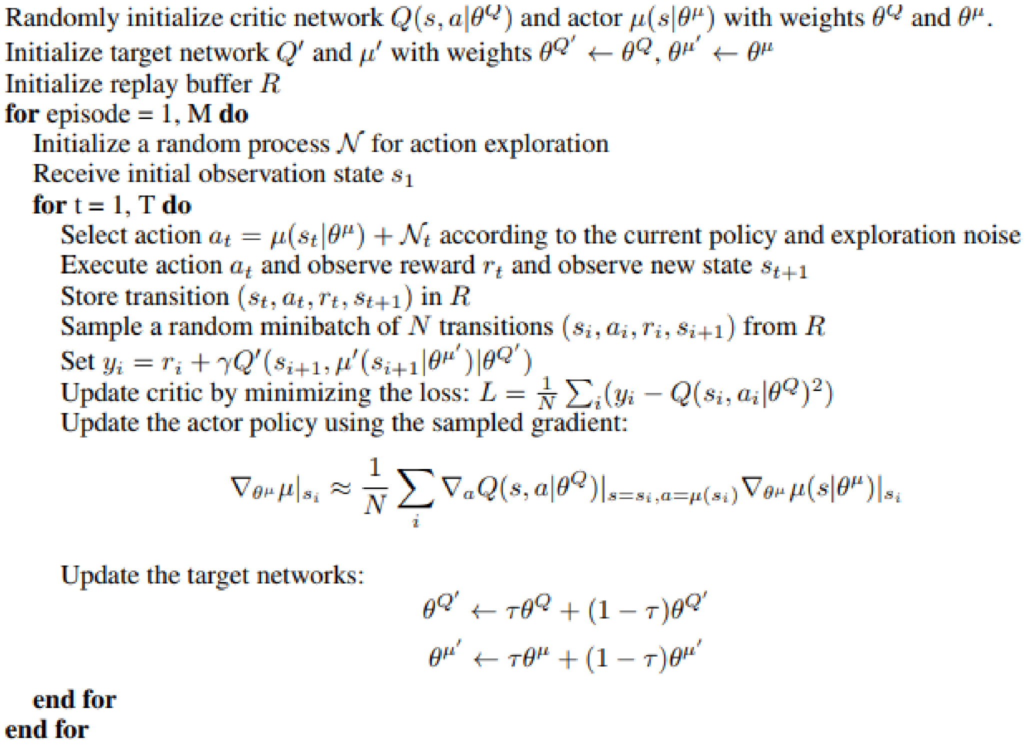
The algorithm is then presented below in figure 1 where the full process of generating experience and updating networks is given. Note that batch normalization is not explicitly included here.

4.3.1. Hindsight Experience Replay

DDPG is, as said, relatively simple in its implementation and thus easily expanded upon. Furthermore, the presented DDPG algorithm will not be able to effectively develop beneficial behaviour in an environment with sparse rewards. This due to the fact that the initial policy will have a very low or zero success rate leading to it experiencing no change in rewards and thus no gradient in the critic network. An example of such an environment is an environment where the goal is to minimize the time to reach a target, a reward of -1 would be given every timestep until the target is reached. Only when success is achieved relatively consistently, more beneficial behaviour can be learned.

To alleviate this problem, Hindsight Experience Replay (HER) is introduced. The core concept behind HER is to reuse generated episodes or trajectories where a different goal is used in the replay [36]. Thus a generated trajectory in an environment featuring sparse rewards can be described by a state sequence $s_1, ..., s_T$ and a goal $g \neq s_1, ..., s_T$ which thus does not contribute to learning to reach goal g. However, by reusing this state sequence and replacing g by $g' = s_T$ and recomputing rewards for each transition, valuable learning experience is gained which is added to the replay buffer. Thus an initial policy which does not successfully achieves its goal can still learn beneficial behaviour from these episodes by changing the goal. The full algorithm is given in figure 2. It has to be noted that the algorithm can be applied to any off-policy RL

Algorithm 1 DDPG algorithm [29]



```
Randomly initialize critic network $Q(s,a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights $\theta^Q$ and $\theta^\mu$.
Initialize target network $Q'$ and $\mu'$ with weights $\theta^{Q'} \leftarrow \theta^Q$, $\theta^{\mu'} \leftarrow \theta^\mu$
Initialize replay buffer $R$
for episode = 1, M do
  Initialize a random process $\mathcal{N}$ for action exploration
  Receive initial observation state $s_1$
  for t = 1, T do
    Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise
    Execute action $a_t$ and observe reward $r_t$ and observe new state $s_{t+1}$
    Store transition $(s_t, a_t, r_t, s_{t+1})$ in $R$
    Sample a random minibatch of $N$ transitions $(s_i, a_i, r_i, s_{i+1})$ from $R$
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i(y_i - Q(s_i, a_i|\theta^Q)^2)$
    Update the actor policy using the sampled gradient:
        $\nabla_{\theta^\mu}\mu|_{s_i} \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i,a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$
    Update the target networks:
        $\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$
        $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$
  end for
end for
```

algorithm therefore the algorithm is more generalized and slightly different nomenclature is used. Research has shown that HER improves learning performance in both environments featuring shaped or sparse rewards [36].

4.3.2. OVERCOMING EXPLORATION WITH DEMONSTRATION

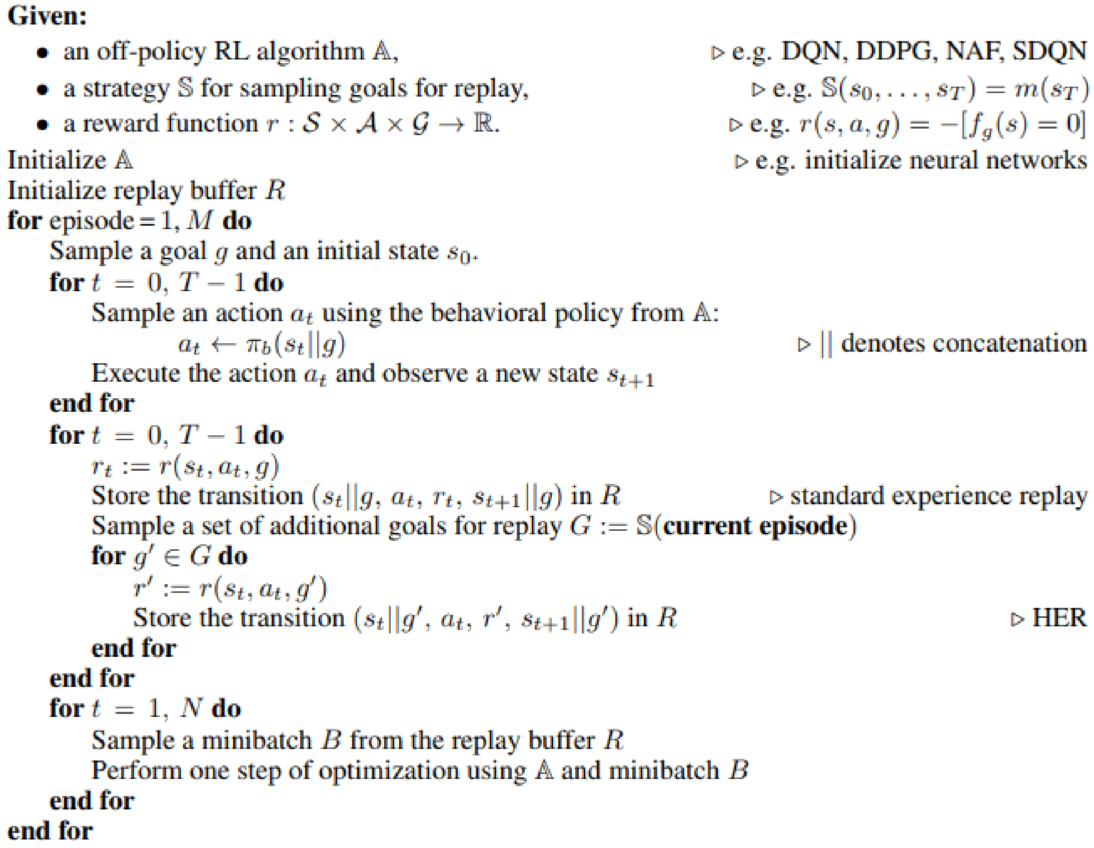
Under the same premise as HER, learning from demonstration is introduced. Like HER, DDPG from Demonstration (DDPGfD) overcomes the problem of exploration in an environment with sparse rewards. Learning from demonstrations does imply that demonstrations are available and in this case optimal trajectories can be generated as demonstrations by using optimal control as introduced in chapter 3.

To enable learning from demonstration a second replay buffer R_D is introduced where the transitions obtained from demonstration are stored. During training an extra set of samples is drawn from this replay buffer and added to each minibatch in both the actor and critic updates. Behavior cloning loss is introduced where the policies action is compared to the demonstrated action in a specific state. It is defined as [37]

$$L_{BC} = \sum_{i=1}^{N_D} ||\ \mu(s_i\,|\,\theta_\mu) - a_i\ ||^2\ \mathbb{1}_{Q(s_i,a_i)>Q(s_i,\mu(s_i))} \tag{4.10}$$

where a so called Q-filter is added determining if the demonstrated action is better than the action determined by the policy. If not, the demonstration is not used. The gradient used to update actor is then [37]

Algorithm 2 Algorithm for off-policy RL methods incorporating HER [36]



```
Given:
  • an off-policy RL algorithm $\mathbb{A}$,                         ▷ e.g. DQN, DDPG, NAF, SDQN
  • a strategy $\mathbb{S}$ for sampling goals for replay,            ▷ e.g. $\mathbb{S}(s_0, \ldots, s_T) = m(s_T)$
  • a reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathbb{R}$.   ▷ e.g. $r(s, a, g) = -[f_g(s) = 0]$
Initialize $\mathbb{A}$                                                ▷ e.g. initialize neural networks
Initialize replay buffer $R$
for episode = 1, $M$ do
    Sample a goal $g$ and an initial state $s_0$.
    for $t = 0, T - 1$ do
        Sample an action $a_t$ using the behavioral policy from $\mathbb{A}$:
                $a_t \leftarrow \pi_b(s_t||g)$                          ▷ || denotes concatenation
        Execute the action $a_t$ and observe a new state $s_{t+1}$
    end for
    for $t = 0, T - 1$ do
        $r_t := r(s_t, a_t, g)$
        Store the transition $(s_t||g, a_t, r_t, s_{t+1}||g)$ in $R$      ▷ standard experience replay
        Sample a set of additional goals for replay $G := \mathbb{S}($current episode$)$
        for $g' \in G$ do
            $r' := r(s_t, a_t, g')$
            Store the transition $(s_t||g', a_t, r', s_{t+1}||g')$ in $R$      ▷ HER
        end for
    end for
    for $t = 1, N$ do
        Sample a minibatch $B$ from the replay buffer $R$
        Perform one step of optimization using $\mathbb{A}$ and minibatch $B$
    end for
end for
```

$$\eta_1 \nabla_{\theta_\mu} \mu - \eta_2 \nabla_{\theta_\mu} L_{BC} \tag{4.11}$$

which is a weighted combination of the regular policy gradient as given in equation 4.7 and the gradient of the policy with respect to the behavior cloning loss. Due to the behavior cloning loss being a loss, one wants to minimize this where one wants maximize the rewards obtained from the policy which explains the subtraction of the gradient of the loss. In [37] no further elaboration on the choice of the weights η_1 and η_2 is given thus the standard values provided will be used. Practically the weighting represents which loss is prioritized where a higher weight represents a higher priority.

4.3.3. EXPLORATION

To generate new experience often noise is added to the policy. As this is a problem with mass inertia, random Gaussian noise would not have a great effect on the resulting trajectory. Therefore Ornstein-Uhlenbeck noise is used which is temporarily correlated [38]. This is achieved by the state of the noise process being correlated to the noise at the previous state of the process. Furthermore it has the mean reverting property meaning that it drifts back to the mean overtime. Mathematically it is expressed as

$$dx_t = \alpha(\beta - x_t)dt + \sigma dW_t \tag{4.12}$$

where $\alpha > 0$, β, and $\sigma > 0$ are parameters and W_t is a Brownian motion process. In literature α and β are usually denoted using different symbols however these have been changed to not conflict with other symbols used in this report. α, β, and σ can be described as the tendency to revert to the mean, the mean, and the volatility respectively.

4.4. TRAINING OF AGENT

Thus the training algorithm has been established, however this training algorithm needs to interact with an environment to perform training and simulation. It should be noted that up to this point no specific environment has been considered whilst setting out the training methodology and thus the methodology is not constraint to the described environment and can be applied to many problems with a similar problem definition.

4.4.1. ENVIRONMENT

As is depicted in figure 4.2 the environment takes an action at every timestep and transitions to a new state and is rewarded with some scalar reward. However, first the environment is reset to obtain an initial state. A random initial state for the missile is used for this initial point which is uniformly distributed over the domain. In this case, the initial state is constraint using a maximum range of 300 km in the horizontal plane, whilst the altitude is constrained between 2 and 11 km. Furthermore the missile is always initialized using a full fuel load. Initial velocities can range from 250 to 500 m/s as would be the case when launched from a fighter. Furthermore, initial off boresight angles are limited to be between ±45 degrees in elevation, whilst the horizontal off boresight angle is left free between ±180 degrees. During training only a stationary target is considered as stated before. For training, this target is always initialized at $x, y = 0$ whilst the altitude of the target can vary between 2 km and 11 km.

Starting from the initial state, the environment integrates over time with a timestep of 0.01 seconds whilst an action is selected every 50 timesteps or 0.50 seconds when far from the target and an action every 5 timesteps or 0.05 seconds when closer than 1 km to the target. This is done to limit computational time of episodes which speeds up training significantly. Furthermore, an episode is limited to either 650 timesteps or 300 seconds depending on which is reached earlier. The agent is provided a different vector as an observation than the states as used in the equations of motion. This is done to facilitate easier learning due to the provided observations being more intuitive (e.g. range to target is more directly correlated to a guidance command than six position coordinates). The provided observation vector is defined as $s = (\theta_{el}, \theta_{azi}, \dot{\theta}_{el}, \dot{\theta}_{azi}, R, z, V, \gamma, \psi, m)$. Where θ_{el} and θ_{azi} are the vertical and horizontal lines of sights with respect to the missiles body reference frame and $\dot{\theta}_{el}$ and $\dot{\theta}_{azi}$ are their respective derivatives. Furthermore R is the range to target.

The agent achieves success when coming within 10 m of the target. However, it is found that the agent fails to achieve sufficient accuracy in the terminal stage due to the relatively large domain (volume in the order of $10 \times 10^{14}\,\text{m}^3$) with respect to the goal (volume in the order of $10 \times 10^3\,\text{m}^3$) combined with the limited estimation power of the neural network. The solution to this problem will be elaborated on further in section 4.5. However to alleviate this problem in training, the success criteria in training is changed where the agent achieved success when coming within 2.5 km of the target. A reward for a low LOS angle as then ensures the agent learns to point itself directly at the target at this boundary. This reward is equal to 1×10^4 multiplied by the final LOS angles, thus $\cos(\theta_{el})$ and $\cos(\theta_{azi})$.

Each timestep the agent receives a penalty of $[-1]$ thus the goal of the agent is to reach the target in the minimal time possible. This objective is chosen for several reasons. First of all as is established, a shorter time of flight has several advantages. The main reasons being less time for the target to take evasive action and it incentivises a high terminal velocity and thus energy. Adding to this, the agent will always strive to reach the target and thus at maximum range it will choose the only solution also being the minimum time solution.

Lastly, constraints are enforced by penalizing the agent if a constraint is violated. For each constraint violated, 1×10^3 is added to the reward. An example of a constraint employed is the minimum altitude of $0\,\mathrm{m}$. The reason why the episode is not simply terminated once a constraint is violated is that it will be seen as beneficial behaviour due to the agent not receiving more penalties.

4.4.2. Training Setup

The full implementation uses DDPGfD combined with HER. Furthermore Ornstein-Uhlenbeck noise is used to explore and generate new experience. The training process then consists of a specified number of epochs. At the end of each epoch the obtained policy is tested and can be stored. During each epoch a specified number of cycles are performed. Each cycle consists of a number of episodes being generated and stored in the replay buffer. After this the networks are trained using batches of data sampled from the replay buffers. These batches of transitions consist of regular experience, HER transitions, and demonstration transitions. The demonstrations are generated using optimal control as presented in chapter 3. Due to the fact that these demonstrations are assumed to be optimal, the Q-filter is not used on demonstration transitions.

In table 4.1 the training parameters used are set out. It should be noted that the training process is fully parallelized, therefore some of the parameters are expressed as per thread.

4.4.3. Convergence

Using the setup provided above, the training can be monitored using results from test episodes which are ran every epoch. In figures 4.4 and 4.5 the median success rate and mean Q-value obtained for each epoch using the described algorithm on the 2-DoF and 3-DoF environment are visualized. In both environments the agent achieves a high success rate early due to the provided demonstrations. With subsequent epochs, behaviour generally improves indicated by the increasing success rate and mean Q-value. Furthermore, learning is relatively stable.

Table 4.1: Parameters used for DDPGfD + HER training process

Description	**Value**	**Comment**
Neural Networks		
Hidden layers	3	
Neurons per layer	512	
Optimizer	Adam	Deep learning optimizer for network weights
Q_{lr}	1E-3	Learning rate of critic
μ_{lr}	1E-4	Learning rate of actor
DDPGfD and HER		
Buffer size	1E6	-
Polyak coefficient	0.8	-
γ	$1-1/T$	Discount factor with $T = 600$ (max. timesteps)
Minibatch size	1024	Transitions per thread
HER ratio	4/1	Ratio of HER transitions to regular transition
Demo batch size	256	Demonstration transitions per thread
η_1	0.001	Behaviour cloning loss weighting
η_2	0.0078	Behaviour cloning loss weighting
Demo episodes	800	Episodes entered in the demonstration buffer
Training		
Threads	128	Parallel processes used for training
Epochs	50	-
Cycles	50	Per epoch
Minibatches	16	Per cycle
Exploration		-
α	0.1	Mean reverting property of Ornstein-Uhlenbeck process
β	0	Mean of Ornstein-Uhlenbeck process
σ	0.01	Volatility of Ornstein-Uhlenbeck process

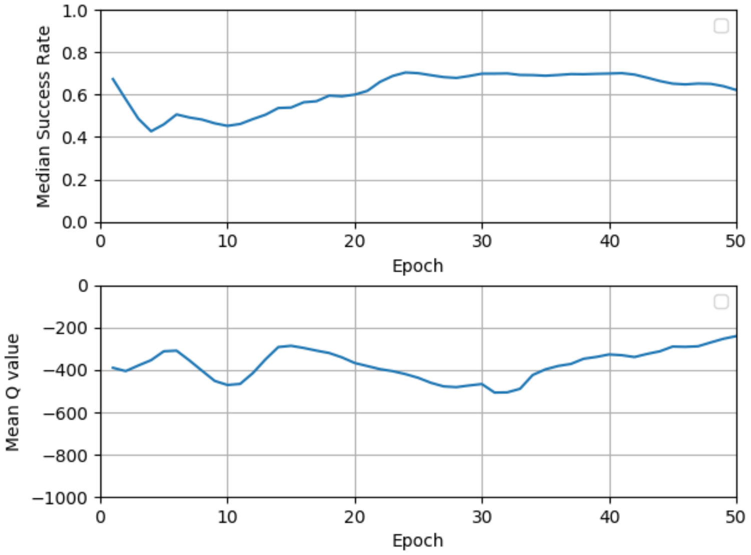


Figure 4.4: Convergence of median success rate and mean Q-value for training of 2-DoF guidance agent

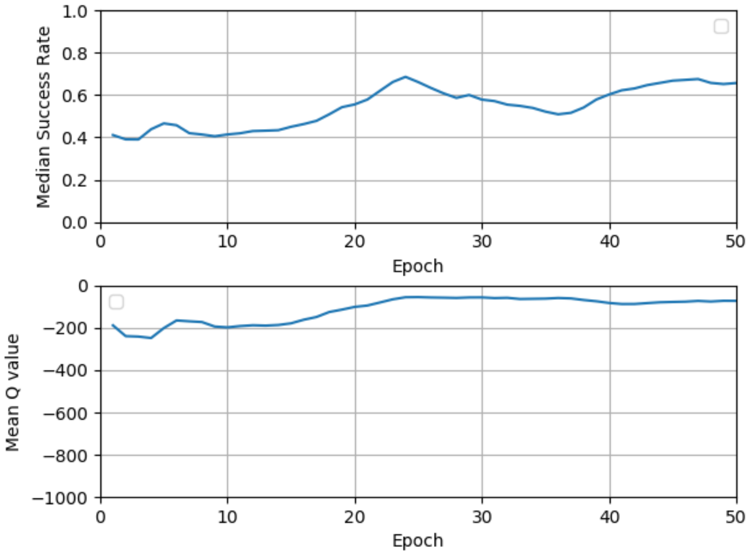


Figure 4.5: Convergence of median success rate and mean Q-value for training of 3-DoF guidance agent

4.5. IMPLEMENTATION OF GUIDANCE LAW

The established agent through previously explained methods is only part of a guidance law. As is stated before, an algorithm is still required to obtain an PIP. Furthermore, in the previous section the problem is raised that the agent does not perform well enough in minimizing miss distance in the terminal stage. To overcome these problems a PIP algorithm is presented below and in the terminal guidance phase PN is applied in favour of the RL agent. The guidance law is presented schematically in figure 4.6.

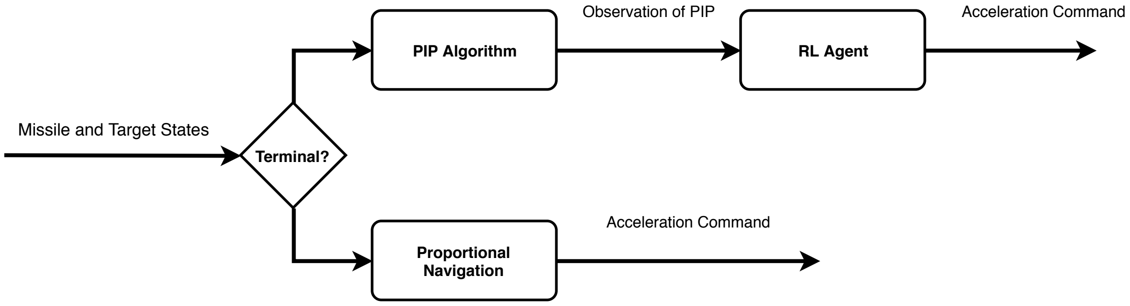


Figure 4.6: Schematic of RL agent based guidance law

The framework presented provides a lot of options due to the possibility of using different PIP algorithms, RL agents, switching criteria, and terminal guidance law in combination with each other.

4.5.1. RL AGENT

The RL agent used in the guidance algorithm is obtained through methods explained in sections 4.3 and 4.4. The policy generated through training with the highest median success rate is used as the RL agent in the guidance algorithm.

4.5.2. TERMINAL GUIDANCE

The decision to switch over to terminal guidance has to be made based on the states or observations of the system. In this case the choice is made to switch over the horizontal and vertical guidance individually. This is done to increase robustness of the guidance law whilst still benefiting from the increased performance obtained from the RL agent in early and midcourse stages of the trajectory.

The guidance in the horizontal plane is switched from the RL agent to PN guidance after the horizontal off boresight angle θ_{azi} is lower than a specified number (in this case 5 deg). This is a practical choice due to the RL agent providing better performance in high off boresight angles however once aligned with the target no major benefit is provided by the RL agent whilst unpredictable behaviour might be shown by it.

The guidance in the vertical plane is switched over to PN at a specified range (in this case 10 km). The navigation constant then used for PN is equal to 5. This point is usually after the horizontal guidance switch-over and thus after this point the missile is fully guided using PN.

Furthermore, a "sanity" check is introduced to the guidance scheme where the command provided by the RL agent is compared with the PN guidance command for the specific state of the system. This check is only performed after the initial 10 seconds of flight due to large differences in commands between the two guidance schemes present during this initial phase. If the difference is larger than 20% of the maximum acceleration between 10 and 20 seconds, or larger then 10% of the maximum acceleration after 20 seconds the switch to PN guidance is made. By doing so the handover to terminal guidance is relatively smooth minimizing energy losses due to a sudden large acceleration. Furthermore it cannot be guaranteed that the RL agent has a correct solution for every state of the system. By adding this sanity check the guidance scheme should be more robust.

4.5.3. PIP ALGORITHM

Two PIP algorithms are used in this study to provide an observation to the RL agent. The PIP algorithm predicts the position at which the missile will reach the target and derives a virtual observation of the target at this position. Thus the RL agent observes the target to be stationary at the PIP. The two PIP algorithms both assume the velocity of the target to be constant where the difference between both lies in the assumed direction of this velocity. The first mode assumes the velocity to remain in the direction it is at the current timestep whilst the second mode assumes the target to fly away from the missile thus assuming a worst case scenario.

