Cooperative Control Strategies for Autonomous Agents Using Nonlinear Model Predictive Control

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

by

Amith M.

Roll no: PhD17106

Under the supervision of Dr. P.B. Sujit

ELECTRONICS AND COMMUNICATIONS ENGINEERING

INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI– 110020

APRIL 17, 2023

Acknowledgements

First, I would like to express my gratitude to my advisor, Dr. P. B. Sujit, who taught me that the three key ingredients to success are hard work, teamwork, and perseverance. I would also like to thank Dr. Rajnikant Sharma for his valuable collaboration during my work. I am also grateful to Dr. Randal W. Beard for the support and suggestions I received during my time at BYU.

The help and advice received from my Ph.D. committee members, Dr. Shobha Sundar Ram and Dr. A. V. Subramanyam, are greatly acknowledged.

I would like to thank the Ph.D. Admin, Ms. Priti Patel, for her assistance during my term at IIIT-D. I also thank my institute for providing the infrastructure and funding, including the overseas research fellowship.

I thank my colleagues in IIIT-D: Dr. Alvika Gautam and Dr. Parikshit Maini, for their support and valuable feedback on this work. Special thanks to my friends Mandeep, Yogesh, Rajat, Shiju, John, Ajay, Jithin, and Midhun. I would also like to thank my fellow labmates and interns in IIIT-D, IISER-B, and BYU.

Last but not least, I would like to thank my family: my father, Manoharan; mother, Seena; brother, Ajith; and my wife, Vandana, for supporting me throughout my life.

Amith

Amith M.
PhD17106

Certificate

This is to certify that the thesis titled " Cooperative Control Strategies for Autonomous Agents Using Nonlinear Model Predictive Control", being submitted by Amith M. to Indraprastha Institute of Information Technology Delhi, for the award of the Doctor of Philosophy, is an original research work carried out under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree.

The results contained in this thesis have not been submitted in part or in full to any other university or institute for the award of any degree/diploma.

Sujit

Dr. P. B. Sujit
Electronics and Communications Engineering
Indraprastha Institute of Information Technology Delhi
April 17, 2023

Abstract

Autonomous aerial vehicles (AAVs) are extensively used in civilian and military applications like aerial surveying, search and rescue, transportation, border patrolling, etc. In most applications, achieving the objectives using a single AAV is difficult. Hence, multiple cooperative AAVs are used to accomplish the mission quickly and efficiently. However, achieving cooperation is challenging in real-world scenarios due to the uncertainties in obtaining other vehicle states (position, velocity, etc.) and measurement information. The constraints, such as limited sensor range and availability, noises, and environmental disturbances, must be handled properly to obtain an efficient system. In this thesis, we provide solutions for a three-agent and four-agent pursuit-evasion problem, path planning under localization constraints, and tracking ground vehicles for cinematography purposes. The optimal control commands for the cooperative agents in each of these problems are found using the nonlinear model predictive control (NMPC) framework. We analyze the theoretical properties of the proposed solutions and show the performance through numerical simulations. Brief explanations of the proposed solutions are given in the following paragraphs.

Chapter 2 presents a cooperative target defense guidance strategy using a nonlinear model predictive control (NMPC) framework for a target-attacker-defender (TAD) problem. The TAD problem consists of an attacker and a cooperative target-defender pair. The attacker's objective is to capture the target, whereas the target-defender team acts together such that the defender can intercept the attacker and ensure target survival. We assume that the cooperative target-defender pair do not have perfect knowledge of the attacker states, and hence the states are estimated using an extended Kalman filter (EKF). The capture analysis based on the Apollonius circles is performed to identify the target survival regions. The efficacy of the NMPC-based solution is evaluated through extensive numerical simulations, and hardware experiments performed using ground rovers. We also compare our approach against previous studies,

which use the command to line of sight (CLOS), and augmented command to line of sight (A-CLOS) guidance strategies.