For both PIP algorithms first the time-to-go t_{go} is estimated. This is done using the relative velocity V_r and the range to target R where the time-to-go is found using

$$t_{go} = \frac{R}{V_r} \tag{4.13}$$

Based on the t_{go} the position of the target can be extrapolated. For the first PIP-mode, which will be referred to as PIP-mode 1, the actual velocity vector of the target is used to determine the PIP. Whilst for the second PIP-mode, PIP-mode 2, a virtual velocity vector is used to calculate the PIP. This virtual velocity vector assumes ψ_{tgt} as if the target is flying directly away from the missile in the horizontal plane. In this manner the worst case is assumed where the target flies away from the missile. PIP-mode 1 and PIP-mode 2 are visualized in figures 4.7 and 4.8 respectively.

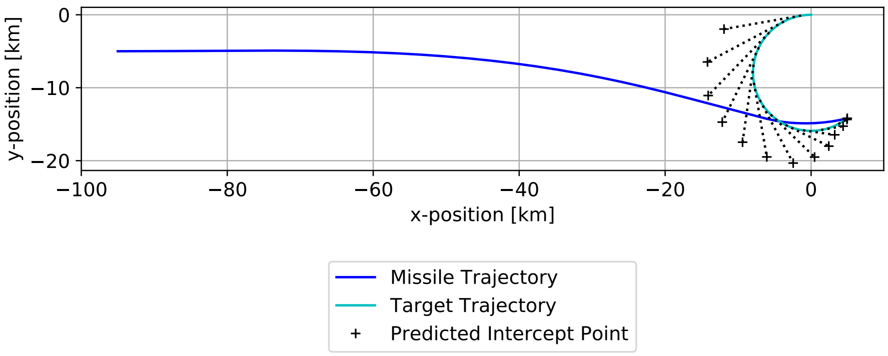


Figure 4.7: Visualization of PIP-mode 1 in the x-y plane

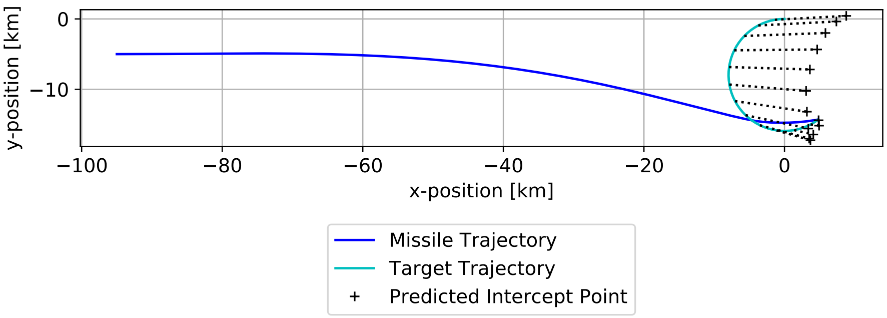


Figure 4.8: Visualization of PIP-mode 2 in the x-y plane

5 Results

The methodology and experimental environment as described in chapter 4 will be used in this chapter to analyze the performance of the resulting guidance law. First the algorithm is applied to a simplified environment to proof the concept and establish an initial performance quantification. After this, the simplifications of the environment are removed and the performance of the establish guidance law is evaluated through individual cases and the performance is generalized using results from a multitude of engagements. To isolate the performance of the RL agent and the PIP algorithm, first an analysis is performed with stationary targets after which target maneuvers are introduced to study the full guidance law using a realistic environment.

5.1. Two-Degrees of Freedom

As is stated first a proof of concept is established using a simplified environment. The engagement is constrained to a planar engagement by setting $y = \dot{y} = 0$ and $\psi = \dot{\psi} = 0$. This leaves 5 states and thus $\mathbf{x} = (x, z, V, \gamma, m)$. Furthermore, the concept which has to be proven is the RL agent thus no moving target will be considered and subsequently no PIP algorithm is required. An agent is trained for this environment using the methodology as described in sections 4.3 - 4.4.

5.1.1. Case Study

Using the simplified environment a case study is performed. The case study is initialized using the states and target states as given in tables 5.1 and 5.2.

The resulting trajectory is then given in figure 5.1. It can be seen that the RL agent based guidance outperforms PN navigation guidance by a significant margin in terms of time of flight. Furthermore, the RL agent based guidance approximates performance obtained using optimal control in terms of time of flight. This is a promising results since the specific case simulated here is most likely not encountered in training, which insinuates that the RL agent is able to generalize behaviour and approximate performance obtained from optimal control.

During the initial 10 seconds of flight the optimal trajectory and the trajectory obtained using RL agent based guidance closely match as can be seen from figure 5.2. Both choose

Table 5.1: Initial missile states

State	Value	Unit
x	-50	km
y	N/A	km
z	5	km
V	300	m/s
γ	0	deg
ψ	N/A	deg
m	160	kg

Table 5.2: Initial target states

State	Value	Unit
x_{tgt}	0	km
y_{tgt}	N/A	km
z_{tgt}	5	km
V_{tgt}	0	m/s
γ_{tgt}	0	deg
ψ_{tgt}	N/A	deg

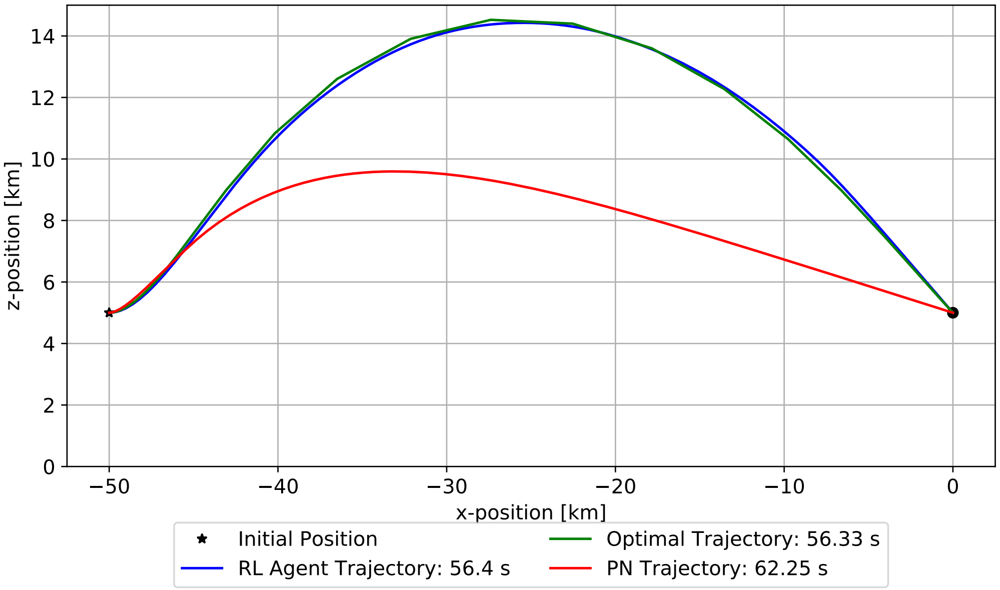


Figure 5.1: Resulting trajectories plotted for missile intercepting stationary target using PN guidance with lofting, RL agent based guidance, and the optimal trajectory in a 2-DoF environment

choose a significantly higher trajectory then PN with lofting. This higher trajectory is achieved by sustaining a higher elevation angle longer as can be seen in figure 5.3 resulting in a higher sustained velocity. This in term leads to a shorter time of flight even though both RL agent based guidance and optimal control travel a longer distance due to the higher trajectory.

Looking at figure 5.4 it can be seen that the control commands commanded by the RL agent closely match the control commands obtained using optimal control. The main difference being that the RL agent provides slightly more erratic commands. It can also be seen that the lofting algorithm seems effective at first however is quickly overridden by the LOS-rate generated by the lofting leading to a nose down command. This in term prevents the missile using PN to loft as effectively as the RL agent based guidance and optimal trajectory.

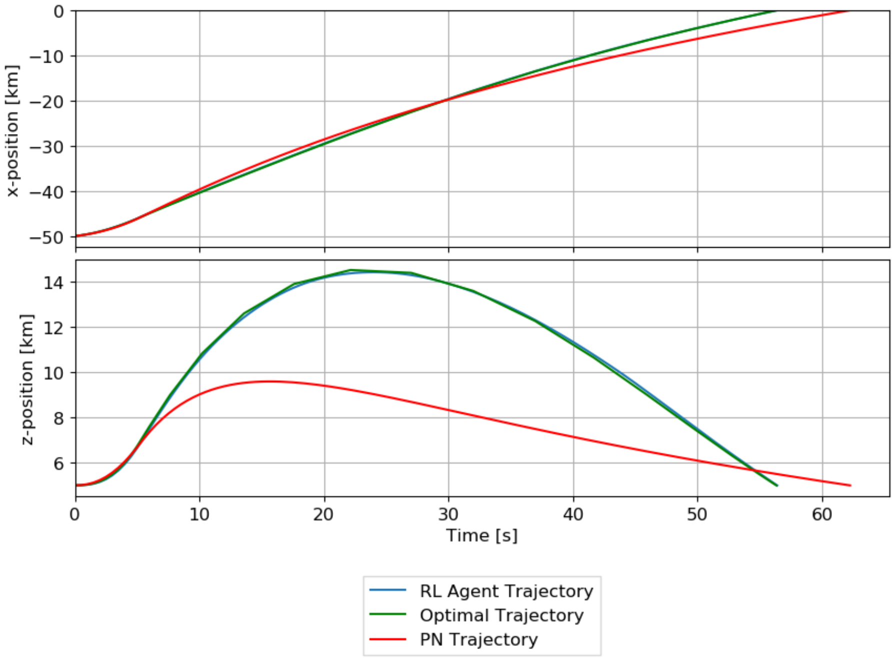


Figure 5.2: x and z-position plotted versus time for missile trajectories obtained using PN guidance with lofting, RL agent based guidance, and the optimal trajectory

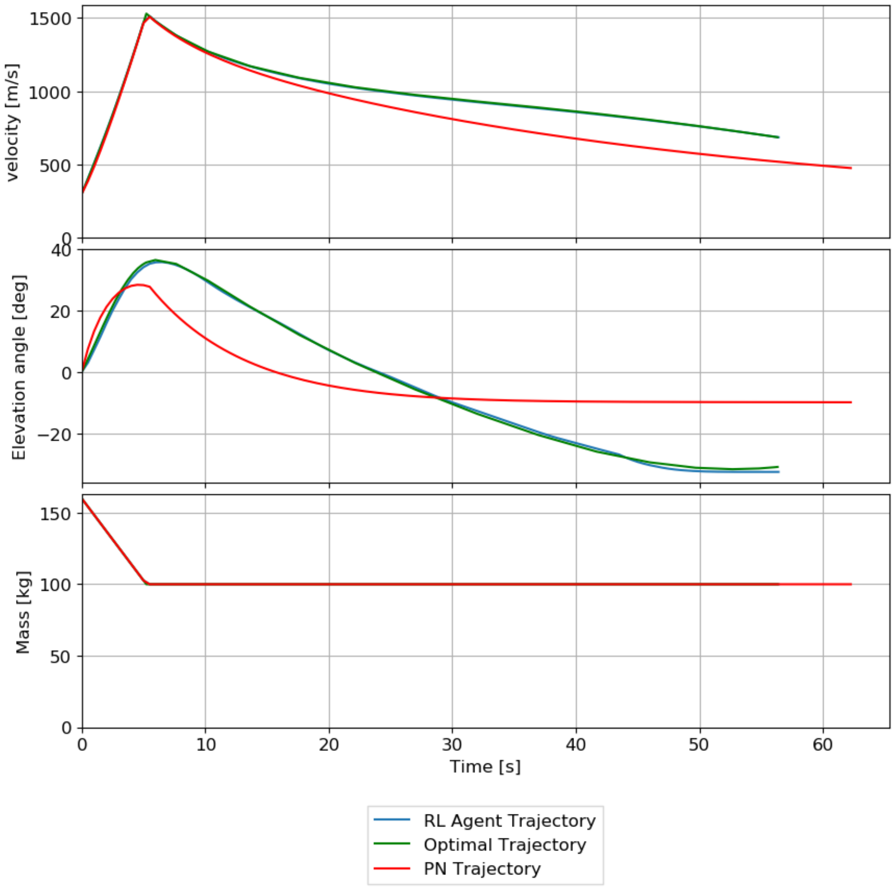


Figure 5.3: Velocity, elevation angle, and missile mass plotted versus time for missile trajectories obtained using PN guidance with lofting, RL agent based guidance, and the optimal trajectory

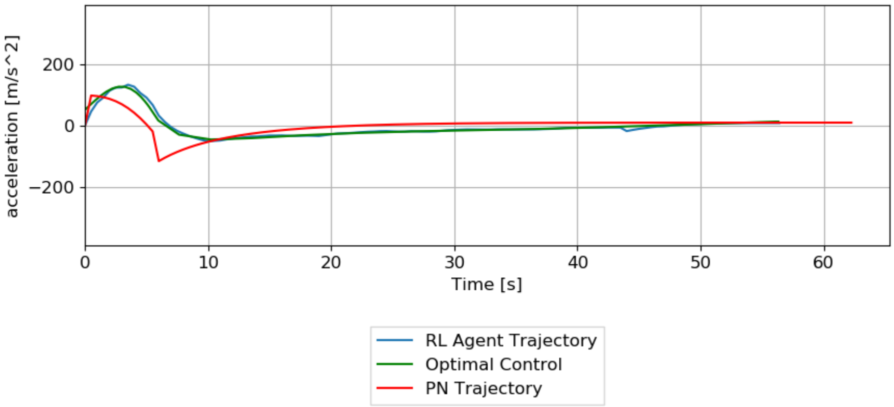


Figure 5.4: Commanded control plotted versus time using PN guidance with lofting, RL agent based guidance, and optimal control

5.1.2. Range Envelope

A single case does not yet prove that the method outperforms PN guidance or approximates optimal performance. Therefore, to establish a measure of performance, the maximum range is iteratively found for PN with lofting and the RL agent based guidance law. The theoretical maximum range is then found using optimal control. Plotting the results gives figure 5.5 where in figure 5.6 the same result is presented however this time normalized using the theoretical maximum range.

As can be seen, the RL agent based guidance achieves between approximately 60% and 90% of theoretical optimal range performance whereas the PN guidance law with lofting achieves between approximately 45% and 60% of this. The lofting does improve performance of the PN guidance law significantly as without lofting only between 20% and 30% of theoretical optimal performance is obtained. The optimal range is still significantly larger than the achieved using the RL agent based guidance, however a significant performance increase is achieved with respect to the PN guidance. An interesting observation is the increase of performance with altitude for the RL agent, at lower altitudes the normalized range is significantly lower. This is reasoned to be caused by the thicker atmosphere punishing non-optimal solutions more harshly. Thus a difference in behaviour with respect to optimal control will lead to a larger difference in performance at lower altitude.

Combined with the case study showing performance approximating performance obtained using trajectory optimization, it is established that the methodology is feasible, can increase performance over traditional guidance laws, and approximate performance obtained using trajectory optimization.

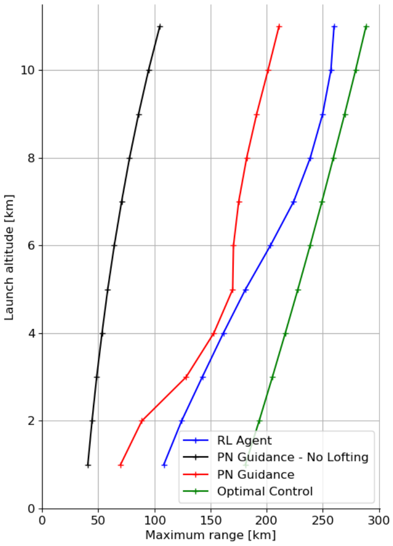


Figure 5.5: Maximum range achieved using PN guidance, PN guidance with lofting, RL agent based guidance and trajectory optimization at altitudes between 1 and 11 km

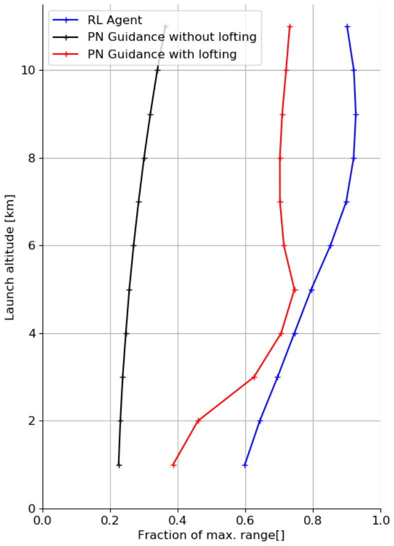


Figure 5.6: Normalized maximum range achieved using PN guidance, PN guidance with lofting and RL agent based guidance at altitudes between 1 and 11 km

5.2. Three-Degrees of Freedom - Non Moving Target

Using the 2-DoF environment it is shown that the method is feasible and able to improve on PN guidance. Removing the simplification of restricting the engagement to the vertical plane, a 3-DoF scenario is obtained. Again, an RL agent is trained for this environment and implemented in the framework as described in section 4.5. The target is set to have zero velocity and is thus stationary. By doing so the performance of the RL agent alone, thus without the extra complexity added by the PIP-algorithm, can be evaluated and compared to optimal control. The trajectory obtained using PN with lofting is also plotted to compare the developed RL based guidance law to a traditional guidance law.

5.2.1. Case Study

An engagement is initialized where the initial states of the missile and target can be found in tables 5.3 and 5.4. Trajectories are generated using the RL agent based guidance algorithm, PN guidance with lofting, and the optimal trajectory as obtained from trajectory optimization. These trajectories are presented in figure 5.7.

As can be seen the optimal trajectory and the trajectory generated using the RL guidance match closely and both reach the target in a very similar time. The RL agent is outperformed by optimal control and reaches the target 0.9% slower than the time obtained from optimal control. Differences in behaviour are observed mainly in the vertical plane. Figure 5.8 shows the RL agent choosing a higher trajectory leading to a slightly higher sustained velocity in the final stage of the trajectory as displayed in figure 5.9. This however does not lead to a shorter time of flight.

Table 5.3: Initial missile states

State	Value	Unit
x	-95	km
y	-50	km
z	5	km
V	300	m/s
γ	0	deg
ψ	90	deg
m	160	kg

Table 5.4: Initial target states

State	Value	Unit
x_{tgt}	0	km
y_{tgt}	0	km
z_{tgt}	5	km
V_{tgt}	0	m/s
γ_{tgt}	0	deg
ψ_{tgt}	0	deg

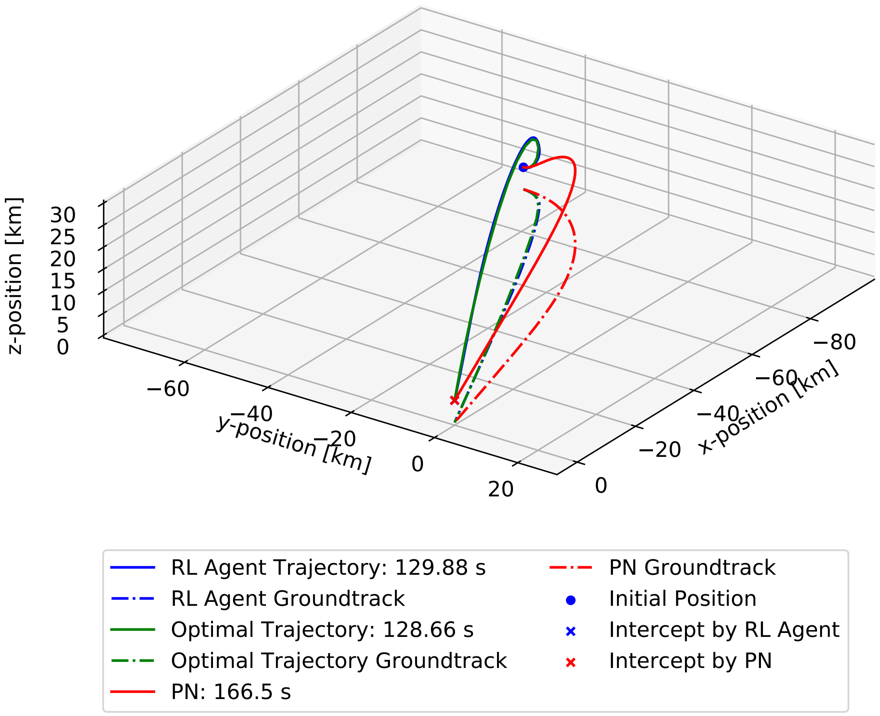


Figure 5.7: Resulting trajectories plotted for missile intercepting stationary target using PN guidance with lofting and RL agent based guidance

A clear disadvantage of the switching between guidance schemes can also be found in figure 5.9 and 5.10. Due to the change in guidance scheme a relatively large acceleration command is given between $t = 105$ and $t = 110$ seconds. At this point the velocity of the RL agent trajectory decreases significantly when compared to the optimal trajectory. Thus finding a way to improve or all together avoid this switching point would benefit the guidance scheme. The control commands provided by the RL agent show close correlation with the optimal control solution as seen in figure 5.10, the main differences being that the RL agent commands are more erratic and the switch to terminal guidance used for the RL agent based guidance law.

The trajectory generated using PN with lofting hits the target however achieves a time which is 29% slower than the time achieved by optimal control. Differences in behaviour can be found in both the horizontal and vertical plane. In the horizontal plane both the optimal and RL agent choose a more aggressive turn whilst the PN guidance performs a long and wide turn, this is clearly seen in figure 5.9. Due to this, the missile travels a longer distance instead of taking a direct approach to the target. The lofting algorithm elevates the missile to a compara-

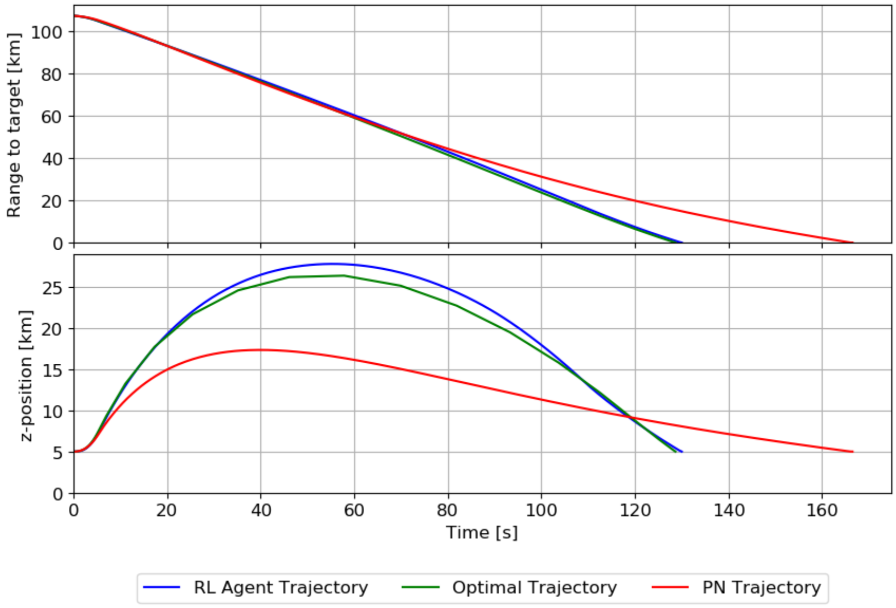


Figure 5.8: Range and z-position plotted versus time for missile trajectories obtained using PN and RL agent based guidance

ble elevation angle as obtained through optimal control however the PN guidance law quickly reduces this angle after initial lofting. Because of this it reaches a lower maximum altitude as and then chooses a more direct approach to the target. This leads to a lower velocities in the second half of the trajectory and a longer time of flight.

As is stated, this is a single case. The main conclusion which can be drawn from this case is that the methodology is feasible for a 3-DoF case and a stationary target. Furthermore, the methodology seems to approximate optimal control. An important note to make here is that the RL agent successfully generalizes learned behaviour since this exact case is unlikely to be encountered during training.

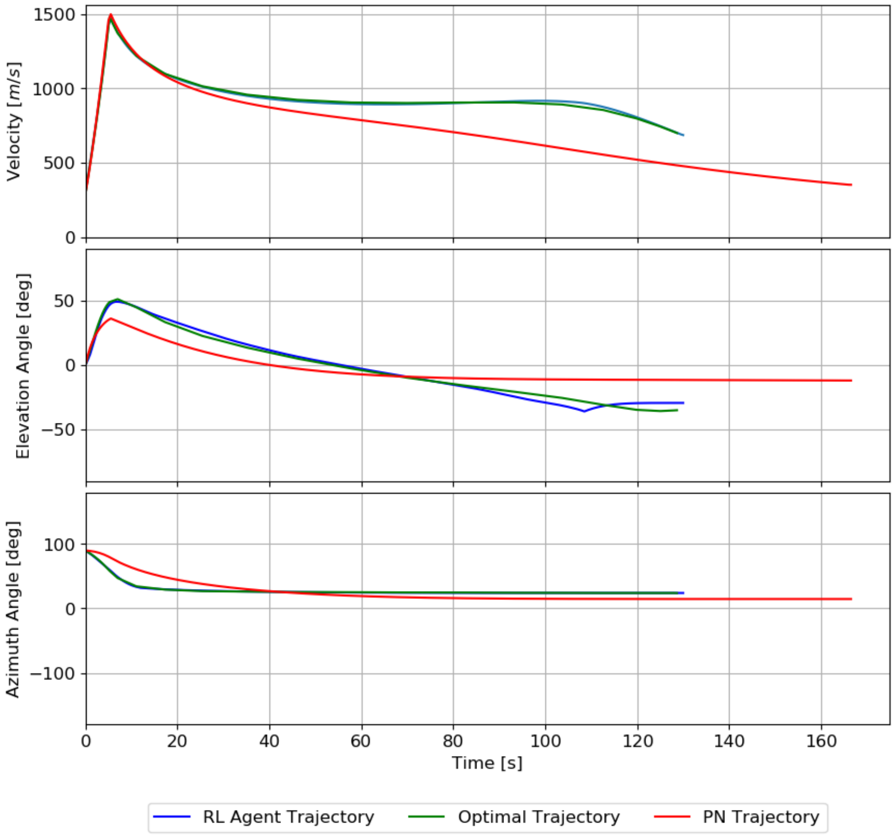


Figure 5.9: Velocity, elevation angle and azimuth angle plotted versus time for missile trajectories obtained using PN and RL agent based guidance

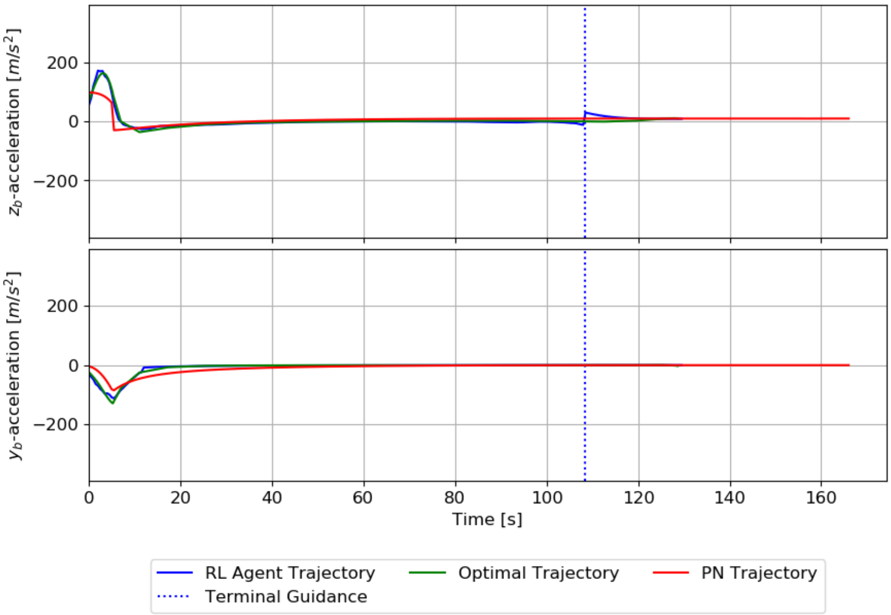


Figure 5.10: Control commands in missile body z and y directions plotted versus time for missile trajectories obtained using PN and RL agent based guidance

5.2.2. Range Envelope

To generalize performance the maximum range is found using several different guidance algorithms. This maximum range can be interpreted as R_{max} as defined in section 2.3.1. The guidance algorithms compared are PN, PN with lofting, the RL agent based guidance law and optimal control. For optimal control the objective function is redefined to optimize for maximum range whilst for the others the maximum range is found in an iterative manner. The maximum ranges are obtained for three different initial altitudes used for both missile and target, where both missile and target are initialized at the same altitude. The initial missile off-boresight angle is varied between $[-90, 90]$ degrees with a step of 15 degrees. The remaining initial states are defined in tables 5.5 and 5.6.

Table 5.5: Initial missile states

State	Value	Unit
x	Variable	km
y	Variable	km
z	[3, 6, 9]	km
V	300	m/s
γ	0	deg
ψ	Variable	deg
m	160	kg

Table 5.6: Initial target states

State	Value	Unit
x_{tgt}	0	km
y_{tgt}	0	km
z_{tgt}	[3, 6, 9]	km
V_{tgt}	0	m/s
γ_{tgt}	0	deg
ψ_{tgt}	0	deg

The results are then normalized using the maximum ranges obtained using optimal control and visualized in figures 5.11 - 5.13 where the missiles initial position is at the origin pointed in the 0 degrees direction. The non-normalized range results can be found in appendix A.1.

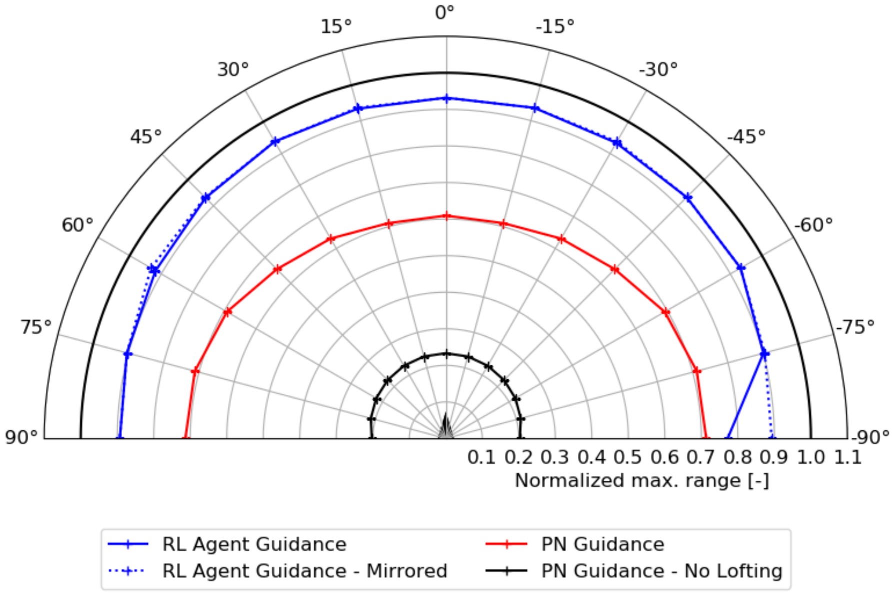


Figure 5.11: Maximum ranges plotted versus varying initial off-boresight angles at an altitude of 3 km for different guidance methods

It can be observed in figures 5.11 - 5.13 that the RL agent based guidance achieves approximately between 90% and 95% of the range obtained using optimal control excluding an outlier. PN without any lofting only attains between 20% and 30% of the range attained

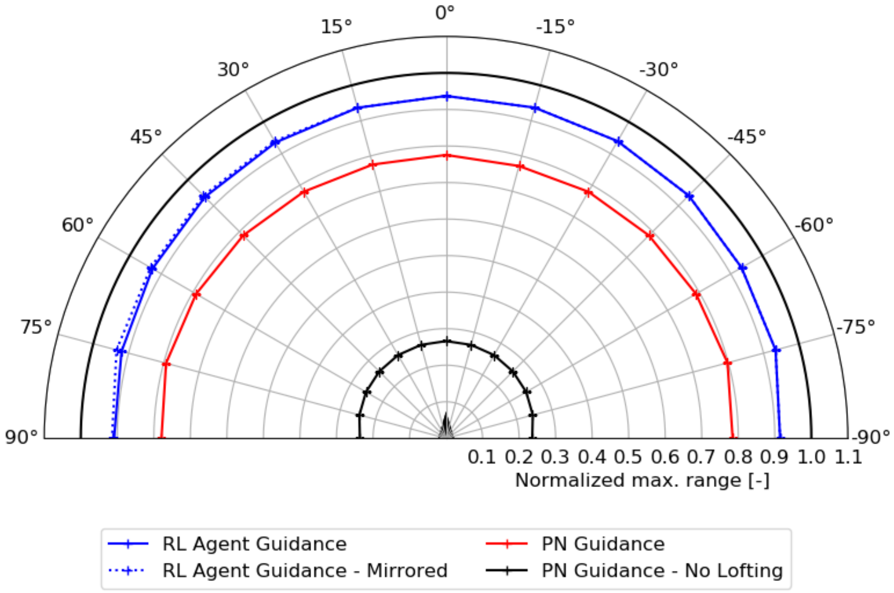


Figure 5.12: Maximum ranges plotted versus varying initial off-boresight angles at an altitude of 6 km for different guidance methods

using optimal control. Adding lofting greatly improves performance of PN especially at higher altitudes. At 9 km altitude the PN guidance with lofting approaches the performance of the RL agent based guidance law. It can however be concluded that the RL agent based guidance law achieves better range performance, especially at lower altitudes against stationary targets. This showcases the ability of the RL agent to optimize behaviour for every possible state whereas the lofting algorithm only works well in certain cases.

Two interesting observations are made in figure 5.11 at an off-boresight angle of −90 degrees and in figure 5.13 at an off-boresight angle of 90 degrees. In both cases the RL agent based guidance significantly under performs if compared to the other datapoints. Due to the method being based on RL it is possible that the agent does not have an appropriate solution for a state encountered during simulation. This would be due to insufficient learning coverage in the specific state region. This highlights a major disadvantage of the employed method, it cannot guarantee an optimal or even a suitable solution for every state both due to it being based on experience and the black box nature of the algorithm. The method does technically have a solution for these outliers due to the problem being symmetrical. Therefore, the outliers are not a limitation of the methodology itself but a result of the imperfect quality of training. To highlight the performance potential of the methodology, a line is added to figures 5.11 - 5.13 showing the range obtained when mirroring the left and right half of the results among the zero degrees axis and using the maximum from each resulting data series.

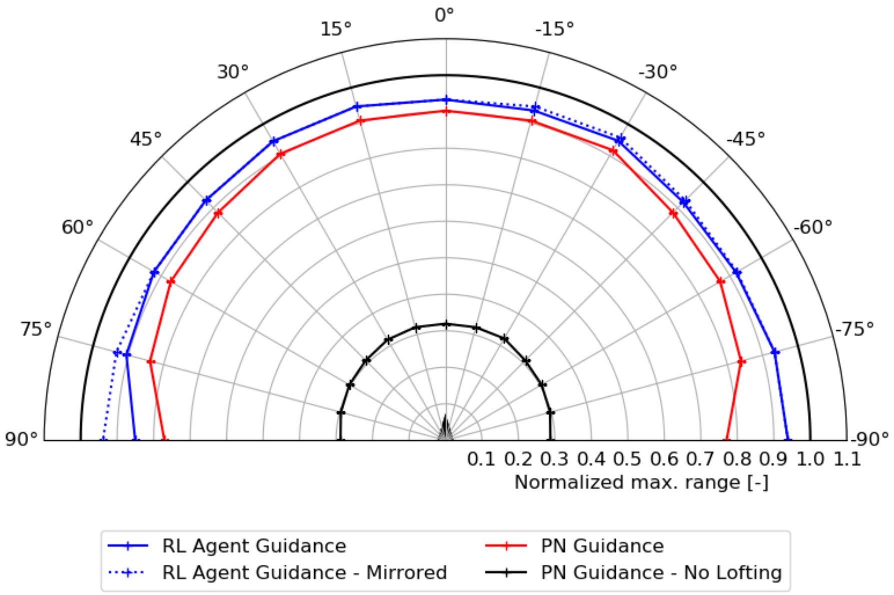


Figure 5.13: Maximum ranges plotted versus varying initial off-boresight angles at an altitude of 9 km for different guidance methods

5.2.3. Time of Flight

It is established that the RL agent based guidance law increases maximum range over PN, especially at lower altitudes. Furthermore, the maximum range achieves approximately 90% to 95% of the maximum range obtained from optimal control. However there are also many scenarios at lower ranges. Therefore the time of flight to a stationary target at varying ranges and off-boresight angles is evaluated as a secondary performance indicator. The same initial conditions are used as for the maximum range evaluation, however in this case the range is not iteratively varied but set at specified values. The results of comparing the RL agent to PN with lofting and optimal control at 6 km altitude can be found in figures 5.14 and 5.15 respectively. In appendix A.2 the figures showing absolute times can be found as well the same relative results at 3 and 9 km altitude.

As can be seen from figure 5.14, PN with lofting is outperformed significantly over the domain excluding two datapoints (at off-boresight angle of 30 degrees and ranges 60 and 70 km). The performance increase using the RL agent based guidance law in terms of time of flight is in excess of 15% over large parts of the domain. In the scenario in which the described outliers are obtained, the target is hit however the time in which this is done is slower than that obtained using PN with lofting. This region of sub-optimal performance by the RL agent is more pronounced at an altitude of 3 km. These results both highlights the performance of the RL methodology and also a major disadvantage. The major disadvantage being that an appropriate solution is not guaranteed. Similar as in section 5.2.2, the symmetry of the domain proves that this is not a limitation of the methodology but a result of the quality of training where there is still potential for improvement.

When comparing the results obtained to optimal control solutions, figure 5.15 is obtained. As can be seen, the RL agent performs very similarly to optimal control again excluding the two mentioned datapoints. The time of flight lies within 5% of the time of flight achieved using optimal control over large regions of the domain. Again, the symmetry of the domain

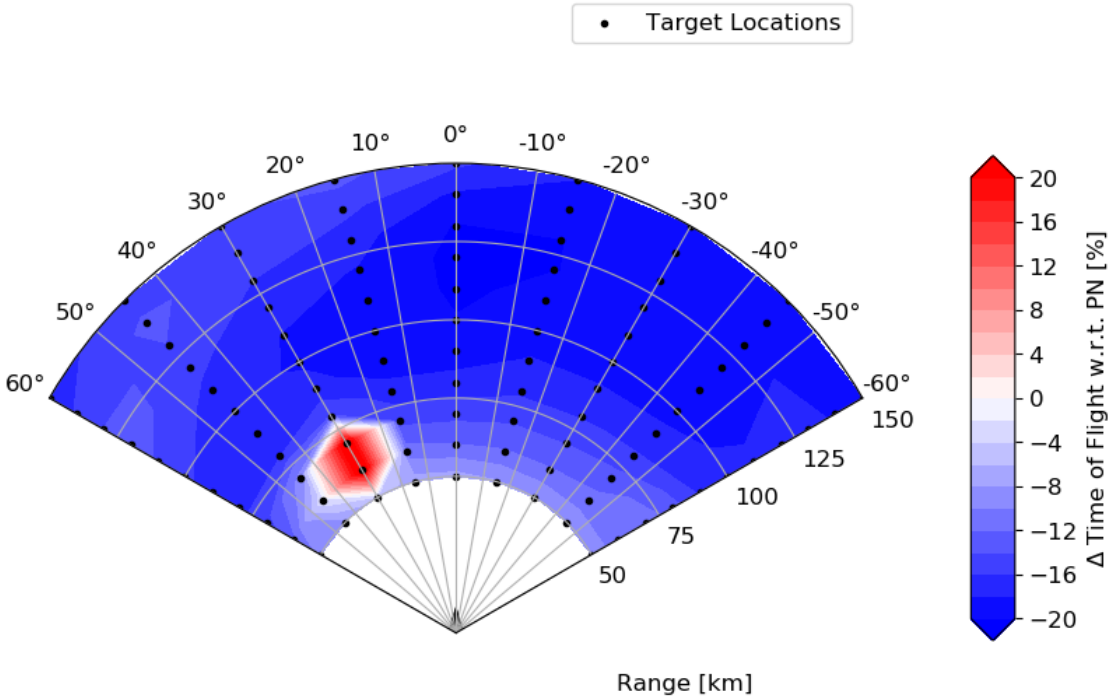


Figure 5.14: Time of flight achieved using RL agent compared to PN guidance with lofting solutions at 6 km altitude

could be exploited to improve these results where it can be seen that performance at negative off-boresight angles is superior to that at positive off-boresight angles. However these outliers do highlight the drawback of the solution and are therefore deemed relevant to the report.

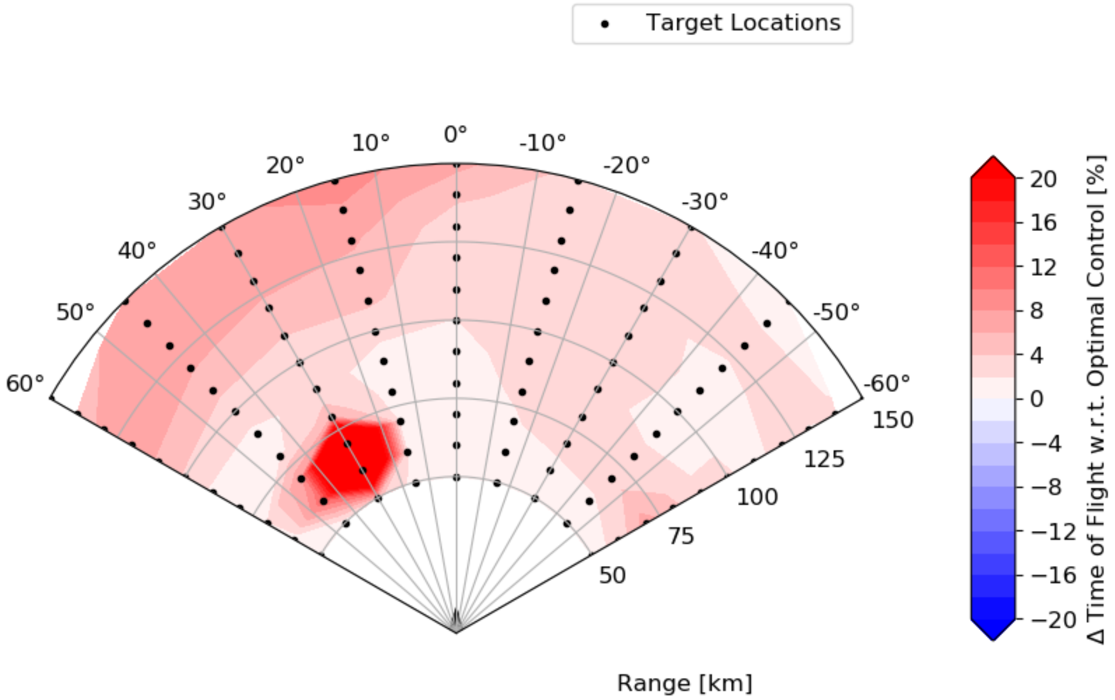


Figure 5.15: Time of flight achieved using RL agent compared to optimal control solutions at 6 km altitude

5.3. Three-Degrees of Freedom - Turn and Run Maneuver

Again a simplification is removed and the target is now non-stationary and performs a scripted maneuver. This scripted maneuver is the so called turn and run maneuver as described in section 2.4.6. A case study is presented after which the maximum range is found for varying scenarios.

5.3.1. Case Study

Two cases are presented using the same initial conditions using only a different PIP-mode. In both cases the solutions obtained using optimal control and PN guidance with lofting are also visualized. The initial states are presented in tables 5.7 and 5.8. A note which should be made here is that the target starts at its maximum velocity and thus does not accelerate after turning.

In figures 5.16 and 5.17 the trajectories generated by the RL agent based guidance are visualized using PIP-mode 1 and PIP-mode 2 respectively. The associated PN with lofting and optimal control trajectories are also plotted. It should be noted here that PN guidance with lofting does not intercept the target which is not directly visible from the plots.

Table 5.7: Initial missile states

State	Value	Unit
x	-100	km
y	0	km
z	5	km
V	300	m/s
γ	0	deg
ψ	60	deg
m	160	kg

Table 5.8: Initial target states

State	Value	Unit
x_{tgt}	0	km
y_{tgt}	0	km
z_{tgt}	5	km
V_{tgt}	250	m/s
γ_{tgt}	0	deg
ψ_{tgt}	180	deg

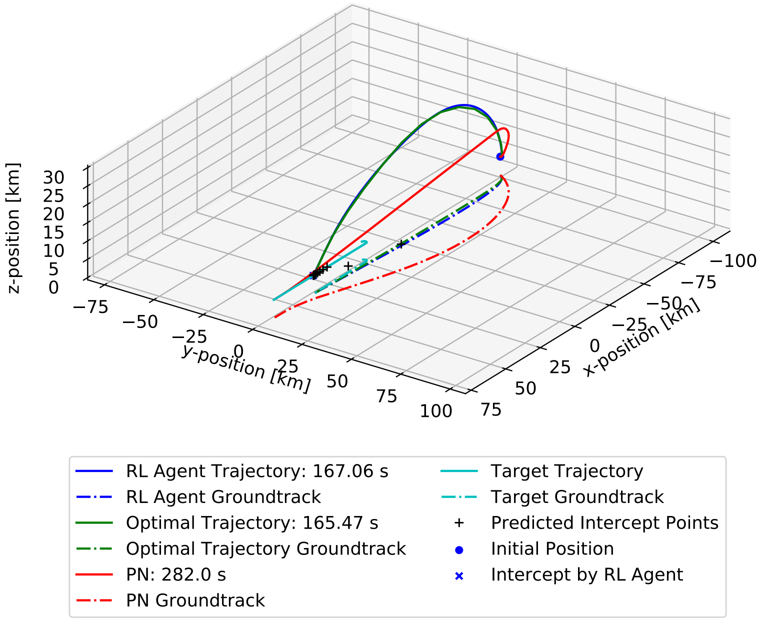


Figure 5.16: Resulting trajectories plotted for missile intercepting target performing turn and run maneuver using PN guidance with lofting and RL agent based guidance using PIP-mode 1

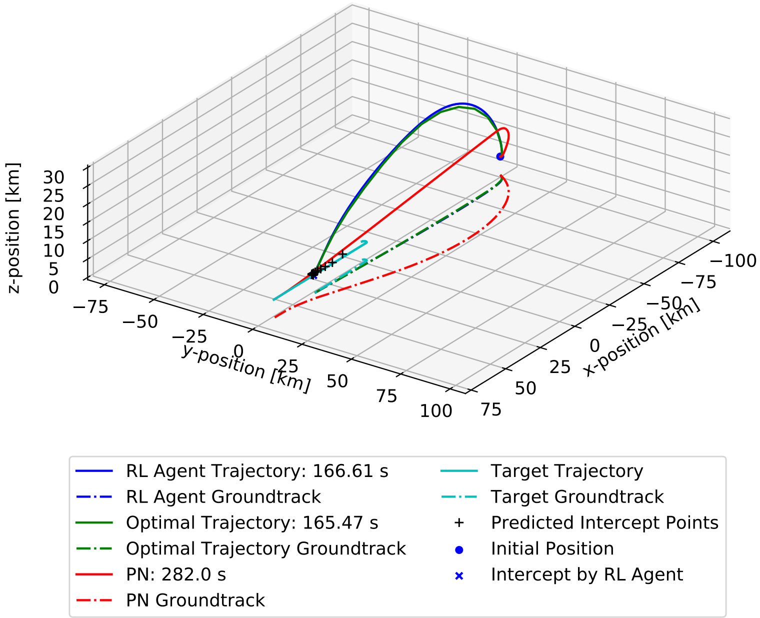


Figure 5.17: Resulting trajectories plotted for missile intercepting target performing turn and run maneuver using PN guidance with lofting and RL agent based guidance using PIP-mode 2

Comparing RL agent based guidance to optimal control, the first observation which can be made is that both the RL agent with PIP-mode 1 and with PIP-mode 2 perform very similarly to the optimal control solution. Furthermore the trajectories generated using each PIP-modes are very similar. The time of flight is within 1% of the optimal solution for both scenarios where the RL agent combined with PIP-mode 2 performs slightly better. This is however to be expected since the targets behaviour almost directly correlates with the assumption made for target behaviour in PIP-mode 2. Both PIP-modes combined with the RL agent approximate optimal performance in a scenario with a maneuvering target and thus uncertain future target states, showcasing the potential of the methodology. It should be noted that after the initial turn the target acts very predictable by performing no more maneuvers, thus both PIP-modes predict its future states relatively accurately.

When comparing the RL agent based guidance using either PIP-mode to PN guidance it is clear that the RL agent performs better due to it successfully hitting the target whereas PN guidance does not. The PN guidance ends up in a tail chase where it loses velocity in thick atmosphere leading to its PIP lying further in the future. The missile will then have a longer time-to-go which in term leads to lower velocities. This is obviously a vicious circle leading to a miss.

Due to the small difference in behaviour between the two scenarios, only relevant states resulting from using PIP-mode 2 are set out. The remaining figures for the scenario using PIP-mode 1 can be found in appendix A.3. Figure 5.18 plots the range to target and z-position of the missile versus time. What can be observed is that the solutions obtained using optimal control and the RL agent based guidance have a more negative range rate compared to PN guidance. This can be attributed to the energy initially used to reach a higher altitude after which the beneficial conditions (thinner atmosphere) translate into a higher sustained velocity which is visualized in figure 5.19. Furthermore, the RL agent based guidance again chooses to turn more aggressively in the horizontal plane as displayed in figure 5.19 where the azimuth angle drops significantly faster. This in term leads to a more direct approach to the target.

The RL agent based guidance and optimal control solution effectively chooses to use more energy before the maximum speed is reached to transfer the missile to a more beneficial state. Using energy before the maximum speed is reached is more effective than after. This because a higher kinetic energy level is associated with more drag whilst a higher potential energy level (in this case altitude) is associated with less drag (thinner atmosphere). Thus, energy which would be expended faster due to a higher peak velocity, is now used more effectively to reach a higher altitude. The lofting algorithm implemented for PN guidance attempts the same however is not as effective at it.