Chapter 3 presents a cooperative target defense strategy using nonlinear model-predictive control (NMPC) framework for a two–target two–attacker (2T2A) problem. Each attacker needs to capture a designated target individually. The objective of the two targets is to cooperate such that they lure the two attackers into a collision. We assume that the targets do not have perfect knowledge of the attacker states, and hence they estimate the attacker states using EKF. The NMPC scheme computes closed-loop optimal control commands for the targets while satisfying state and control constraints. Theoretical analysis is carried out to determine escape regions that will lead to the targets' survival or capture. Numerical simulations are carried out to evaluate the performance of the proposed NMPC-based strategy for different scenarios validating the theoretical results.

From Chapters 2 and 3, we observe that the NMPC-based solution offers robustness to the different unknown attacker models and has better performance than the CLOS and A-CLOS based strategies. With the help of escape zone maps, it is now possible to identify the outcome of the games beforehand. Also, the experimental results proved that the proposed online-feedback scheme is a suitable alternative to conventional optimal control techniques for real-world scenarios.

In Chapter 4, we solve a joint cooperative localization and path planning problem for a group of autonomous aerial vehicles (AAVs) using nonlinear model predictive control (NMPC). The vehicles do not have access to global satellite navigation systems (GNSS). A moving horizon estimator (MHE) is used to estimate the states with the help of relative bearing information to known landmarks and other vehicles. The goal of NMPC is to devise optimal paths for each vehicle between a given source and destination while maintaining desired localization accuracy. The localization error covariance of the vehicles for the NMPC prediction window was calculated using an approximate analytical expression based on the relation between the covariance and path lengths to the landmarks. We show that a vehicle's position accuracy is inversely proportional to the path length to the landmarks. The use of this analytical expression reduces the computation requirement of NMPC compared to the traditional method of propagating and estimating covariances using an extended Kalman filter (EKF). Finally, we present numerical simulation results that validate the proposed approach for different numbers of vehicles and landmark configurations. The proposed framework is useful in the area of urban mobility, where many autonomous aerial vehicles fly through urban areas with buildings and other structures performing tasks such as cargo delivery. The proposed framework allows these AAVs to localize themselves cooperatively in the absence of GNSS signals since, in urban canyons, the accuracy of GNSS is affected by phenomena such as multipath.

In Chapter 5, we introduce a learning-based nonlinear model predictive control (L-NMPC) scheme for the iterative task of filming race cars using gimbaled cameras mounted on autonomous aerial vehicles (AAVs). The controller is capable of avoiding inter-vehicle collisions and the environmental obstacles that block the path of the AAVs. It also ensures that the cars always lie in the field of views (FOVs) of the cameras while satisfying the control and state constraints. The controller is able to learn from the previous iterations and improve the tracking performance with the help of reinforcement learning (RL). Simulation results are given to demonstrate the efficacy of the proposed learning-based control scheme. The proposed scheme helps reduce manual effort in tuning weights for the cost components of the NMPC. Also, with the help of RL-tuned weights, the NMPC scheme gives tight tracking of the cars even in environments containing obstacles.

Finally, in Chapter 6, we conclude the thesis by providing the inferences from the various experiments and simulations for different problems discussed in Chapters 2-5. The main extensions for the NMPC framework discussed in the thesis are to broaden the framework to three dimensions taking the terrain map and urban environments into account. The current NMPC solver suffers from a high value of computation time for such complex environments with a large number of agents. Hence, for real-time implementation, we would like to drive our approach toward the use of fast-MPC based solvers for speeding up the computations. In the future, we would like to use the actual dynamics for the agents instead of point-mass models. Also, environmental challenges like wind disturbances could be taken into effect when formulating the control law for the agents. A detailed description of future directions for each problem is given at the end of this chapter.

In summary, this thesis focuses on developing solutions for different applications – pursuit- evasion games, path planning under uncertainty, and cinematography using the NMPC framework. We analyze the theoretical properties of the proposed solution and show the performance through simulations and experiments.