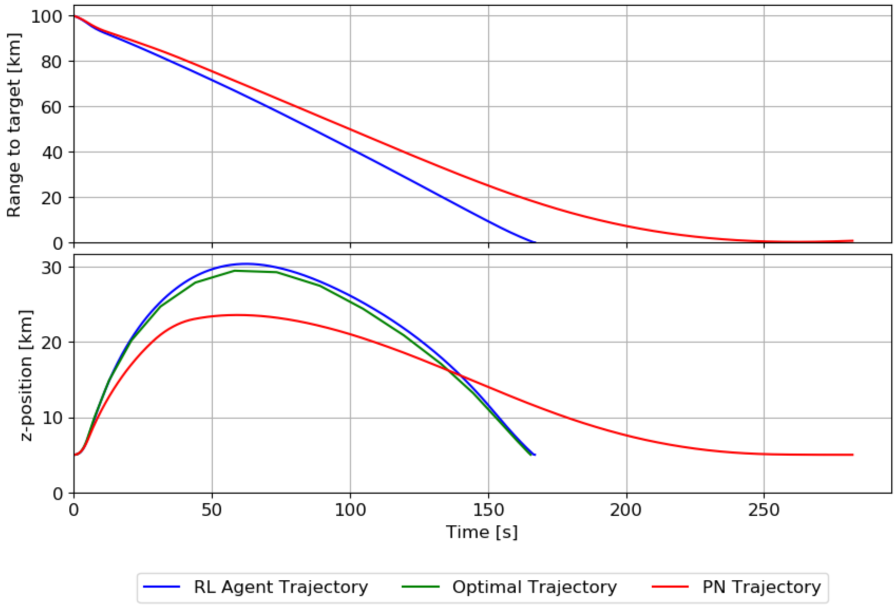


Figure 5.18: Range and z-position plotted versus time for missile trajectories obtained using PN and RL agent based guidance

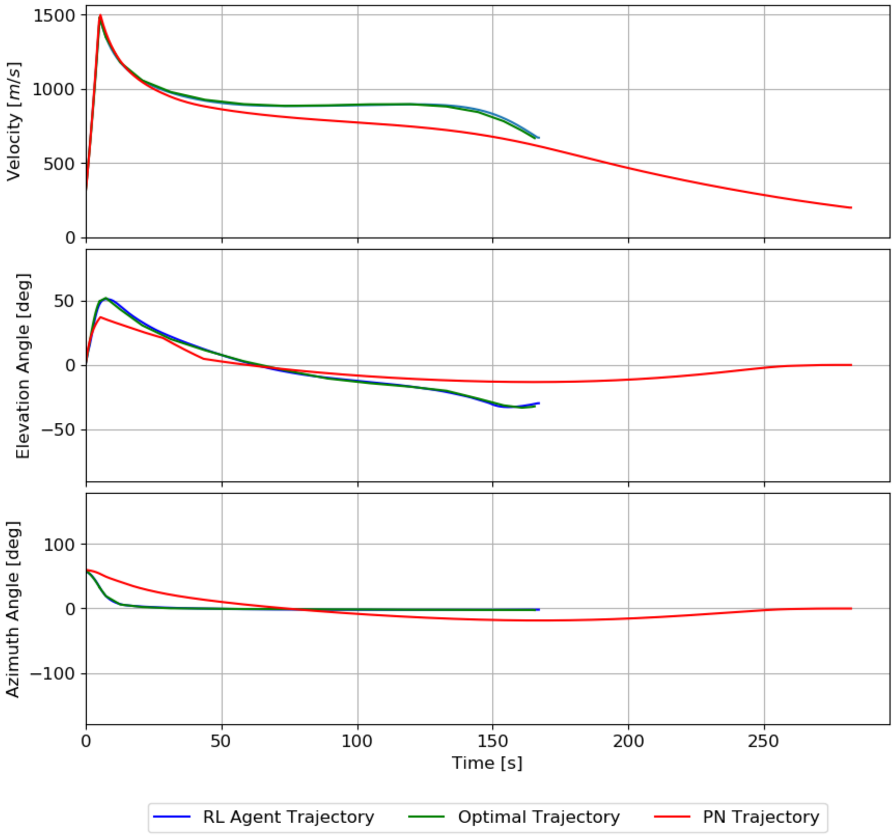


Figure 5.19: Velocity, elevation angle and azimuth angle plotted versus time for missile trajectories obtained using PN and RL agent based guidance

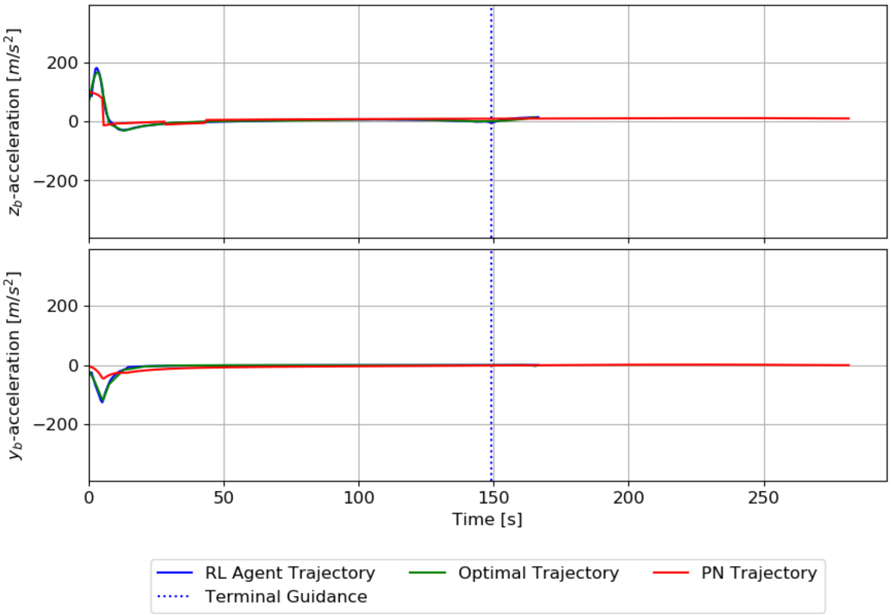


Figure 5.20: Control commands in missile body z and y directions plotted versus time for missile trajectories obtained using PN and RL agent based guidance

5.3.2. RANGE ENVELOPE

Using the same methodology as described in section 5.2.2, only this time using a target performing the turn and run maneuver, figures 5.21 - 5.23 are obtained. The results are again normalized using optimal control. The non-normalized results can be found in appendix A.4. Both PIP-modes are again employed and compared.

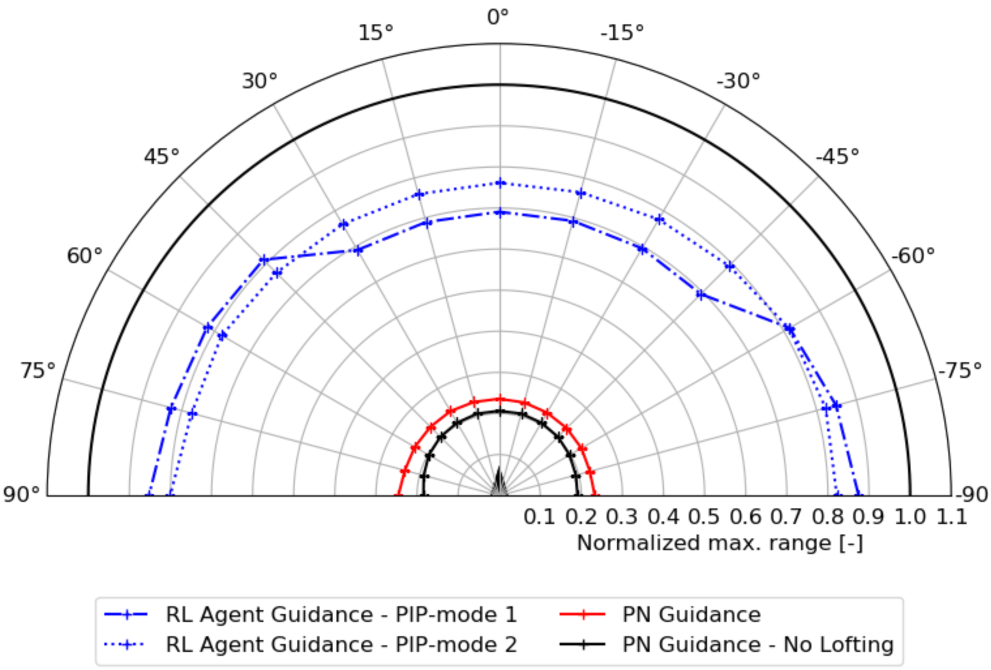


Figure 5.21: Normalized maximum launch ranges plotted versus varying initial off-boresight angles at an altitude of 3 km for different guidance methods versus a target performing a turn and run maneuver

Several observations can be made. First of all, when compared to the maximum range obtained using optimal control, PIP-mode 2 achieves between approximately 75% and 90% of maximum range depending on altitude barring several outliers. For PIP-mode 1 very similar results are obtained, only at 3 km altitude a significant difference is observed between off-boresight angles between 30 deg and −45 deg. Since the same RL agent is used, this is most likely due to the interaction between the PIP algorithm and RL agent leading to undesirable behaviour. PIP-mode 1 accounts for lateral movement of the target which is present in these scenarios whereas PIP-mode 2 does not. At low initial off-boresight angles this difference is especially prevalent since in those cases horizontal maneuvers are redundant whereas at high off-boresight angles the missile should be turning towards the target regardless of its maneuvers.

The aforementioned negative outliers will not be discussed since they are deemed to be caused by the same reasons as set out in section 5.2.2. One positive outlier however is highlighted which can be found at an altitude of 6 km and an off-boresight angle of 90 deg. PIP mode 1 significantly outperforms PIP-mode 2 and achieves approximately 90% of range obtained using optimal control. The combination of PIP-mode and RL agent performs very well in this specific instance, raising the question if more performance can be attained when the PIP algorithm is further developed. However, it might also be a very favourable combination of

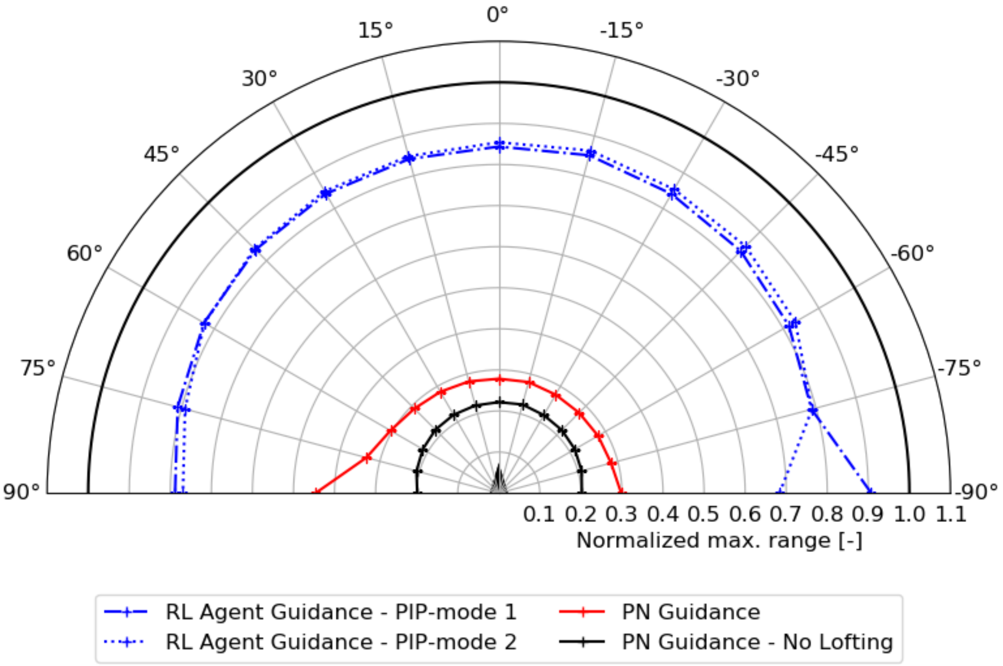


Figure 5.22: Normalized maximum launch ranges plotted versus varying initial off-boresight angles at an altitude of 6 km for different guidance methods versus a target performing a turn and run maneuver

specific circumstances which cannot effectively be generalized. This is not further expanded on in this research but is a potential point of future development.

Comparing RL agent based guidance to PN guidance it is clear that PN guidance is outperformed significantly, this is especially true at lower altitudes. PN guidance achieves between approximately 25% and 75% of the maximum range obtained using optimal control dependant on altitude. The RL agent based guidance outperforms PN over the entire tested domain.

If compared to the stationary target envelopes, the relative difference between PN and RL based agent guidance is larger in the turn and run scenario. This is reasoned to have two reasons, namely the higher maintained velocity of the RL agent based guidance and the fact that the PN guidance enters in a tail chase. Logically the higher maintained speed will lead to a higher closing velocity and a earlier intercept due to which the target travels less from its initial position. Secondly, the PN guidance levels of in a tail chase behind the target due to the target flying away from the missile. The RL agent steering for a PIP will lead to a more direct trajectory and vertical approach to the target. Due to a relatively high velocity with respect to the target in the terminal phase, the terminal guidance law does not have to correct as much and does not enter in a tail chase. This effect is visualized in figure 5.18 where range to target and altitude profiles are given. As can be seen the PN guidance algorithm flies through the denser atmosphere at a lower altitude for longer. This explains the large difference in performance of the PN guidance law when compared to the case where a stationary target is intercepted.

Secondly it can be seen that the results, especially the PN envelope, is quite asymmetrical. This is the result of the direction of the targets turn where it always performs a right handed turn. For the PN guidance this results in a more aggressive turn at positive off-boresight angles

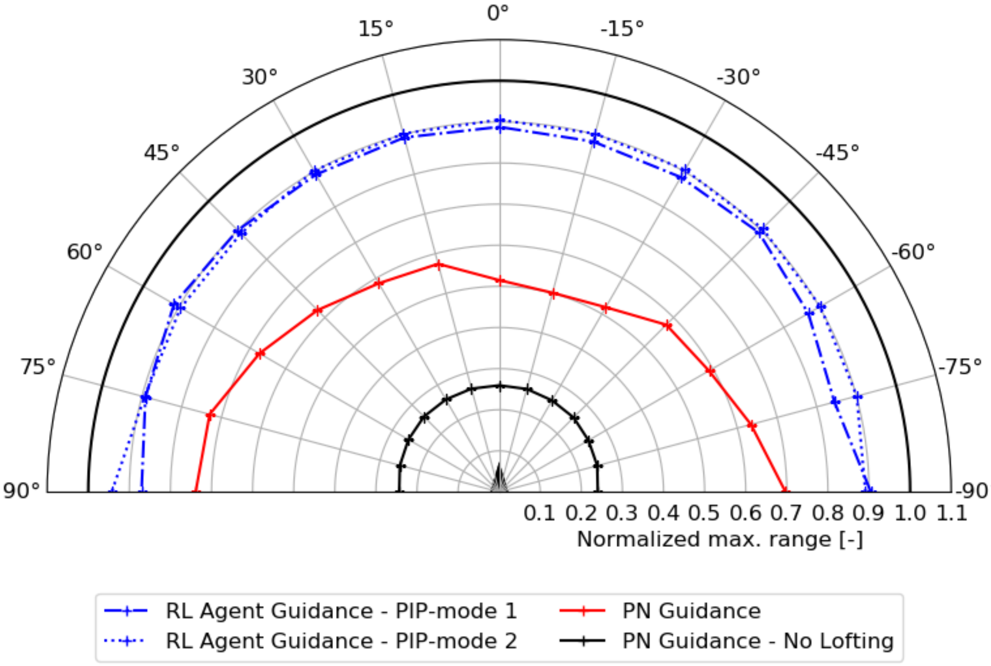


Figure 5.23: Normalized maximum launch ranges plotted versus varying initial off-boresight angles at an altitude of 9 km for different guidance methods versus a target performing a turn and run maneuver

due to the LOS-rate increasing. This in term leads to the missile more directly approaching the target. Furthermore, the maximum range is lower for targets at low off-boresight angles. This is attributed to the added lofting bias being overruled by the PN guidance earlier. Due to the missile flying more directly at the target a higher vertical LOS-rate is achieved leading to a more negative PN guidance command in the missile body z-direction. This in term, combined with the relatively poor performance of PN with lofting at low altitudes, leads to the conclusion that the PN guidance could still be improved. However it is also argued that this is not the goal of this research and the current implementation gives a reasonable baseline.

5.4. THREE-DEGREES OF FREEDOM - MANEUVERING TARGET

To ensure the algorithm is robust and can handle unpredictable target behaviour, random target behaviour as described in section 2.4.6. First a case study is presented using this target behaviour, after which hit probabilities are found for different initial conditions.

5.4.1. CASE STUDIES

Again two scenarios are presented using PIP-mode 1 and PIP-mode 2 respectively. The initial states and target states are presented in tables 5.9 and 5.10 and the target behaviour is determined randomly using the process as described in section 2.4.6.

When comparing the RL agent based guidance combined with PIP-mode 1 to PN guidance it is found that, in this specific scenario, both methods achieve success in hitting the target. However the RL agent based guidance outperforms PN significantly if looking at time of

Table 5.9: Initial missile states

State	Value	Unit
x	-50	km
y	-50	km
z	5	km
V	300	m/s
γ	0	deg
ψ	60	deg
m	160	kg

Table 5.10: Initial target states

State	Value	Unit
x_{tgt}	0	km
y_{tgt}	0	km
z_{tgt}	8	km
V_{tgt}	250	m/s
γ_{tgt}	0	deg
ψ_{tgt}	180	deg

flight as can be seen in figure 5.24. The RL agent reaches the target 17.5 % faster. Furthermore, the RL agent based guidance also has a higher terminal velocity as can be seen in figure 5.27. Looking at figure 5.28, the varying PIP due to target maneuvers does not seem to cause any issues with regards to the guidance command determined by the RL agent.

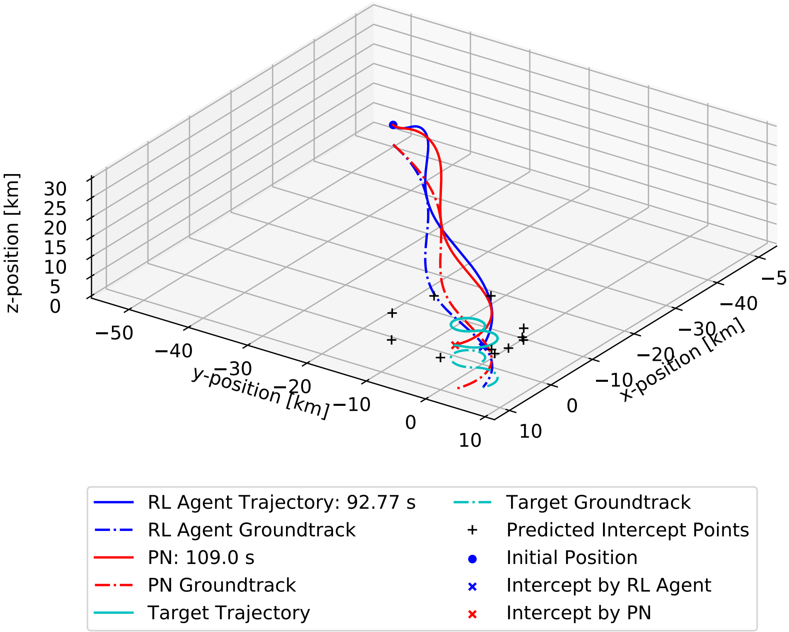


Figure 5.24: Resulting trajectories plotted for missile intercepting random maneuvering target using PN guidance with lofting and RL agent based guidance combined with PIP-mode 1

In figure 5.25 a close-up of the terminal stage is provided where it is clear that the RL agent based guidance intercepts the target significantly earlier. As is stated before, this faster intercept effectively means the target has less time to fly away from the missile, effectively increasing its range. The higher terminal energy is favourable for its hit probability being able to use this additional energy to correct for target maneuvers.

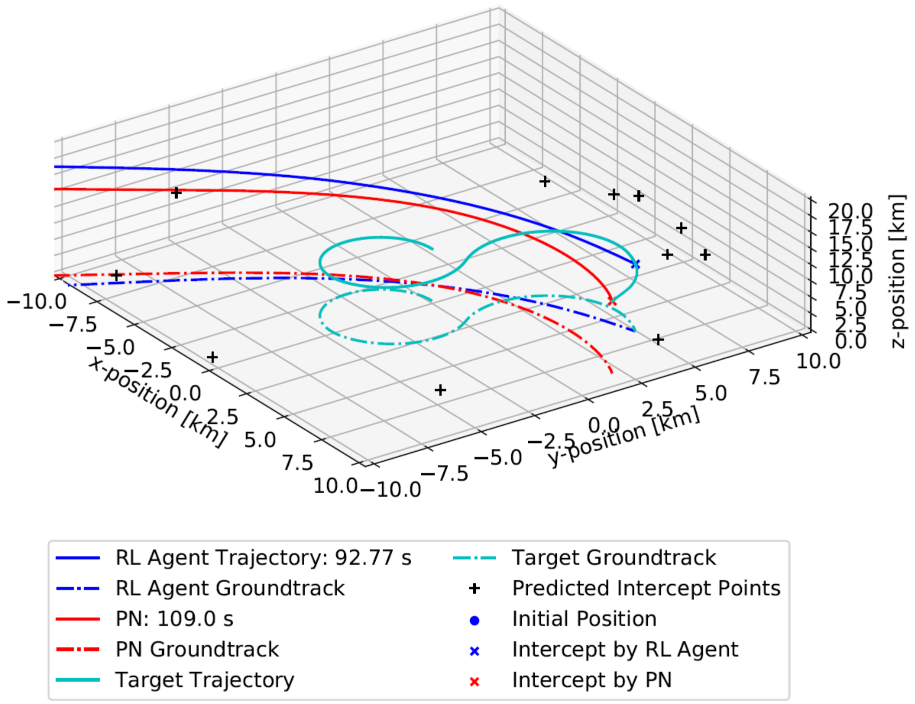


Figure 5.25: Resulting terminal phase trajectories plotted for missile intercepting random maneuvering target using PN guidance with lofting and RL agent based guidance combined with PIP-mode 1

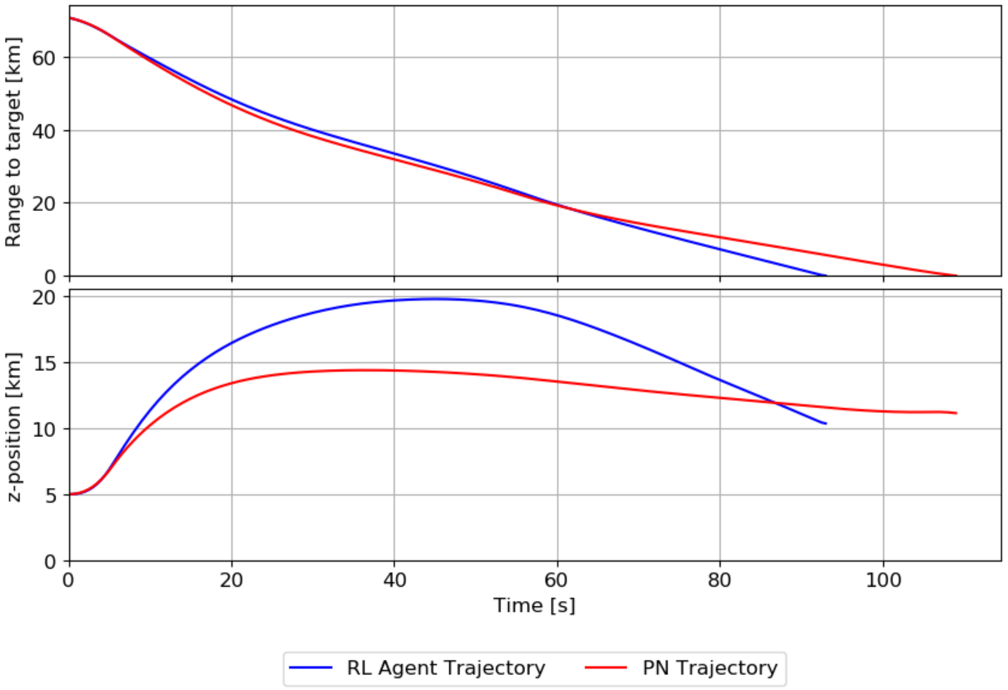


Figure 5.26: Range and z-position plotted versus time for missile trajectories obtained using PN guidance with lofting and RL agent based guidance combined with PIP-mode 1

It can be concluded that the same beneficial behaviour is observed as observed when the target is stationary. The added variation in observation due to target behaviour is handled

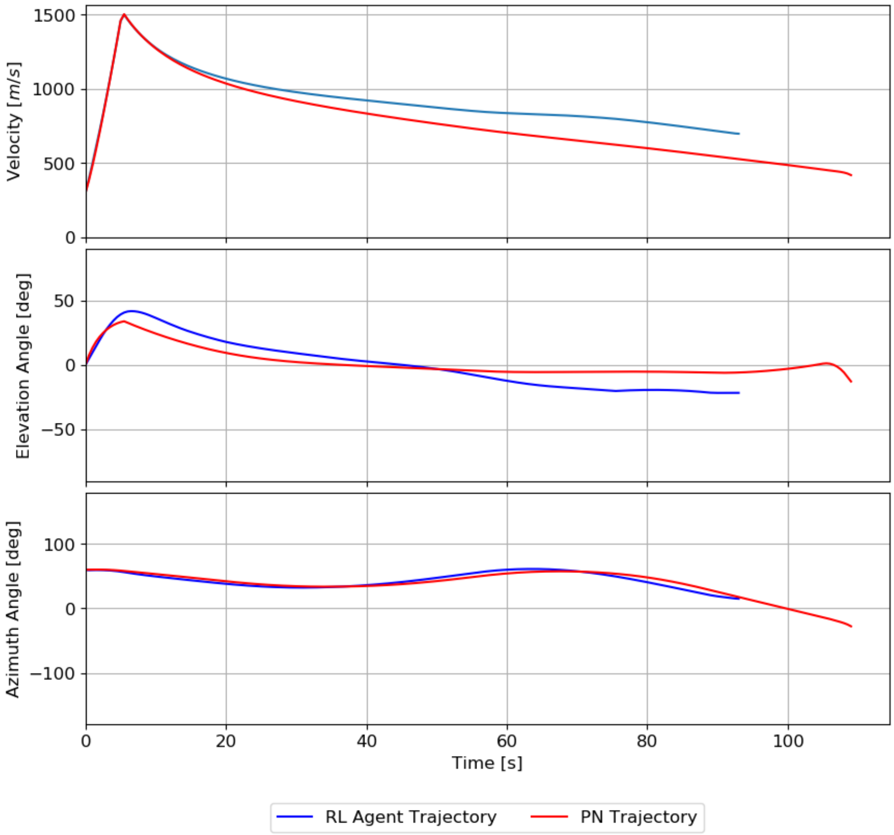


Figure 5.27: Velocity, elevation and azimuth angles plotted versus time for missile trajectories obtained using PN guidance with lofting and RL agent based guidance combined with PIP-mode 1

adequately by the RL agent. Furthermore, the hand over to PN guidance does not significantly hamper the performance of the guidance scheme due to it being switched over before the PN commanded acceleration becomes high. By switching before an extreme command is required by the terminal guidance, velocity or kinetic energy losses are minimized at this handover. If this handover would happen later, the benefits of the guidance scheme could be diminished by the handover.

When using PIP-mode 2 in the same scenario the trajectory as seen in figure 5.29 is obtained where it can be seen that the time of flight is slightly higher. The main difference observed between the two PIP-modes is that using PIP-mode 2 a higher maximum altitude is reached. The higher altitude is a result of the PIP lying further away from the missile using PIP-mode 2 leading the RL agent to steer higher to account for this.

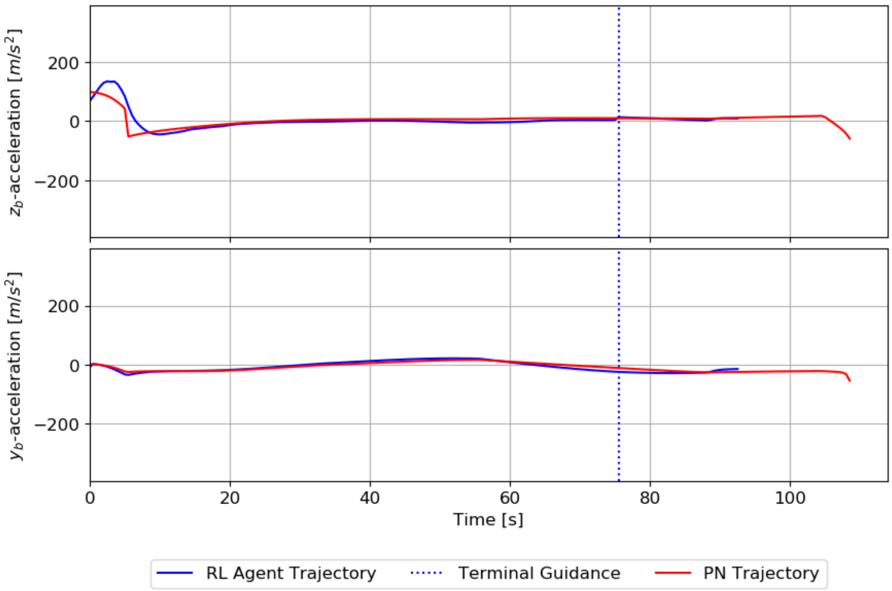


Figure 5.28: Control commands in missile body z and y directions plotted versus time for missile trajectories obtained using PN guidance with lofting and RL agent based guidance combined with PIP-mode 1

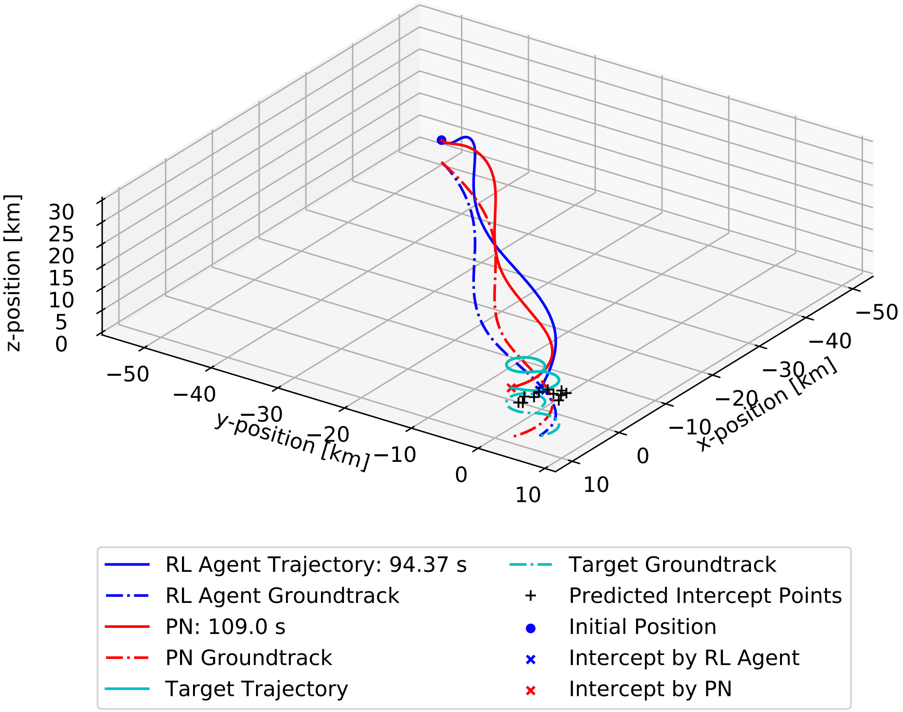


Figure 5.29: Resulting trajectories plotted for missile intercepting random maneuvering target using PN guidance with lofting and RL agent based guidance combined with PIP-mode 2

5.4.2. Launch Envelopes

Using the random target maneuver algorithm, a parameter sweep is performed varying several parameters. The missile is initialized to fly directly at the target, thus the off-boresight angle is zero. Both missile and target are started at the same altitude, which is either 3, 6, or 9 km. The targets initial azimuth angle is varied from 0 to −180 degrees in steps of 20 degrees and the range is varied between 100 and 200 km in steps of 10 km. The remaining states are given in tables 5.11 and 5.12. Each resulting set of initial states is initialized 50 times for each guidance method. By seeding the target maneuvers it is guaranteed that the same target maneuvers are used for each guidance method. This results in a unique set of 50 different target trajectories for each set of initial states used for each guidance method. The guidance methods used are PN with lofting, and RL agent based guidance using PIP-mode 1 and PIP-mode 2. The resulting hit probabilities at an altitude of 6 km are then shown in figures 5.30, 5.31, and 5.32 for each guidance method respectively. Two contours are including outlining zones with over 90% hit probability and over 50% hit probability.

Table 5.11: Initial missile states

State	Value	Unit
x	Variable	km
y	Variable	km
z	[3, 6, 9]	km
V	300	m/s
γ	0	deg
ψ	Variable	deg
m	160	kg

Table 5.12: Initial target states

State	Value	Unit
x_{tgt}	0	km
y_{tgt}	0	km
z_{tgt}	[3, 6, 9]	km
V_{tgt}	250	m/s
γ_{tgt}	0	deg
ψ_{tgt}	0	deg

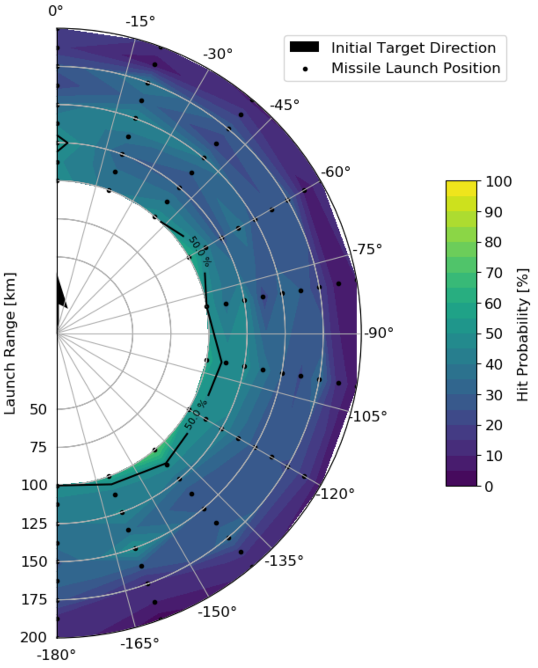


Figure 5.30: Hit probability envelopes for PN guidance with lofting at an altitude of 6 km

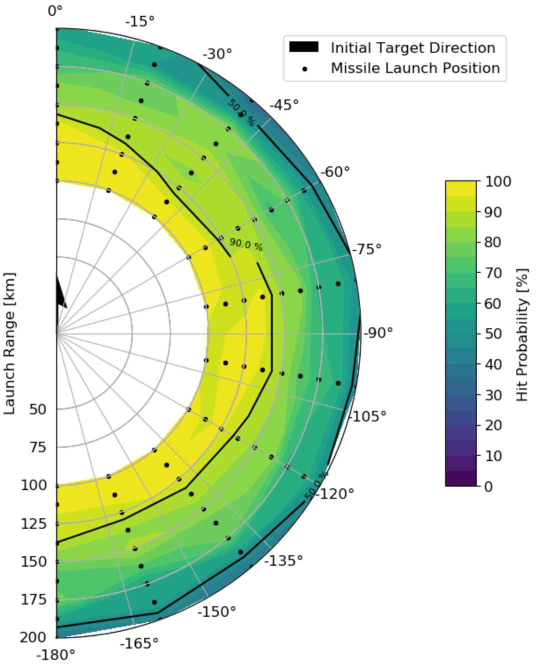


Figure 5.31: Hit probability envelopes for RL agent based guidance using PIP-mode 1 at an altitude of 6 km

Comparing figure 5.30 to figures 5.31 and 5.32 it is quite clear that the RL agent based guidance using either PIP-mode 1 or PIP-mode 2 outperforms PN guidance with lofting for the

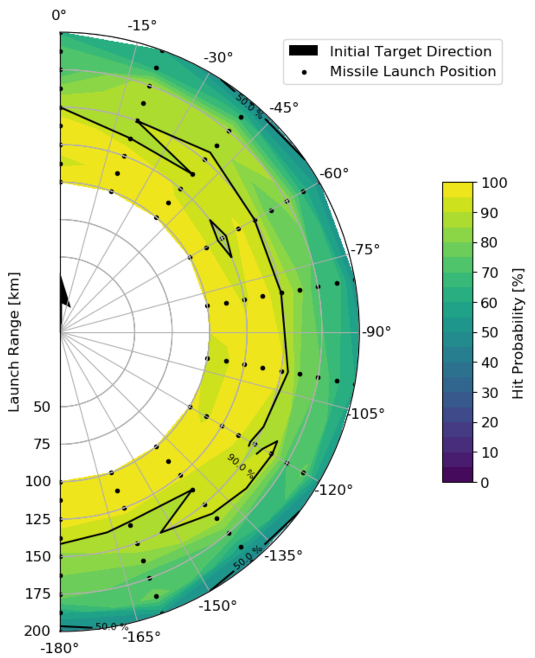


Figure 5.32: Hit probability envelopes for RL agent based guidance using PIP-mode 2 at an altitude of 6 km

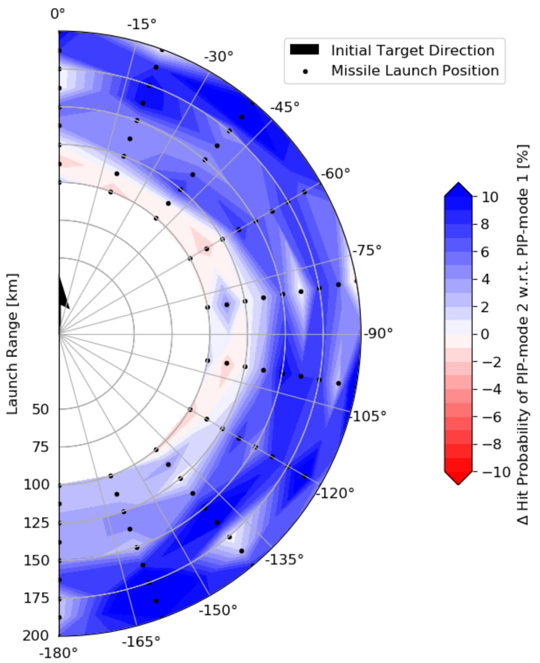


Figure 5.33: Difference in hit probability envelopes between RL agent based guidance using PIP-mode 1 and PIP-mode 2 at an altitude of 6 km

specified envelope. Whilst the RL agent based guidance using either PIP-mode has a significant envelope in which a 90% hit probability is found, the PN guidance using lofting barely achieves 50% at a range of 100 km. Looking at the results for each guidance method at altitudes of 3 km, 6 km and 9 km in figures A.27 - A.35 it is found that PN with lofting is significantly outperformed at all three altitudes by RL agent based guidance using either PIP-mode.

Apart from outperforming PN with lofting, the presented results show robustness of the developed guidance law. The RL agent based guidance law is tested in 4.5×10^3 unique engagements each consisting of in the order of 1×10^2 transitions. Consistent and comparatively high performance is shown using either PIP-mode. The high level of success achieved using either PIP-mode should speak in favour of the robustness of the developed guidance method.

To compare the PIP-modes, figure 5.33 is given. It shows the absolute difference between the hit probabilities at 6 km altitude when using PIP-mode 1 or PIP-mode 2 where blue is in favour of PIP-mode 2. It is clear that PIP-mode 2 outperforms PIP-mode 1 significantly. This is also holds at altitudes of 3 km and 9 km as can be seen in figures A.36 and A.38.

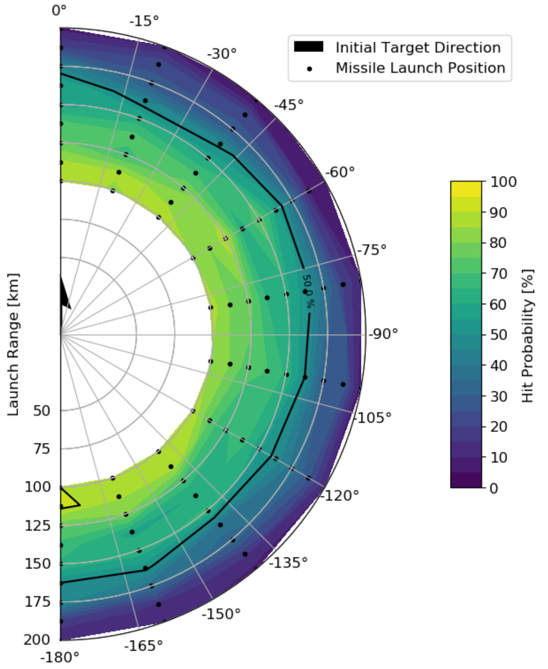


Figure 5.34: Hit probability envelopes for RL agent based guidance using PIP-mode 2 at an altitude of 3 km

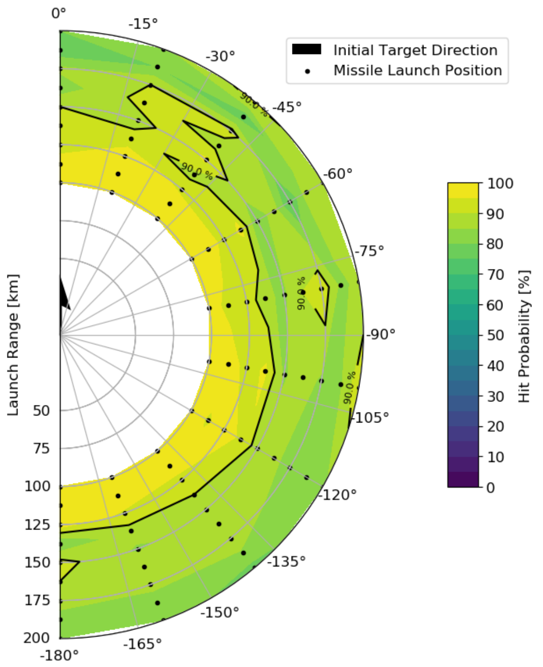


Figure 5.35: Hit probability envelopes for RL agent based guidance using PIP-mode 2 at an altitude of 9 km

5.5. DYNAMIC LAUNCH ZONES

In sections 5.2.2, 5.3.2, and 5.4.2 range results are presented for different scenarios. In this section these results are combined to represent a DLZ as defined in section 2.3.1. Three of these presented ranges can be found using the obtained results, these are the maximum aerodynamic range R_{aero}, the maximum turn and run range R_{tr}, and the maximum no escape range $R_{max,2}$. For this specific representation they are found at an initial off-boresight angle of zero degrees.

The maximum aerodynamic range is then used as found in section 5.2.2 for each altitude and employed guidance law. Since no PIP-mode is required for these results, R_{aero} for the RL agent based guidance law is equal for each PIP-mode. The maximum turn and run range is similarly found only from 5.3.2 however R_{tr} does vary between PIP-modes.

The maximum no-escape range is found from section 5.4.2 where two definitions are now used for $R_{max,2}$. These are the ranges at which 75% and 90% hit probability are achieved and are denoted here as $R_{max,75\%}$ and $R_{max,90\%}$ respectively. They are found by finding the mean range of the contour outlining each zone respectively. The NEZ zone is then defined as the zone between $R_{min,2}$, which is assumed to be 10 km, and the minimum of R_{tr} and $R_{max,90\%}$. For optimal control, it is assumed that R_{tr} is limiting the NEZ and not $R_{max,90\%}$. This is true due to the optimal control having access to all future target states in this analysis enabling it to hit any target within R_{tr}.

These results are then presented in figures 5.36 - 5.41 where both the absolute results are given for altitudes of [3,6,9] km and the matching results normalized using R_{aero} obtained using optimal control. To be consistent in the color scheme used throughout this report whilst representing both PIP-modes in a single figure, the left side of the cone is used for PIP-mode 1 whilst the right is used for PIP-mode 2. Furthermore, the NEZs are represented by the shaded areas. The presented figures effectively summarize a large part of the results obtained in previ-

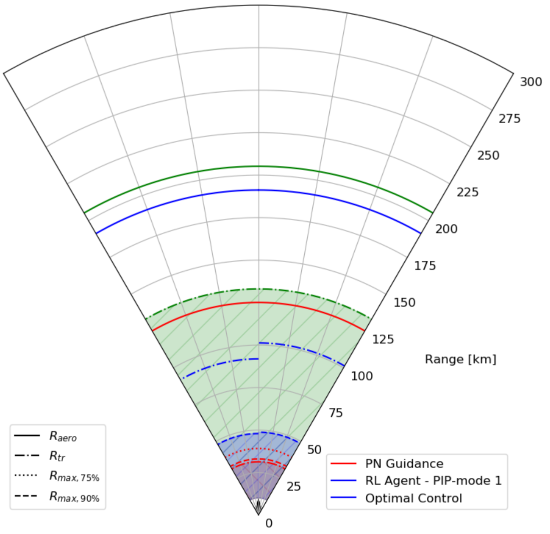


Figure 5.36: DLZ for PN guidance with lofting, RL agent based guidance using PIP-mode 1 (left) and 2 (right), and optimal control at an altitude of 3 km

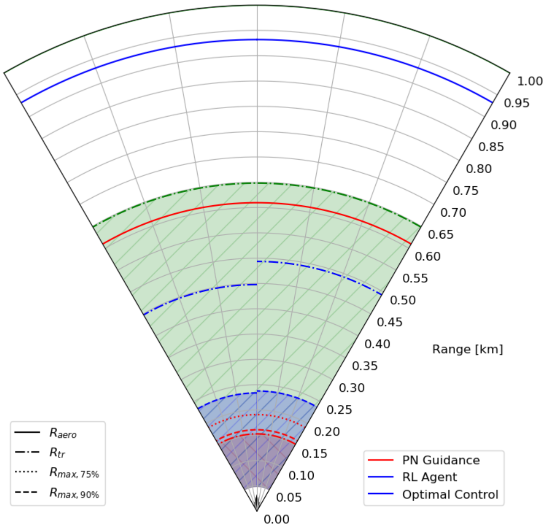


Figure 5.37: DLZ normalized using theoretical maximum range for PN guidance with lofting, RL agent based guidance using PIP-mode 1 (left) and 2 (right), and optimal control at an altitude of 3 km

ous sections and present the performance obtained using the implemented methodology compared to traditional and optimal solutions.

As can be seen from the absolute results, ranges generally increase with altitude for each defined methodology. An interesting observation can be made where the NEZ is relatively small for both PN and RL agent based guidance at 3 km altitude when compared to optimal control. This is deemed to be an effect of the missile maneuvering due to target maneuvers and maneuvers being more costly due to higher atmospheric densities at lower altitudes. Optimal control does not deal with this uncertainty and does therefore not suffer from this.

When comparing PIP-modes used for RL agent based guidance to each other, it is consistently found that PIP-mode 2 outperforms PIP-mode 1. Comparing the RL agent based guidance using PIP-mode 2 to PN with lofting it is clear that RL agent based guidance is superior. Where PN guidance with lofting has a NEZ covering 15% at 3 km altitude to 35% at 9 km altitude of the theoretical maximum range. The RL agent based guidance with PIP-mode 2 achieves a NEZ between 25% at 3 km altitude to 55% at 9 km altitude of this maximum range. Comparing RL agent based guidance to optimal control it is outperformed, however part of the difference in performance with respect to optimal control lies in the fact that optimal control faces no uncertainty regarding future target states.

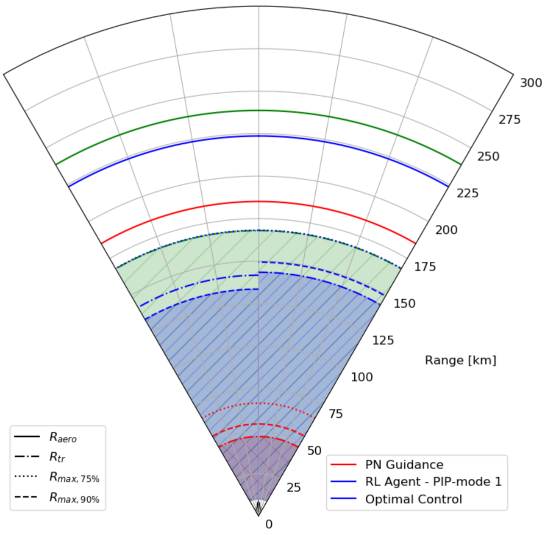


Figure 5.38: DLZ for PN guidance with lofting, RL agent based guidance using PIP-mode 1 (left) and 2 (right), and optimal control at an altitude of 6 km

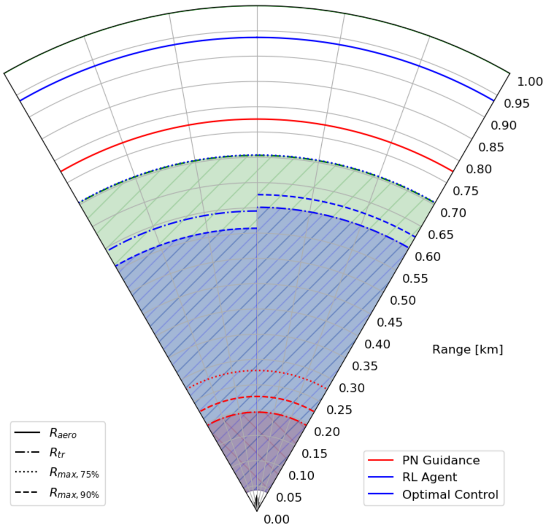


Figure 5.39: DLZ normalized using theoretical maximum range for PN guidance with lofting, RL agent based guidance using PIP-mode 1 (left) and 2 (right), and optimal control at an altitude of 6 km

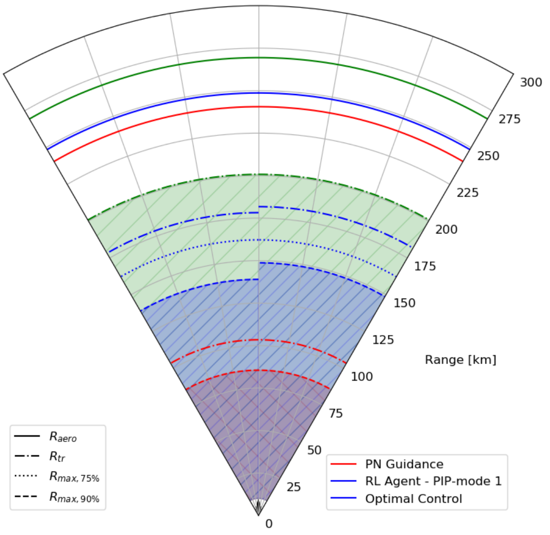


Figure 5.40: DLZ for PN guidance with lofting, RL agent based guidance using PIP-mode 1 (left) and 2 (right), and optimal control at an altitude of 9 km

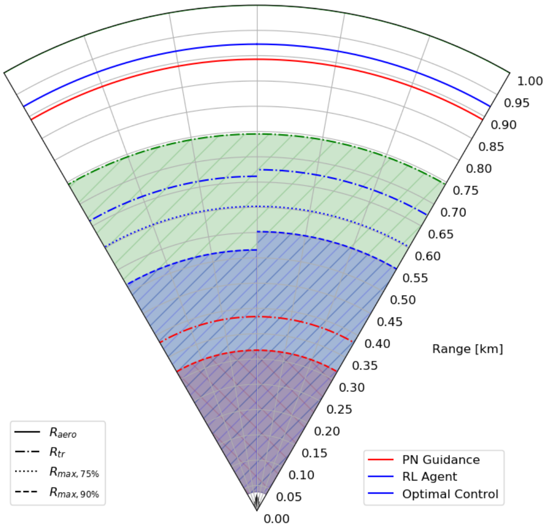


Figure 5.41: DLZ normalized using theoretical maximum range for PN guidance with lofting, RL agent based guidance using PIP-mode 1 (left) and 2 (right), and optimal control at an altitude of 9 km

6

Conclusion & Recommendations

The main research question has been formulated as:

What improvements to the performance of air-to-air missile guidance systems can be made based on the identification of optimal trajectories?