List of publications

Journals

1. A. Manoharan, and P. B. Sujit, "Nonlinear Model Predictive Control Framework For Cooperative Three-Agent Target Defense Game." Accepted for publication in Journal of Intelligent and Robotic Systems, March 2023.
2. A. Manoharan and P. B. Sujit, "NMPC-Based Cooperative Strategy to Lure Two Attackers Into Collision by Two Targets," in IEEE Control Systems Letters, vol. 7, pp. 496-501, 2022, doi: 10.1109/LCSYS.2022.3195819.
3. A. Manoharan, R. Sharma, and P. B. Sujit. Multi-AAV Cooperative Path Planning using Nonlinear Model Predictive Control with Localization Constraints. Submitted to IEEE Transactions on Intelligent Transportation Systems, December 2022.
4. A. Manoharan, D. Soni, P. Tyagi, and P.B. Sujit, Learning-based NMPC Framework for Car Racing Cinematography Using AAVs. To be submitted to IEEE Transactions on Aerospace and Electronic Systems.

Conferences

1. D. Soni, A. Manoharan, P. Tyagi and P. B. Sujit, Learning-based NMPC Framework for Car Racing Cinematography Using Fixed-Wing UAV, International Conference on Unmanned Aircraft Systems (ICUAS), 2022, pp. 1397-1403, doi: 10.1109/ICUAS54217.2022.9836154.
2. A. Manoharan, M. Singh, A. Alessandretti, J. G. Manathara, S. Prusty, N. Mohanty, I. S. Kumar, A. Sahoo, and P. B. Sujit, Nmpc based approach for cooperative target defence, in American Control Conference (ACC), 2019, pp. 5292–5297, doi: 10.23919/ACC.2019.8815386.
3. A. Manoharan, R. Sharma, and P. B. Sujit, Nonlinear model predictive control to aid cooperative localization, in International Conference on Unmanned Aircraft Systems (ICUAS), 2019, pp. 26–32, doi: 10.1109/ICUAS.2019.8797888.
4. M. Singh, A. Manoharan, A. Ratnoo and P. B. Sujit, Three Dimensional UAV Path Following Using SDRE Guidance, International Conference on Unmanned Aircraft Systems (ICUAS), 2019, pp. 482-490, doi: 10.1109/ICUAS.2019.8798259.
5. Y. Kumar, A. Manoharan and P. B. Sujit, Right of Way Rules based Collision Avoidance Approach Using Model Predictive Control, Indian Control Conference (ICC), 2019, pp. 332-337, doi: 10.1109/ICC47138.2019.9123203.

Contents

List of Tables

List of Figures

Nomenclature

A, T, D	attacker, target, and defender
x, y	position coordinates of the agents in the Cartesian frame
t	time
v	velocity of the agent
α or ψ	heading angle (yaw) of the agent
$\bar{v}_T$	maximum velocity of the target
θ	line-of-sight angle between the agents or nodes
ξ	line-of-sight angle between the defender and the attacker
R, r	distance between the agents or nodes
v_R	component of the relative velocity along the $A - T$ LOS
v_θ	component of the relative velocity perpendicular to the $A - T$ LOS
v_r	component of the relative velocity along the $A - D$ LOS
v_ξ	component of the relative velocity perpendicular to the $A - D$ LOS
τ_h	prediction horizon for the continuous-time NMPC
X	state vector
U, U^-, U^+ or $\boldsymbol{\omega}, \boldsymbol{\omega}^-, \boldsymbol{\omega}^+$	control vector and its lower and upper bounds
$\mathcal{PC}$	space of piece-wise continuous function
u_x, u_y	velocity components of the target in x and y directions
e	safe distance of the target from the attacker
k or j	discrete time
f	prediction model for the NMPC or EKF
J	cost function for the nonlinear model predictive control
∇F_X	Jacobian of f with respect to X
h	measurement model for the extended Kalman filter

∇H_X	Jacobian of h with respect to X
z	measurement vector
$\tilde{z}$	predicted measurement vector
q	process noise
μ	measurement noise
P	estimation covariance matrix
ν	innovation parameter or vertex of a graph
K	Kalman gain
Q	state covariance matrix or Q-value for RL
Σ or Γ	measurement covariance matrix
a_A, a_D	lateral accelerations of the attacker and the defender
x_c, y_c	center coordinates of the Apollonius circle
r_{AT}, r_{AD}	radii of the attacker-target and the attacker-defender Apollonius circles
d_c	distance between the centers of the $A - D$ and $A - T$ Apollonius circles
γ_{AT}, γ_{AD}	speed ratios of the target to the attacker and the defender to the attacker
Z_e, Z_c	escape zone for the target and capture zone for the target
σ	standard deviation of the estimation error
κ	proportionality constant of the pure-pursuit guidance law
N	navigation constant of the PN law or the prediction horizon for discrete-NMPC
R_c, r_c	capture radii
$A1, A2, T1, T2$	attacker-1, attacker-2, target-1, and target-2
u_1, u_2	control commands computed for the targets $T1$ and $T2$
$w_1, w_2, w_3, w_4, w_5, W$	weights of the cost function
d_{A1T1}, d_{A2T2}	radii of the $A1 - T1$ and the $A2 - T2$ Apollonius circles
$\gamma_{A1T1}, \gamma_{A2T2}$	speed ratios of $T1$ to $A1$ and $T2$ to $A2$
Y	set of all points on the $y-$axis
Y_1	set of all points on the $A1 - T1$ Apollonius circle
Y_2	set of all points on the $A2 - T2$ Apollonius circle
$\emptyset$	null set
$\overline{y}_1, \underline{y}_1$	interception points of $A1 - T1$ Apollonius circle with the $y-$axis
$\overline{y}_2, \underline{y}_2$	interception points of $A2 - T2$ Apollonius circle with the $y-$axis
S	source location
D	destination

n_v	number of vehicles
n_l	number of landmarks
$G, \mathcal{G}$	graphs
$\mathcal{V}$	node set
$\mathcal{E}$	edge set
n_e	number of edges
ϕ	path in a graph
ϵ	edge in a graph
T_s	sampling time used for discretization
m	current time step
N_E	estimation horizon
$\hat{X}$	estimated states
R_s	circular sensor range
ω	angular velocity of the agent
O	observability matrix
I	identity matrix
i	AAV and car index
n	number of AAVs and cars
θ_A	pitch angle of AAV
q_A, r_A	pitch and yaw rates of the AAV
ϕ_G, θ_G, ψ_G	roll, pitch, and yaw angles of the gimbal
p_G, q_G, r_G	angular rates of the gimbal
v_C	linear velocity of the car
r_C	angular velocity of the car
ψ_C	heading angle of the car
n_o	number of obstacles
R_A	radius of the AAV
R_o	radius of the obstacle
x_o, y_o	coordinates of the center of the obstacle
D_o	Euclidean distance between the AAV and the obstacle
x_F, y_F	center of the FOV
a, b	major and minor axis of the FOV ellipse
$VFOV, HFOV$	vertical and horizontal FOV of the camera

(s, a_{RL})	state-action pair
α_{RL}	learning rate
γ_{RL}	discount factor
r_{RL}	reward function
λ	epsilon decay rate
Acronyms	
AAV	Autonomous aerial vehicle
NMPC	Nonlinear model predictive control
TAD	Target-attacker-defender
EKF	Extended Kalman filter
LOS	Line-of-sight
CLOS	Command to line-of-sight
A-CLOS	Augmented command to line-of-sight
2T2A	Two–target two–attacker
GNSS	Global satellite navigation systems
MHE	Moving horizon estimator
L-NMPC	Learning-based nonlinear model predictive control
FOV	Field of view
RL	Reinforcement learning
ILC	Iterative learning control
MIMO	Multi-input multi-output
PN	Proportional navigation
PP	Pure pursuit
GCS	Ground control station
ROS	Robotic operating system
RPMG	Relative position measurement graph