To answer this question, first past research is set out focusing on methods of missile guidance and optimal control. An experimental environment has been developed in which traditional guidance, optimal control and a novel guidance law is implemented. This novel guidance law is developed with the main research question in mind where it aims to emulate behaviour observed and approximate the performance obtained throught optimal control.

6.1. Conclusions

A methodology based on RL is employed to develop a novel guidance law and is proven to be a viable concept. The RL methodology is based on the DDPG algorithm and extended using learning from demonstrations and HER. The RL training methodology results in an agent which determines a control command based on only the current states of the system and is based on a NN architecture. Due to these reasons the guidance law can be implemented in a real-time feedback manner whereas optimal control cannot due to it requiring future target states and relatively large computational times. The developed experimental environment in which an engagement can be simulated featuring a missile and a target is used to both train the RL agent and simulate engagements using several guidance methods. Trajectory optimization is used to both provide demonstrations to the RL training algorithm and establish the theoretical optimal performance achievable.

The RL agent is able to close the distance to a target by itself however it is not able to sufficiently minimize the miss distance. The RL agent is therefore combined with a terminal guidance law namely PN. Furthermore, the RL agent is trained using stationary targets and when used against moving target it is combined with a PIP algorithm. The combination of the RL agent and PIP algorithm efficiently closing the majority off the distance to the target and then switching to the terminal guidance law providing in close guidance leads to an effective guidance law.

The resulting RL agent based guidance law is compared to trajectories obtained using optimal control engaging a stationary target and a target performing a turn and run maneuver. It is concluded that the RL based guidance law achieves approximately 90% of performance obtained from optimal control in terms of maximum range when engaging a stationary target. In terms of time of flight against a stationary target in the majority of the evaluated domain the RL agent based guidance law achieves a time of flight within 5% of that obtained using optimal control. In limited parts of the evaluated domain the RL agent based guidance law did not hit or had a significantly longer time of flight then the optimal control solution. When engaging a target performing a turn and run maneuver, the maximum range achieved is between 75% and 90% of that obtained using optimal control.

Comparing the RL based guidance law to a traditional guidance law, in this case PN with lofting, it is concluded that the developed RL agent based guidance law outperforms it. This in terms of both range and time of flight against a stationary target over the majority of the evaluated domain. The RL agent furthermore greatly outperforms the PN guidance law in terms of range when engaging a target performing a turn and run maneuver. This is especially true at lower altitudes. When engaging a random maneuvering target it is concluded that the RL agent based guidance law outperforms the traditional guidance law significantly and increases effective engagement range significantly. Two PIP modes are furthermore compared and it is concluded that the mode assuming the target is flying directly away from the missile outperforms the mode assuming its actual flight direction.

A noted disadvantage of the presented methodology is the uncertainty of an appropriate control solution for an encountered state due to potential lack of training coverage. Such states or rather domains of states are also encountered during evaluation of the methodology. In such domains the RL agent provides a non-logical control command leading to lower performance when compared to PN with lofting. A fail safe can and is implemented to limit the effect of such events, however such inappropriate control solutions provided by the RL agent still lead to diminished performance in these domains when compared to PN with lofting. Such domains with insufficient training coverage could be minimized by further developing and extending the training procedure.

The RL methodology is shown to be feasible in two different environments incorporating different levels of complexity. This shows a major advantage of the used RL methodology, it is independent on the environment in which it is used. The experimental environment in which the results are in this case established dictate the specific performance increase observed. The method is however independent of the model and therefore should still be able to approximate optimal performance if transferred to a different experimental environment.

Thus coupling this back to the main research question, it can be concluded that the developed novel guidance law based on identified optimal trajectories is able to improve performance over the implemented PN guidance law with lofting over the majority of the evaluated domain. Specifically in terms of maximum range, time of flight, and effective engagement range. When compared to optimal control, between 75% and 90% of its range performance is obtained using the RL agent based guidance depending on the scenario where the optimal control methodology is provided the full target trajectory giving it a distinct advantage.

6.2. RECOMMENDATIONS

Based on the work presented several recommendations can be made. First of all, the results show lack of training coverage leading to inappropriate control commands by the RL agent in such domains. Further increasing quality of training and developing a method to evaluate training quality would be logical next steps for the presented methodology.

Furthermore, a distinct choice is made in training the RL agent using stationary targets. The guidance law then uses a PIP algorithm to use the RL agent against moving targets. Incorporating moving targets in the training routine could potentially improve performance. By doing so the PIP algorithm would be integrated in the agent. Another option would be to not integrate it in training but to improve the PIP algorithm itself. The currently used PIP methods perform well but a simple example of an improvement would be to incorporate the missiles velocity profile to more accurately approximate the PIP.

Another compromise had to be made by incorporating a terminal guidance law. The terminal guidance law performs well however it would be beneficial to avoid transitioning to a different guidance scheme due to the sudden and often relatively high control command after switching leading to unnecessary kinetic energy loss. A proposed method would be to include additional observations which scale with range to target enabling the neural network to attain higher accuracy in different phases (e.g. $1/R$ where R is range to target).

Regarding the results, it is shown that for many cases the RL agent based guidance law outperforms PN with lofting. However no statistical basis was tied to this. It seems evident that the developed guidance law outperforms the traditional guidance law however this conclusion could be reinforced using a statistical analysis. A second note on this would be that the PN guidance law incorporating lofting could be improved since the employed gain and lofting scheme show decent performance but are not optimized.

Lastly, the presented methodology is at this point only a proof of concept and works in a simplified experimental environment. In a realistic scenario factors such as limited information, disturbances, and interactions with other missile subsystems would test robustness of the guidance law. Furthermore, the physical modelling of the missile is kept simple on purpose for this research, however to further prove the feasibility of the methodology, it should be improved.

Bibliography

[1] A. T. Slawson, *Air Power's First Among Equals: Why Air Superiority Still Matters*, Tech. Rep. (Faculty of the Joint Advanced Warfighting School, 2008).

[2] J. Stillion, *Trends in Air-to-Air Combat - Implications for Future Air Superiority*, Tech. Rep. (Center for Strategic and Budgetary Assessments, 2015).

[3] L. Deng and Z. Shen, *Trajectory optimization of aerodynamically controlled missiles using pseudospectral method,* in *2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE)* (2017) pp. 178–182.

[4] P. Kee, L. Dong, and C. Siong (American Institute of Aeronautics and Astronautics, 1998) Chap. Near optimal midcourse guidance law for flight vehicle, 0.

[5] R. Yanushevsky, *Modern Missile Guidance* (CRC Press, 2008).

[6] Y. S. Alqudsi and E.-B. G. M, *A qualitative comparison between the proportional navigation and differential geometry guidance algorithms,* INCAS (2018).

[7] A. B. Jr and Y. Ho, *Applied Optimal Control* (Taylor & Francis, 1975) revised Printing.

[8] R. Dai (American Institute of Aeronautics and Astronautics, 2007) Chap. Three-Dimensional Minimum-Time Interception Trajectory Planning Using Nonlinear Programming and Collocation.

[9] M. Manickavasagam, A. Sarkar, and V. Vaithiyanathan, *A singular perturbation based midcourse guidance law for realistic air-to-air engagement,* Defence Science Journal; Vol 67, No 1 (2016).

[10] E. Song and M. Tahk, *Real-time neural-network midcourse guidance,* Control Engineering Practice **9**, 1145 (2001).

[11] D. Han, S. N. Balakrishnan, and E. J. Ohlmeyer, *Midcourse guidance law with neural networks,* Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering **219**, 131 (2005).

[12] A. Heydari and S. Balakrishnan, *Adaptive critic based solution to an orbital rendezvous problem,* Journal of Guidance Control and Dynamics **37** (2014), 10.2514/1.60553.

[13] A. Rodriguez-Ramos, C. Sampedro, H. Bavle, P. de la Puente, and P. Campoy, *A deep reinforcement learning strategy for uav autonomous landing on a moving platform,* Journal of Intelligent & Robotic Systems (2018), 10.1007/s10846-018-0891-8.

[14] M. van Hoorn, *Near Optimal Missile Guidance*, Tech. Rep. (TU Delft, 2018).

[15] M. M. Systems, *Meteor - beyond visual range air-to-air missile (bvraam),* `https://www.mbda-systems.com/product/meteor/` (2018).

[16] B. Birkmire, *Weapon Engagement Zone Maximum Launch Range Approximation using a Multilayer Perceptron*, Master's thesis, B.S., Wright State University (2008).

[17] R. Brochu and R. Lestage, *Three-Degree-of-Freedom (DOF) Missile Trajectory Simulation Model and Comparative Study with a High Fidelity 6DOF Model*, Tech. Rep. (DRDC Valcartier, 2003).

[18] M. Manickavasagam, A. Sarkar, and V. Vaithiyanathan, *A singular perturbation based midcourse guidance law for realistic air-to-air engagement,* Defence Science Journal **67** (2017).

[19] N. Harl, S. Balakrishnan, and C. Phillips, *Sliding mode integrated missile guidance and control,* in *AIAA Guidance, Navigation, and Control Conference, Toronto, Ontario Canada* (2010).

[20] L. Pontryagin, V. Boltyanksii, R. Gamkrelidze, and E. Mishchenko, *Mathematical Theory of Optimal Processes* (Interscience, 1962).

[21] M. Patterson and A. Rao, *Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming,* ACM Trans. Math. Softw. **41**, 1:1 (2014).

[22] A. Rao, *A survey of numerical methods for optimal control,* Preprint on webpage at `http://www.anilvrao.com/Publications/ConferencePublications/trajectorySurveyAAS.pdf`.

[23] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, *Continuous control with deep reinforcement learning,* CoRR **abs/1509.02971** (2015), arXiv:1509.02971 .

[24] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, *Deterministic policy gradient algorithms,* in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14 (JMLR.org, 2014) pp. I–387–I–395.

[25] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. (Pearson Education, 2003).

[26] Y. Li, *Deep reinforcement learning,* CoRR **abs/1810.06339** (2018), arXiv:1810.06339 .

[27] C. Burch, *A survey of machine learning first edition,* (2001).

[28] R. S. Sutton and A. G. Barto, *Reinforcement learning - an introduction*, Adaptive computation and machine learning (MIT Press, 1998).

[29] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, *Continuous control with deep reinforcement learning,* CoRR **abs/1509.02971** (2015), arXiv:1509.02971 .

[30] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, *Deterministic policy gradient algorithms,* in *Proceedings of the 31st International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 32, edited by E. P. Xing and T. Jebara (PMLR, Bejing, China, 2014) pp. 387–395.

[31] R. S. Sutton and A. G. Barto, *Reinforcement learning - an introduction*, Adaptive computation and machine learning (MIT Press, 1998).

[32] V. Konda, *Actor-critic Algorithms*, Ph.D. thesis, Cambridge, MA, USA (2002), aAI0804543.

[33] G. Halfacree, *Benchmarking the raspberry pi 3 b+*, `https://medium.com/@ghalfacree/benchmarking-the-raspberry-pi-3-b-plus-44122cf3d806` (2018).

[34] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization,* CoRR **abs/1412.6980** (2014), arXiv:1412.6980 .

[35] S. Ioffe and C. Szegedy, *Batch normalization: Accelerating deep network training by reducing internal covariate shift,* CoRR **abs/1502.03167** (2015), arXiv:1502.03167 .

[36] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, *Hindsight experience replay,* CoRR **abs/1707.01495** (2017), arXiv:1707.01495 .

[37] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, *Overcoming exploration in reinforcement learning with demonstrations,* CoRR **abs/1709.10089** (2017), arXiv:1709.10089 .

[38] G. E. Uhlenbeck and L. S. Ornstein, *On the theory of the brownian motion,* Phys. Rev. **36,** 823 (1930).

A

Additional Figures

A.1. 3-DoF - Range Envelope with Stationary Target

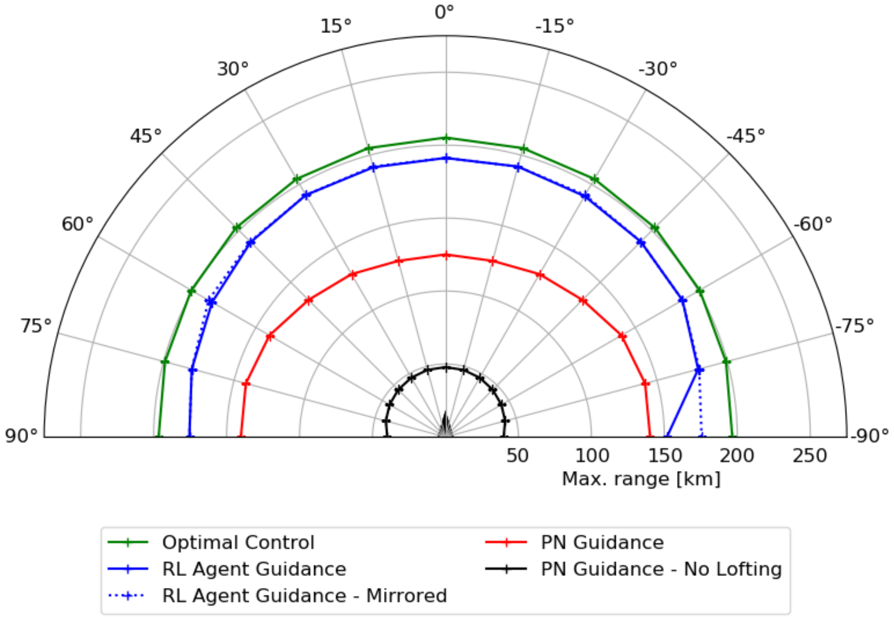


Figure A.1: Maximum ranges plotted versus varying initial off-boresight angles at an altitude of 3 km for different guidance methods

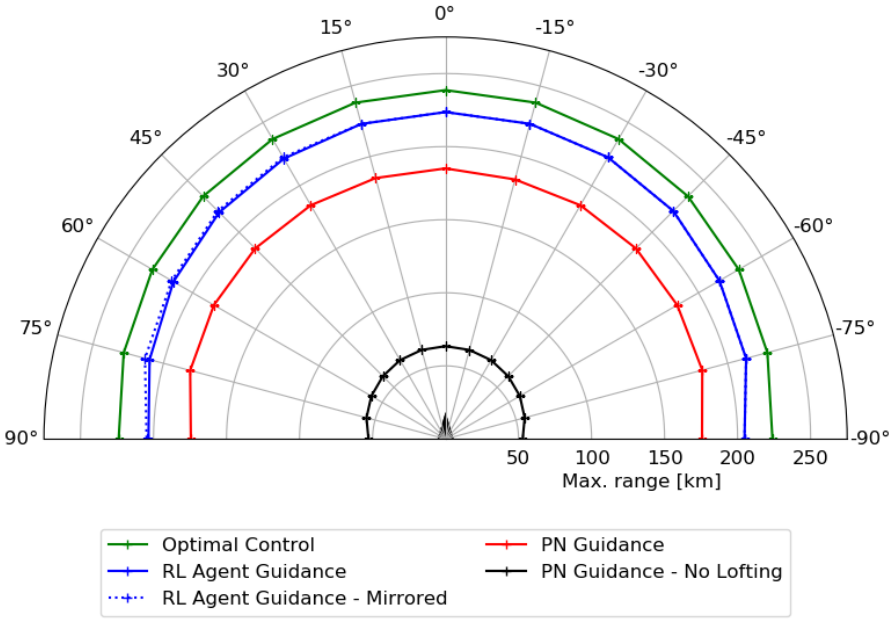


Figure A.2: Maximum ranges plotted versus varying initial off-boresight angles at an altitude of 6 km for different guidance methods

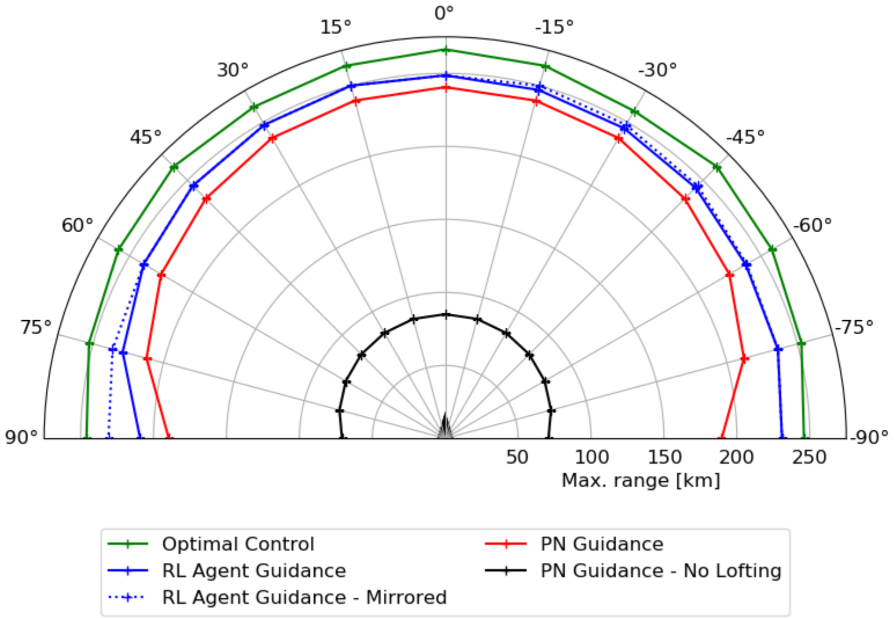


Figure A.3: Maximum ranges plotted versus varying initial off-boresight angles at an altitude of 9 km for different guidance methods

A.2. 3-DoF - Time of Flight with Stationary Target

A.2.1. Absolute Time of Flight - Proportional Navigation

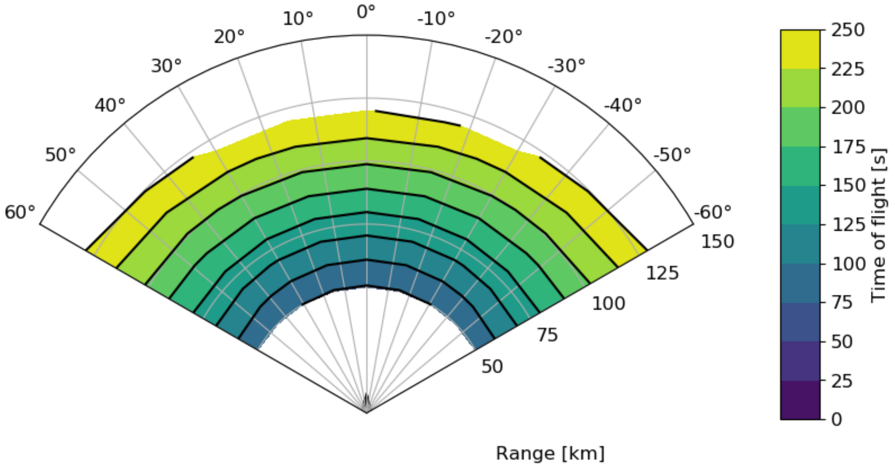


Figure A.4: Time of flight achieved using PN guidance with lofting at 3 km altitude

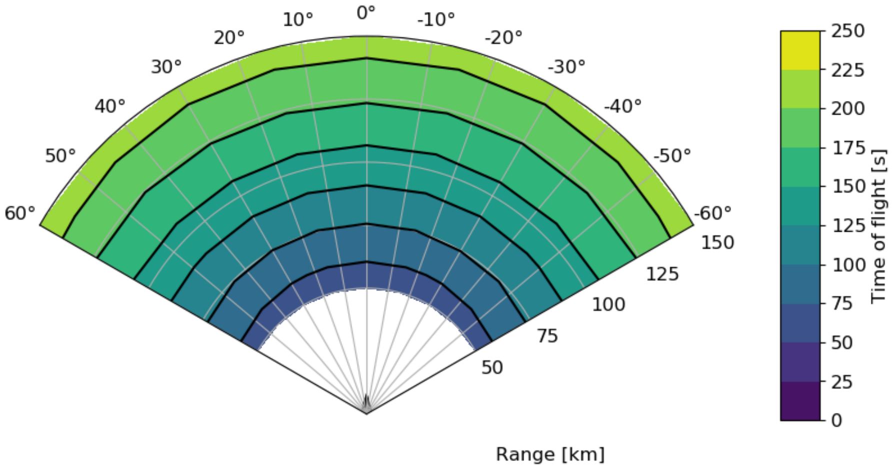


Figure A.5: Time of flight achieved using PN guidance with lofting at 6 km altitude

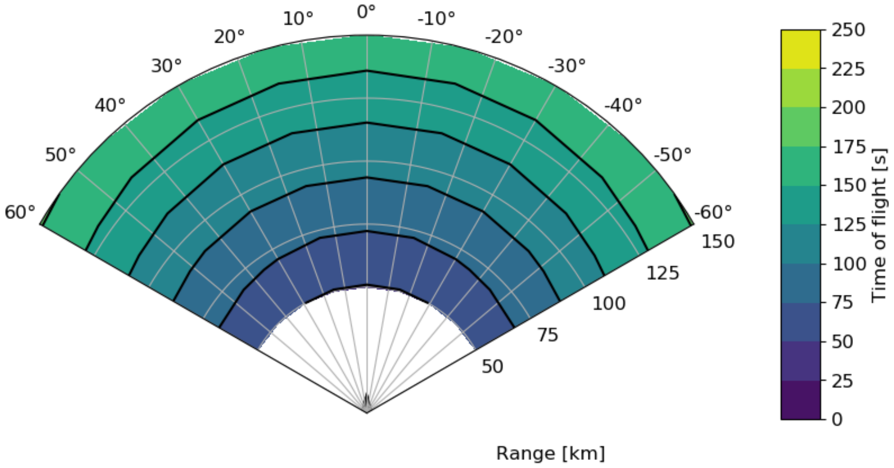


Figure A.6: Time of flight achieved using PN guidance with lofting at 9 km altitude

A.2.2. ABSOLUTE TIME OF FLIGHT - RL AGENT

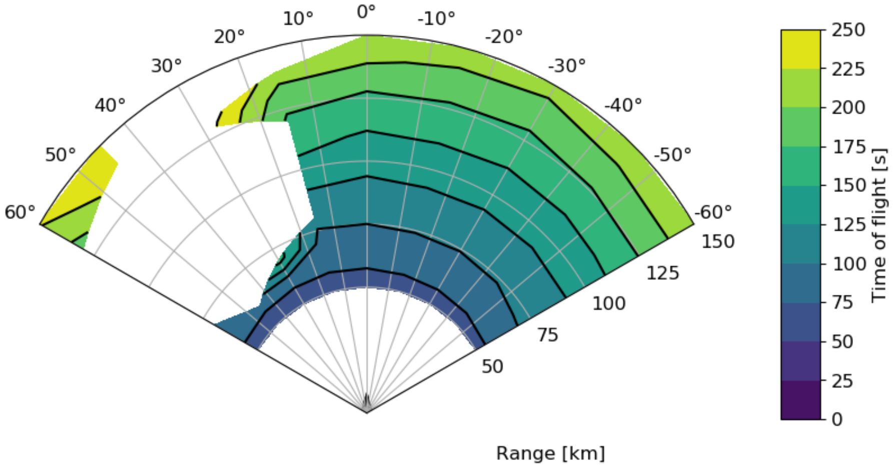


Figure A.7: Time of flight achieved using RL agent based guidance at 3 km altitude

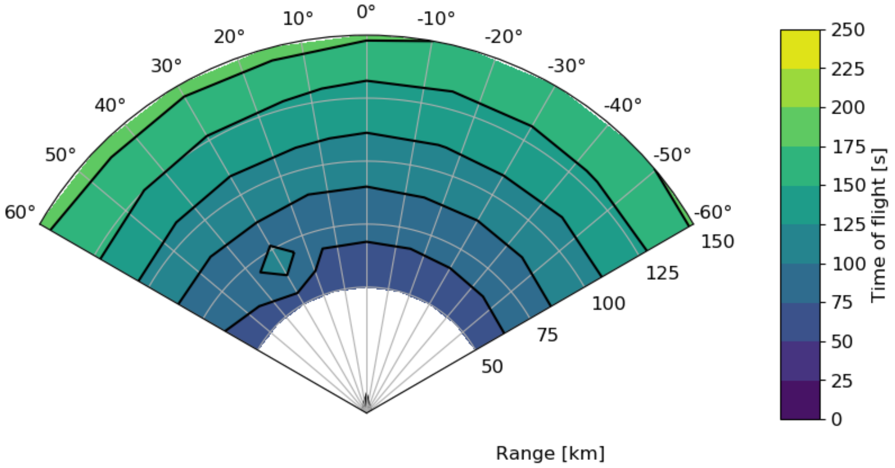


Figure A.8: Time of flight achieved using RL agent based guidance at 6 km altitude

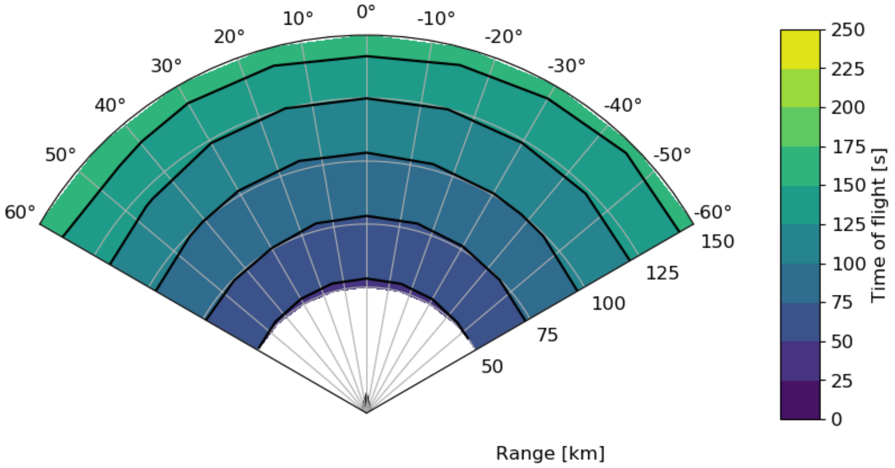


Figure A.9: Time of flight achieved using RL agent based guidance at 9 km altitude

A.2.3. ABSOLUTE TIME OF FLIGHT - OPTIMAL CONTROL

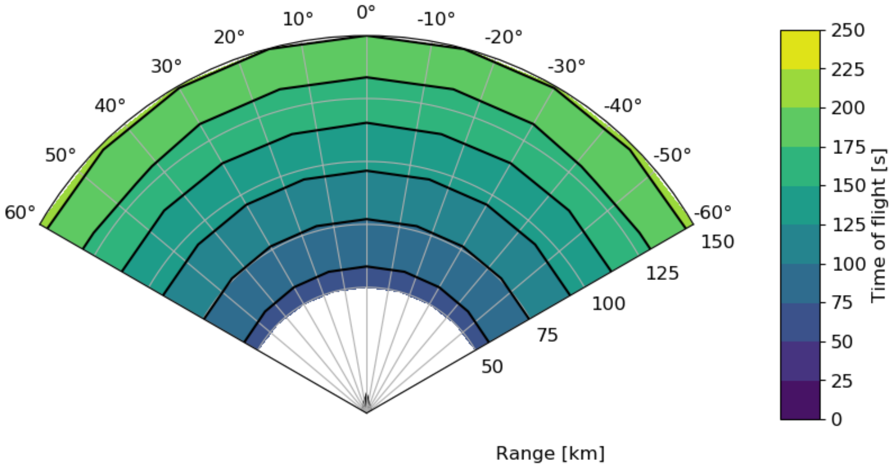


Figure A.10: Time of flight achieved using optimal control at 3 km altitude

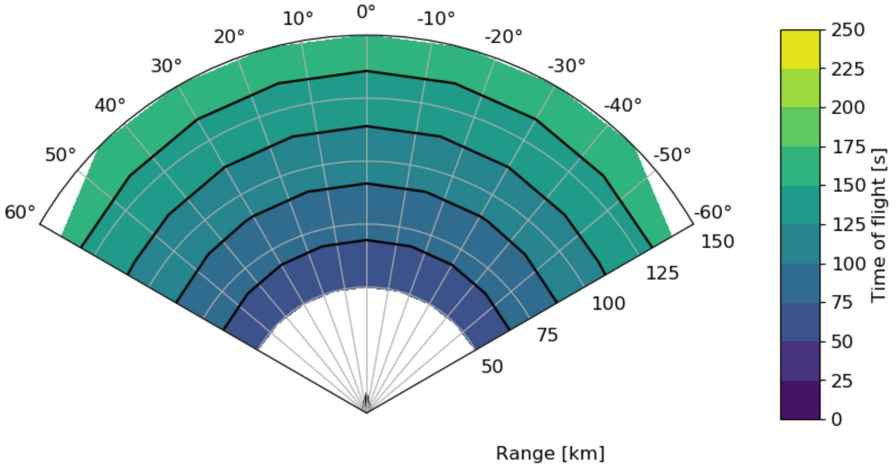


Figure A.11: Time of flight achieved using optimal control at 6 km altitude

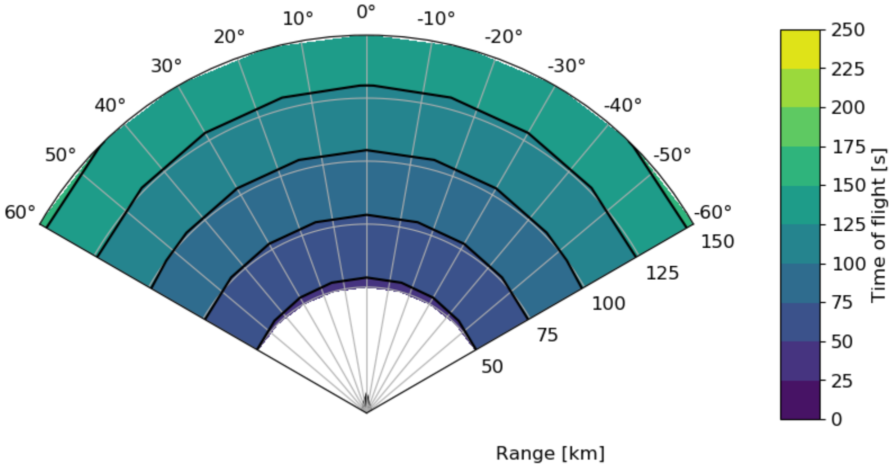


Figure A.12: Time of flight achieved using optimal control at 9 km altitude

A.2.4. Time of Flight relative to Proportional Navigation

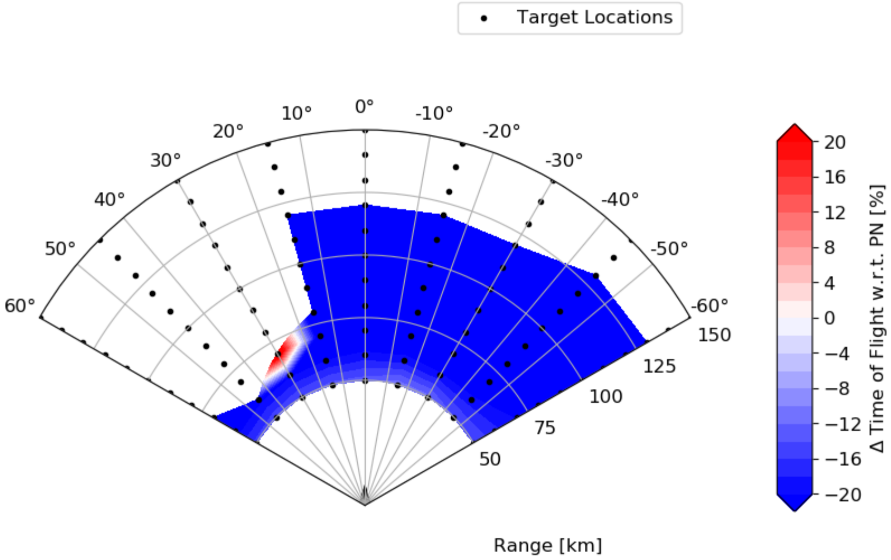


Figure A.13: Time of flight achieved using RL agent compared to PN guidance with lofting solutions at 3 km altitude

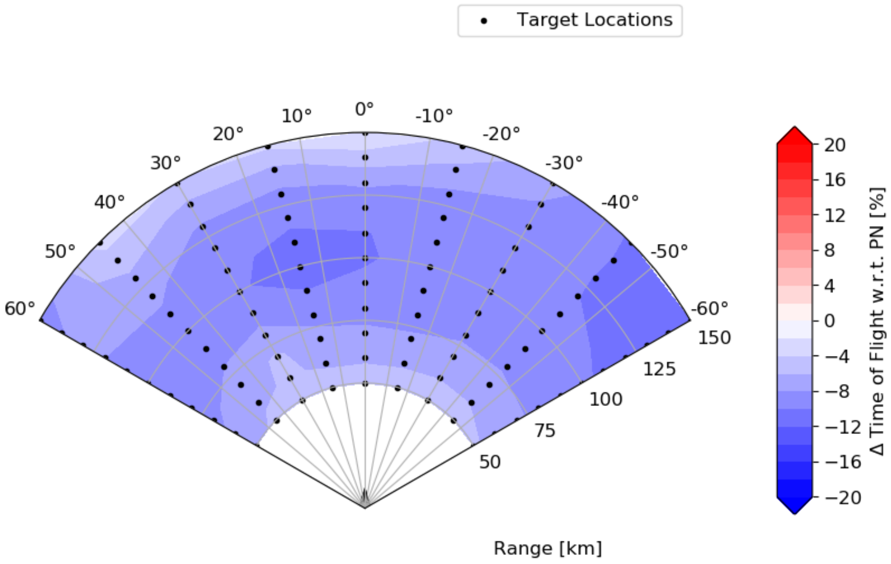


Figure A.14: Time of flight achieved using RL agent compared to PN guidance with lofting solutions at 9 km altitude

A.2.5. Time of Flight relative to Optimal Control

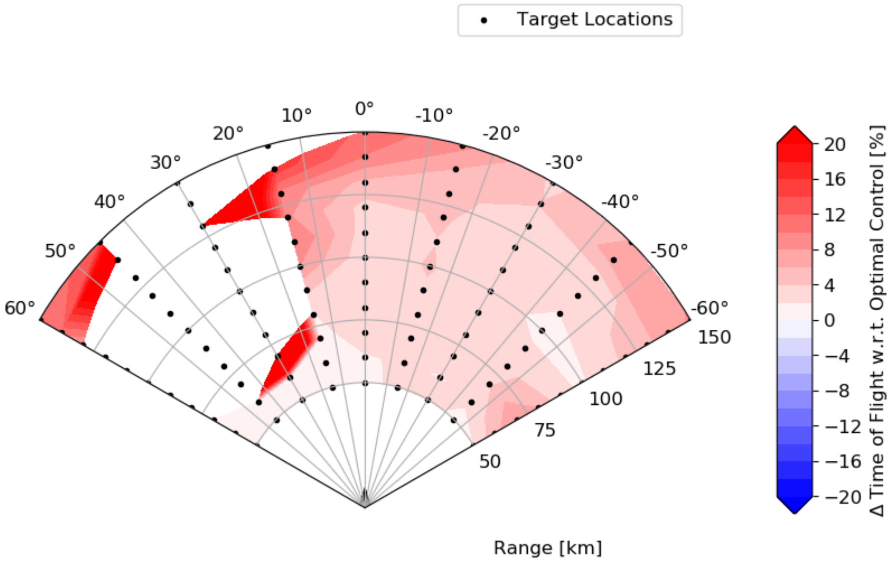


Figure A.15: Time of flight achieved using RL agent compared to optimal control solutions at 6 km altitude

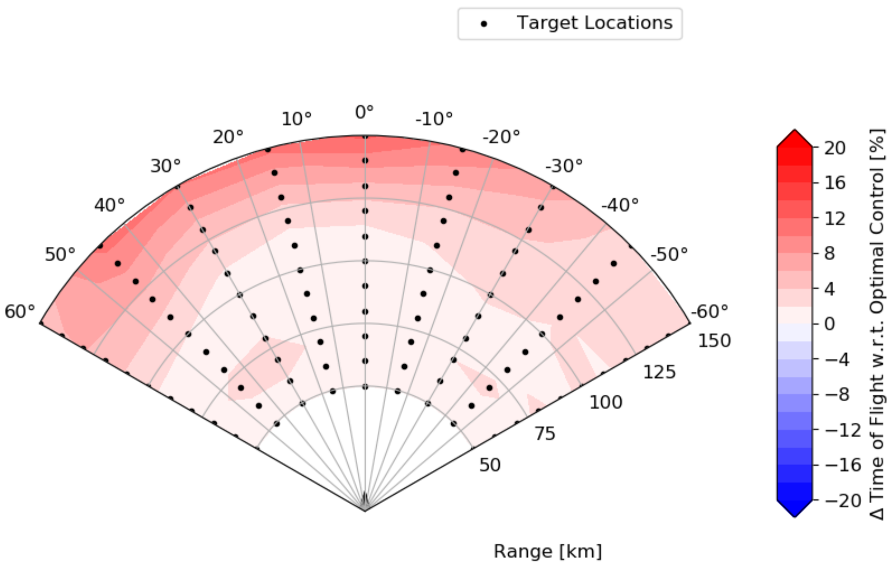


Figure A.16: Time of flight achieved using RL agent compared to optimal control solutions at 6 km altitude

A.3. 3-DoF - Case Study with Turn and Run Target

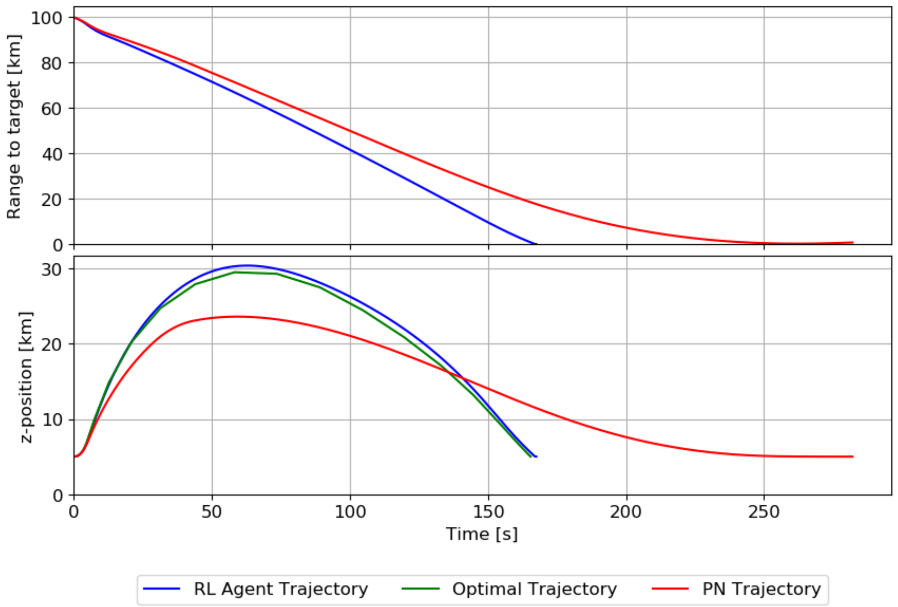


Figure A.17: Range and z-position plotted versus time for missile trajectories obtained using optimal control, PN with lofting and RL agent based guidance combined with PIP-mode 1

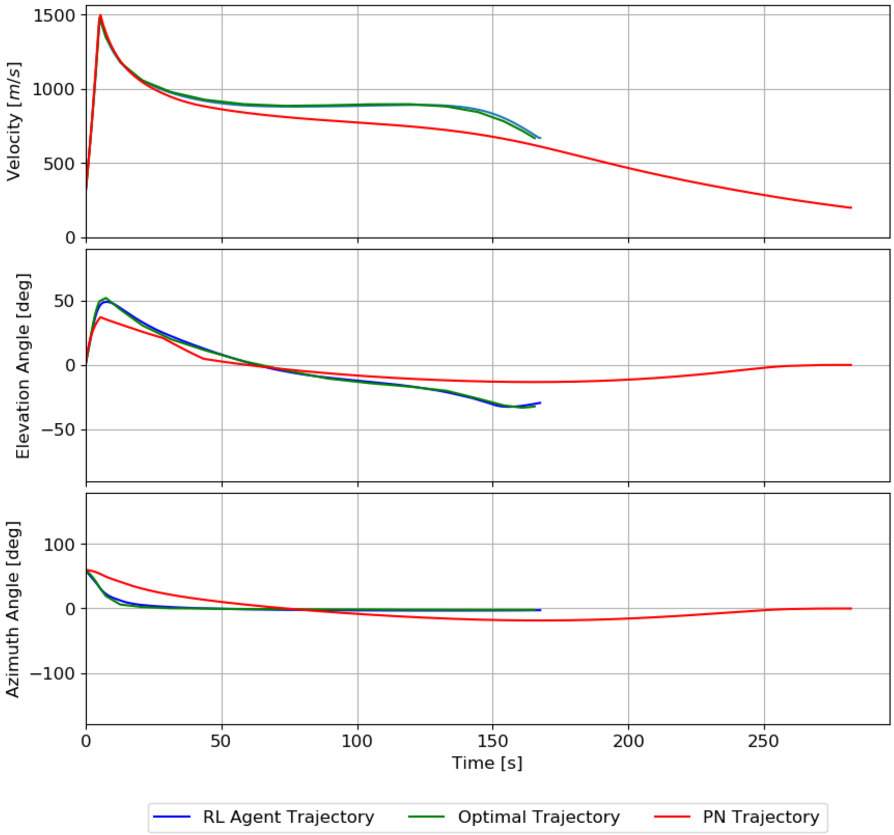


Figure A.18: Velocity, elevation angle and azimuth angle plotted versus time for missile trajectories obtained using optimal control, PN with lofting and RL agent based guidance combined with PIP-mode 1

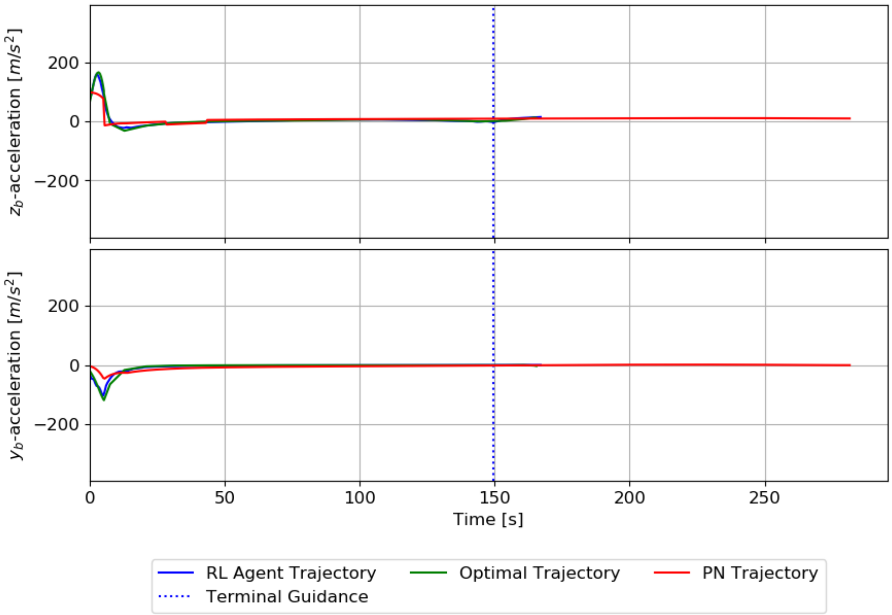


Figure A.19: Control commands in missile body z and y directions plotted versus time for missile trajectories obtained using optimal control, PN with lofting and RL agent based guidance combined with PIP-mode 1

A.4. 3-DoF - Range Envelope with Turn and Run Target

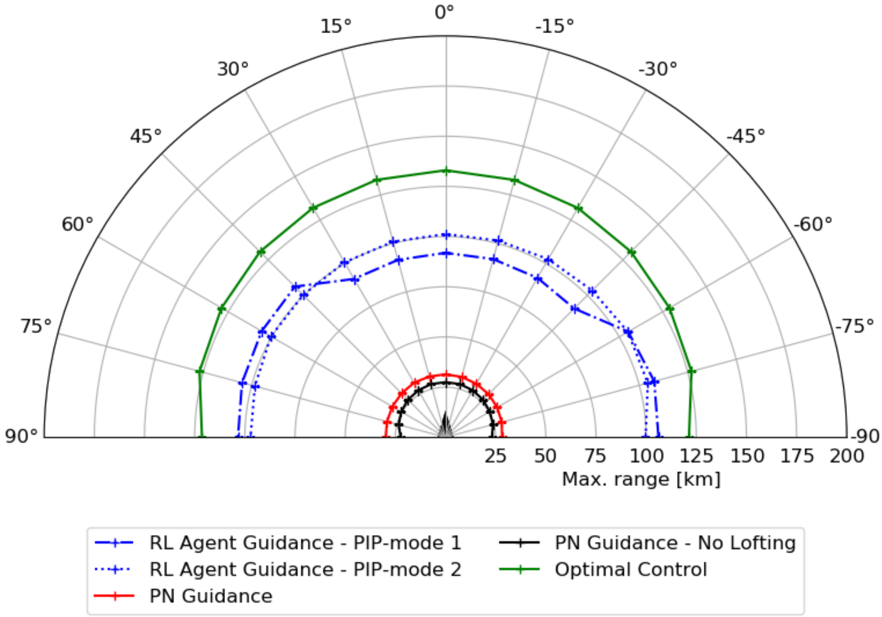


Figure A.20: Maximum launch ranges plotted versus varying initial off-boresight angles at an altitude of 3 km for different guidance methods versus a target performing a turn and run maneuver

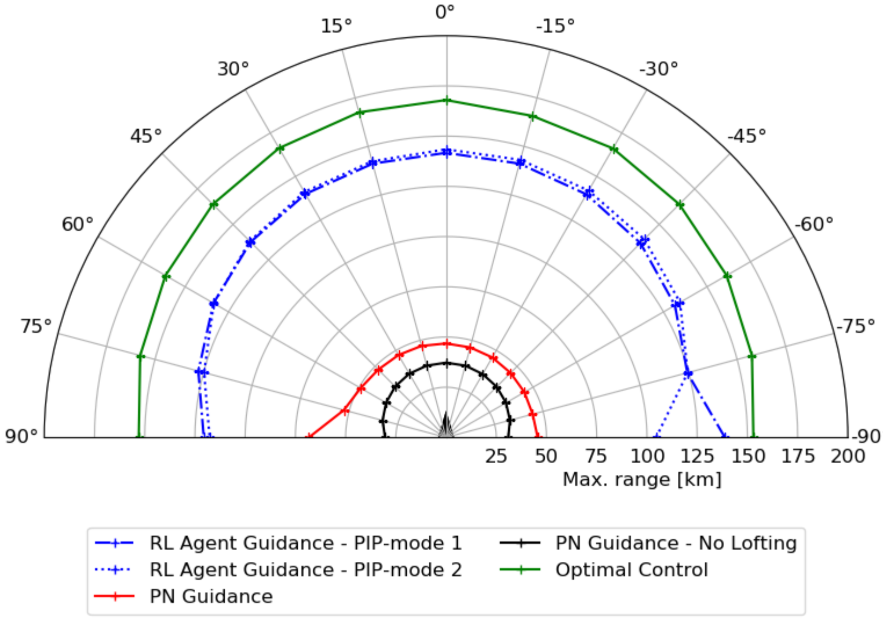


Figure A.21: Maximum launch ranges plotted versus varying initial off-boresight angles at an altitude of 6 km for different guidance methods versus a target performing a turn and run maneuver

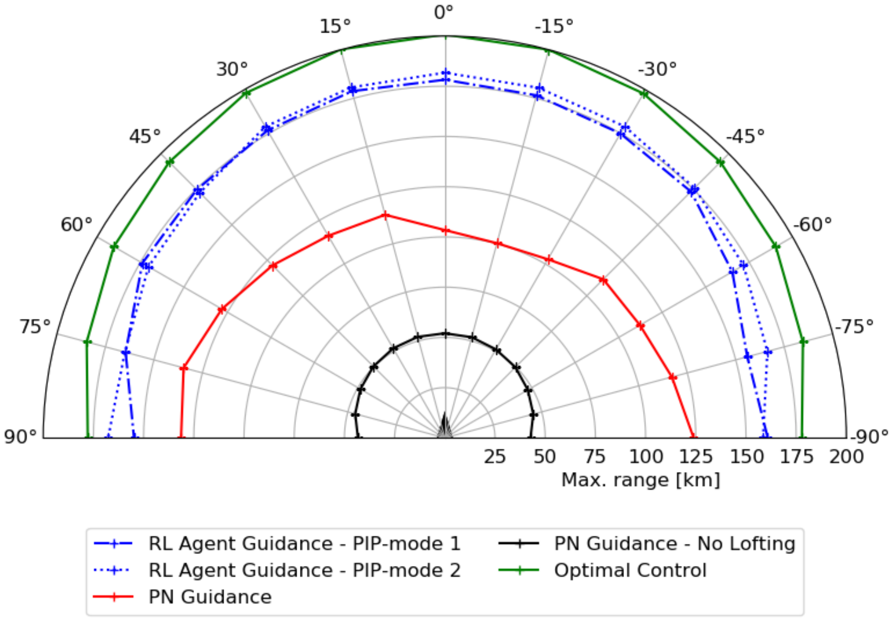


Figure A.22: Maximum launch ranges plotted versus varying initial off-boresight angles at an altitude of 9 km for different guidance methods versus a target performing a turn and run maneuver

A.5. 3-DoF - Case Study with Random Maneuvering Target

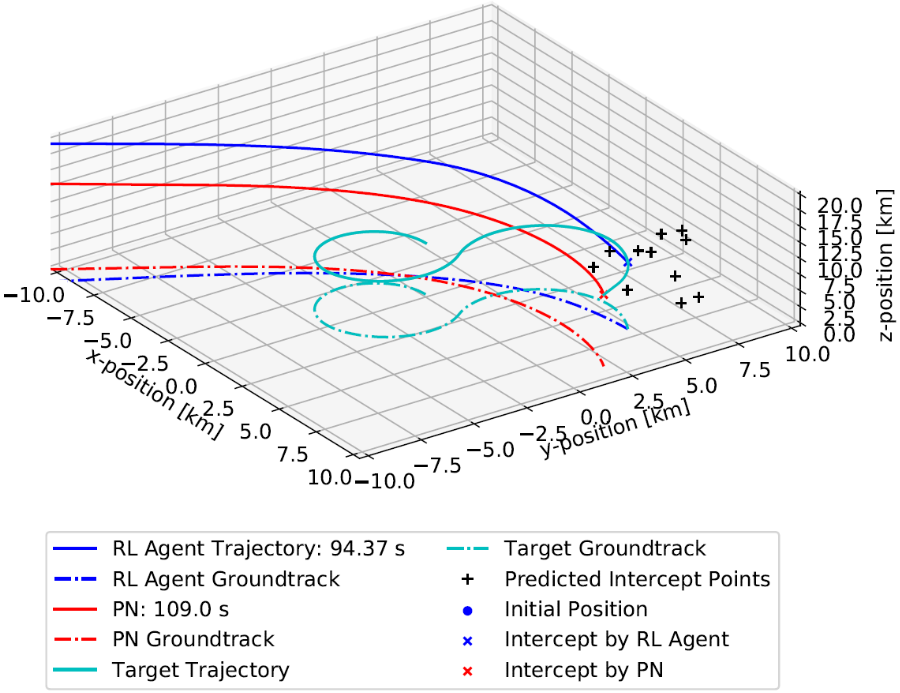


Figure A.23: Resulting terminal phase trajectories plotted for missile intercepting random maneuvering target using PN and RL agent based guidance with PIP-mode 2

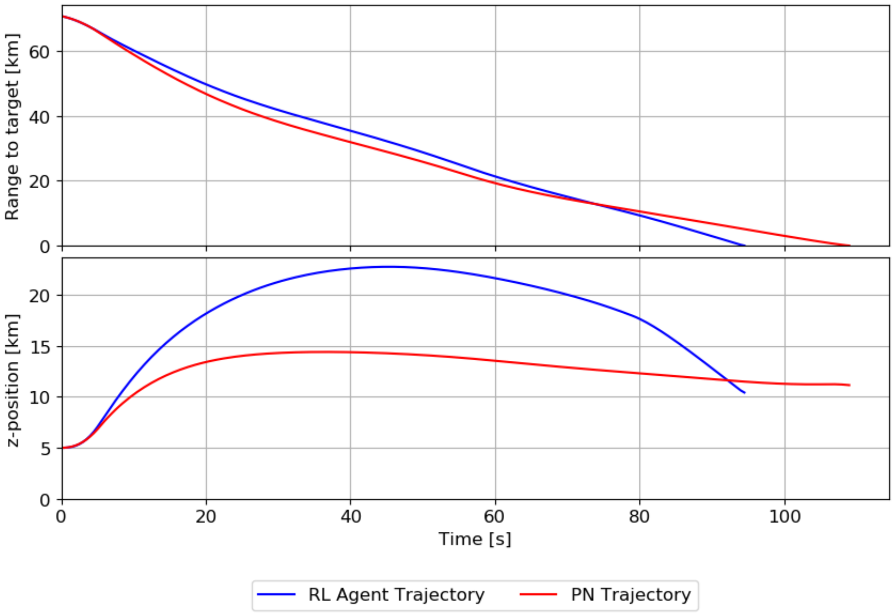


Figure A.24: Range and z-position plotted versus time for missile trajectories obtained using PN and RL agent based guidance with PIP-mode 2

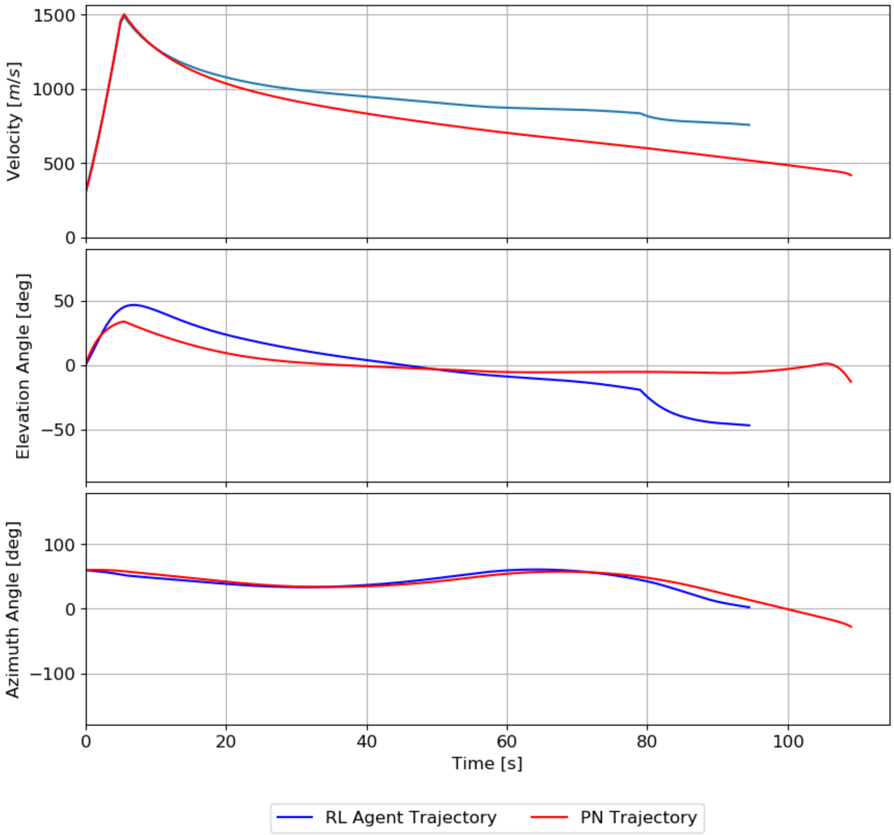


Figure A.25: Velocity, elevation and azimuth angles plotted versus time for missile trajectories obtained using PN and RL agent based guidance with PIP-mode 2

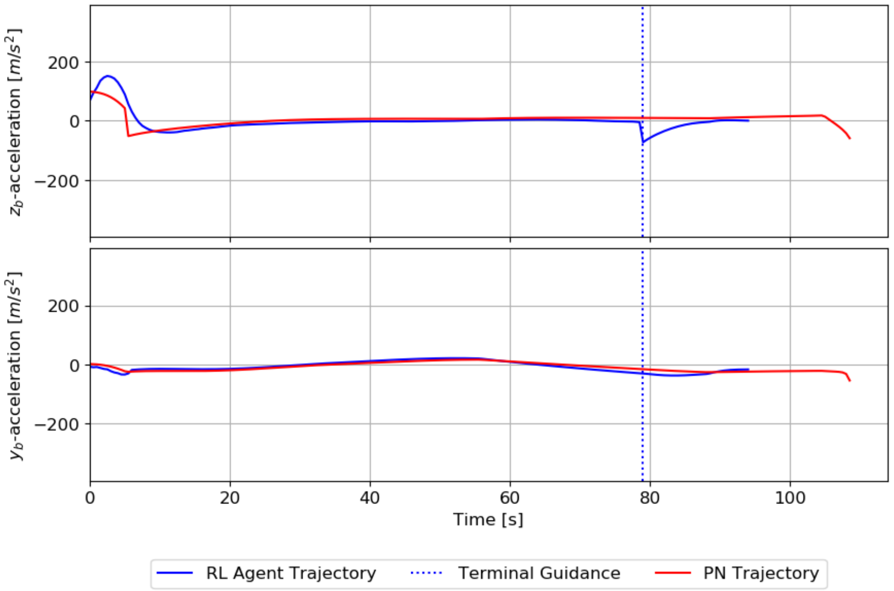


Figure A.26: Control commands in missile body z and y directions plotted versus time for missile trajectories obtained using PN with lofting and RL agent based guidance with PIP-mode 2

A.6. 3-DOF - HIT PROBABILITY ENVELOPES

A.6.1. PROPORTIONAL NAVIGATION

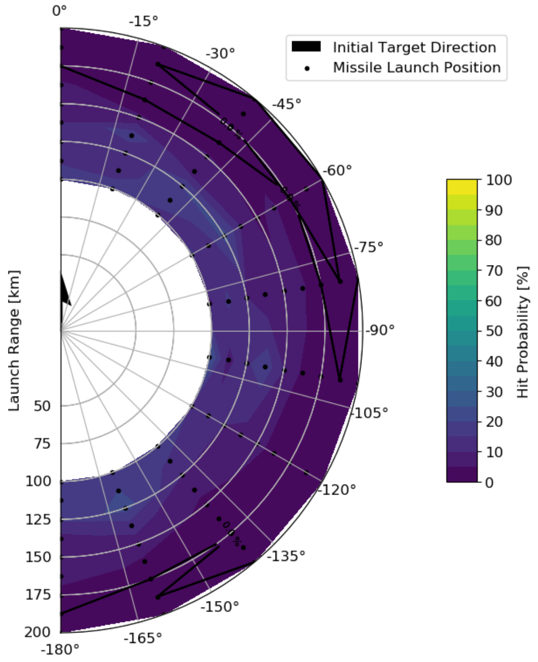


Figure A.27: Hit probability envelopes for PN guidance with lofting at an altitude of 3 km

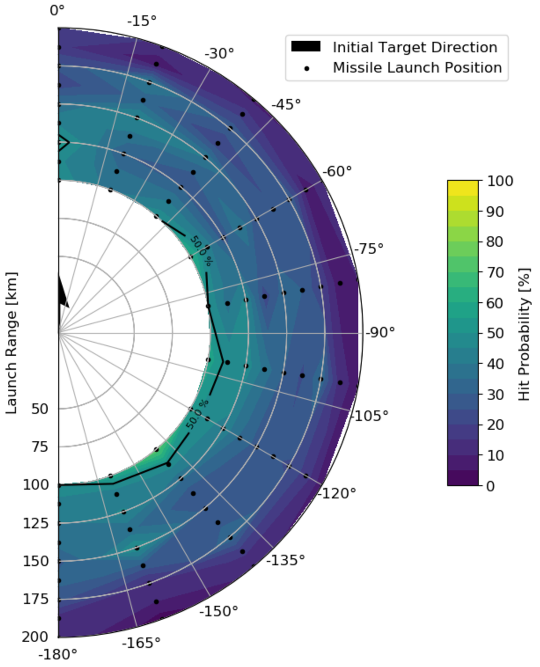


Figure A.28: Hit probability envelopes for PN guidance with lofting at an altitude of 6 km

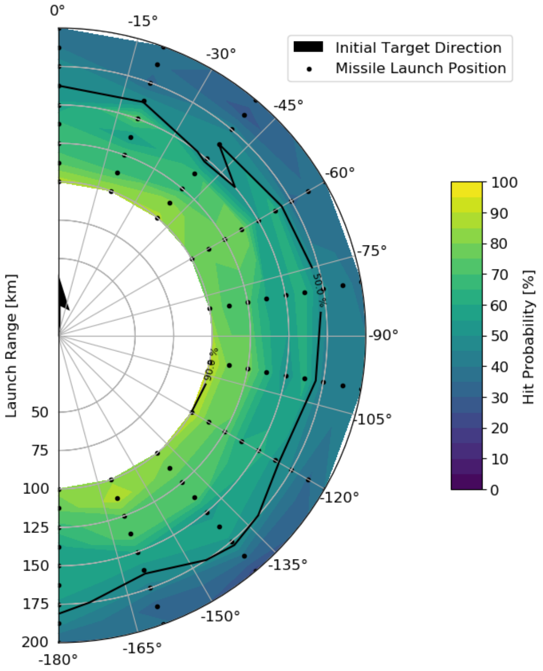


Figure A.29: Hit probability envelopes for PN guidance with lofting at an altitude of 9 km

A.6.2. RL Agent based - PIP-mode 1

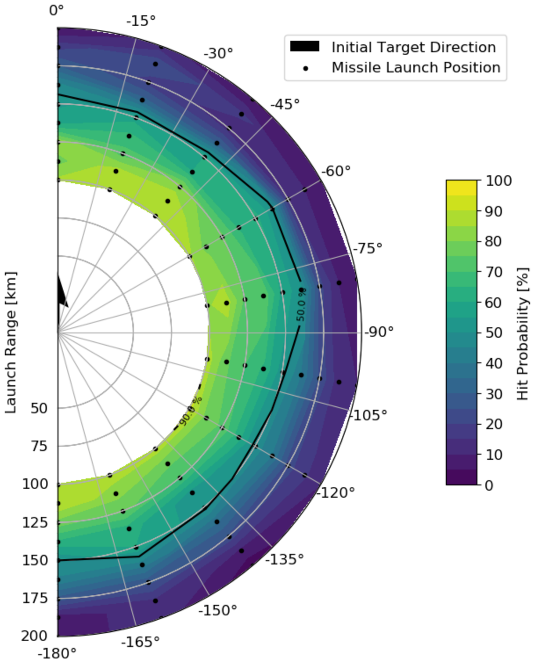


Figure A.30: Hit probability envelope for RL agent based guidance using PIP-mode 1 an altitude of 3 km

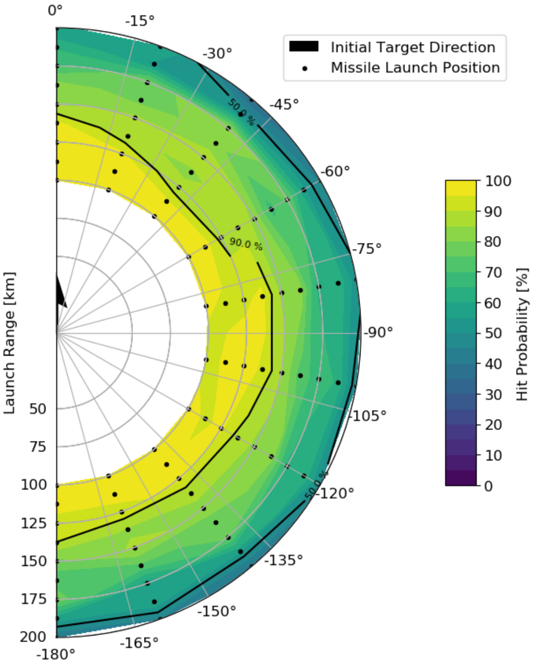


Figure A.31: Hit probability envelope for RL agent based guidance using PIP-mode 1 an altitude of 6 km

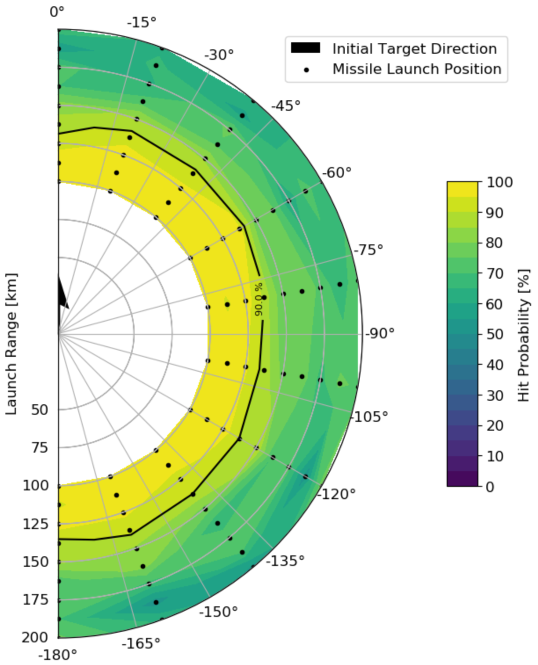


Figure A.32: Hit probability envelopes for RL agent based guidance using PIP-mode 1 an altitude of 9 km

A.6.3. RL Agent based - PIP-mode 2

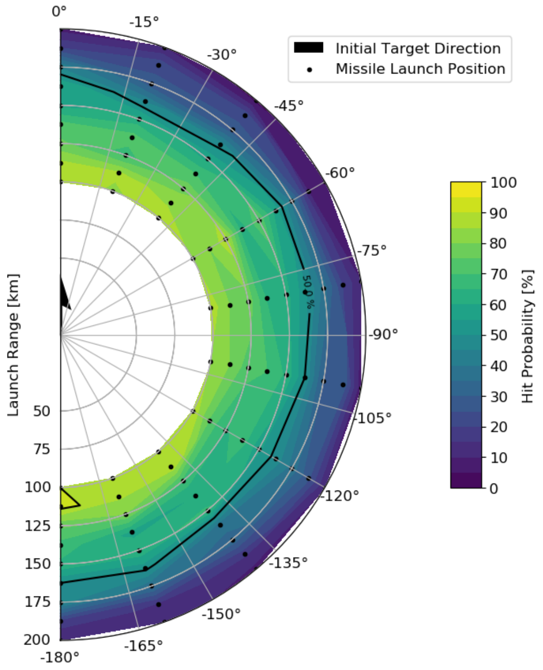


Figure A.33: Hit probability envelope for RL agent based guidance using PIP-mode 2 an altitude of 3 km

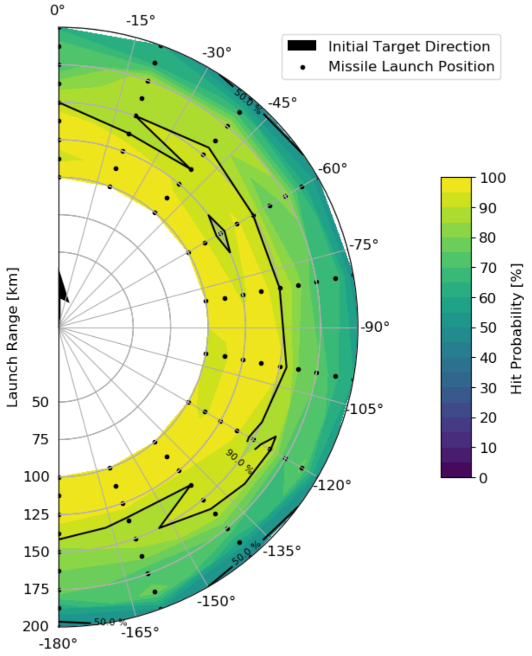


Figure A.34: Hit probability envelope for RL agent based guidance using PIP-mode 2 an altitude of 6 km

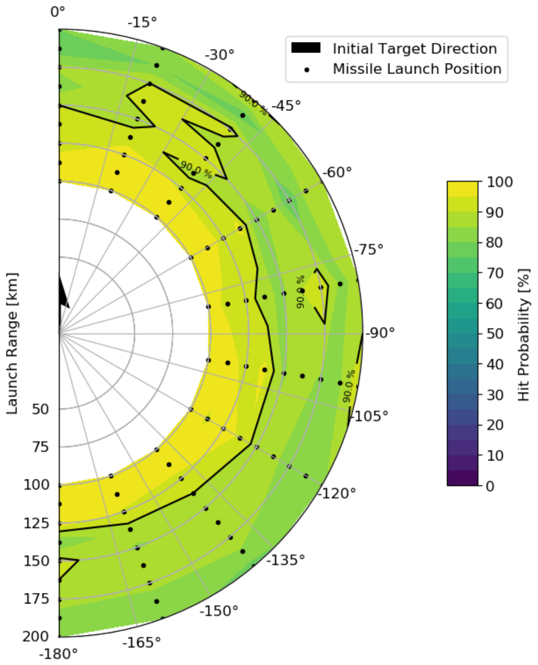
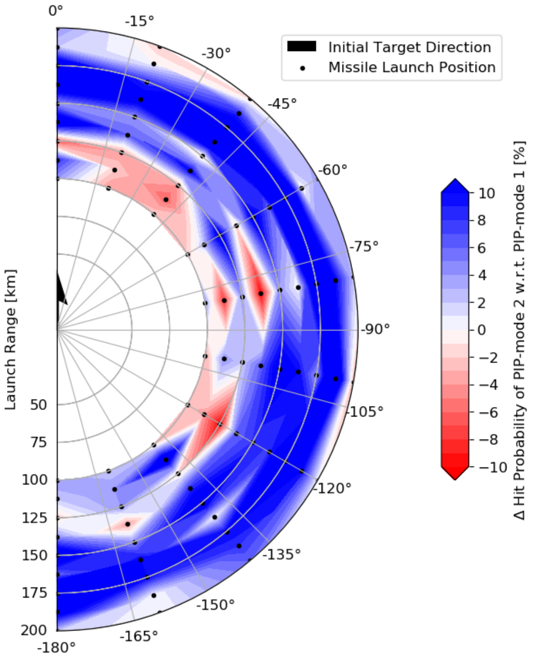


Figure A.35: Hit probability envelopes for RL agent based guidance using PIP-mode 2 an altitude of 9 km

A.6.4. RL Agent based - Comparing PIP-mode 1 and 2



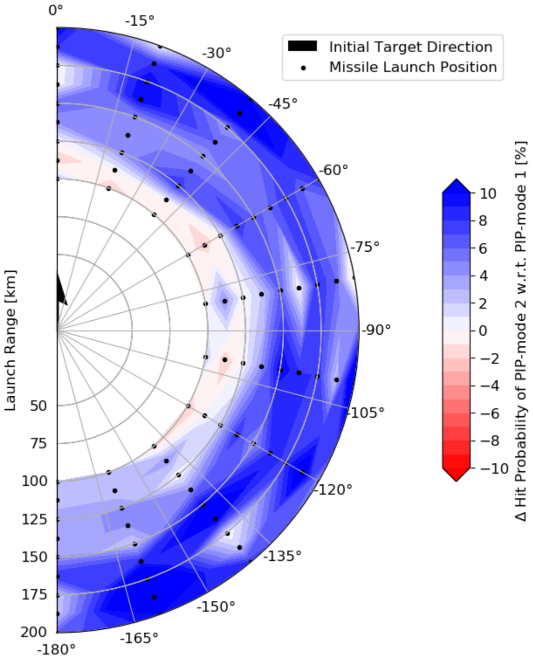


Figure A.36: Hit probability envelope for RL agent based guidance using PIP-mode 2 an altitude of 3 km

Figure A.37: Hit probability envelope for RL agent based guidance using PIP-mode 2 an altitude of 6 km

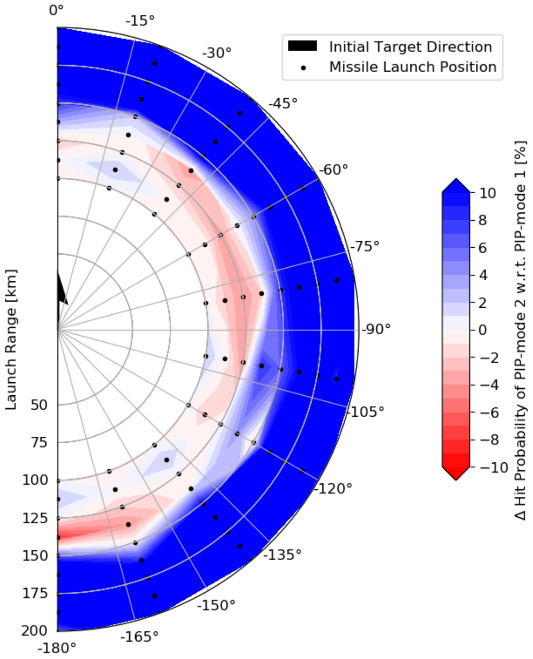


Figure A.38: Hit probability envelopes for RL agent based guidance using PIP-mode 2 an altitude of 9 km