Chapter 1

Introduction

Autonomous aerial vehicles (AAVs) are used in several applications like search and rescue, smart munition interception, cinematography, etc. Multiple agents that cooperate help to accomplish tasks more efficiently and quickly than single, individually acting agents. Developing optimal cooperative control strategies for these agents is difficult due to the lack of model information, localization constraints, and disturbances produced by dynamic environments. In this thesis, we use nonlinear predictive control (NMPC) as a common framework to solve different problems arising in defense and civilian applications. NMPC was selected due to its feedback structure, which allows online computation and inherent disturbance rejection to some extent. Also, if the prediction horizon is suitably selected, the solutions tend to be near optimal. NMPC uses a nonlinear model of the system and can also handle nonlinear cost functions and constraints. Hence, the design flexibility of NMPC is more compared to the schemes such as linear quadratic regulator (LQR), which uses a linear model and cost function, and the absence of inherent input and state constraints satisfaction.

First, we venture into the area of pursuit-evasion games, which has several applications in the civilian and defense sectors. We start with a three-agent pursuit-evasion problem involving a target, an attacker, and a defender agent, where the objective is to determine the control signals of the target and the defender such that the target escapes from the attacker. We are interested in the cooperative aspect that comes from adding a third agent (defender) to the general pursuer-evader game. Then we extend this formulation to a four-agent game with two targets and two attackers, where the objective of the targets is to cooperate and move such that the two attackers collide with each other. Here, we use the cooperative aspect of the targets to win the game without the need for any external defenders.

We then move on to the problem of cooperative path planning for agents with localization constraints. Multi-agent cooperative localization is important in the future of urban air mobility, where many AAVs, such as cargo drones, will be operating in dense cities where GNSS availability is weak. In such cases, relative measurements from known features, such as a mobile phone tower, can be used for localization. Here, we decided to switch to the moving horizon estimation (MHE) framework instead of EKF due to its nonlinear model handling capabilities. Using a nonlinear model will provide better estimation accuracy compared to the linearized versions. The objective is to ensure that a group of AAVs reach their respective goal locations. The vehicles localize using known landmarks and other nearby vehicles. The path from the source to the goal should be such that the vehicles cover enough landmarks along the way to maintain localization accuracy.

Finally, we present a learning-based NMPC (L-NMPC) strategy to accurately film a car race using autonomous aerial vehicles (AAVs). Here, we try to improve the NMPC scheme by eliminating some of its drawbacks by combining it with reinforcement learning (RL). The problem of filming a circuit race was chosen due to its iterative nature. The weights for the NMPC cost function were tuned using an online Q-learning approach. This helps in getting accurate tracking of the cars even in the presence of obstacles. The following sections provide an overview of the control of multi-agent systems and detailed explanations of the problems under consideration.

1.1 Control techniques for multi-agent systems

Multi-agent systems theory focuses on developing control laws for various tasks, such as consensus and formation control [1–3], coverage control [4, 5], target tracking [6–8], optimization [9], and estimation [10]. The control techniques for multi-agent systems can be mainly classified into three types; centralized [11], decentralized [12], and distributed [13]. In the centralized scheme, the multi-agent system is controlled using a single entity with high processing capability. Obviously, this strategy suffers from scalability issues since the computational complexity increases with the number of agents. The decentralized control scheme uses independent agents with local controllers. This type of scheme performs less than the centralized scheme since the agents do not possess global information. The distributed control scheme reduces the computation requirement by sharing information only with the neighboring agents.

Optimal control is one of the most widely used control techniques for multi-agent systems [14]. In [9], a distributed computation model for optimizing a sum of convex objective functions corresponding to multiple agents is studied. The formulated optimization problem is solved using a distributed subgradient method. An inverse optimality formulation that considers the interplays between the communication graph topology and the agent dynamics is considered in [15]. In [16], an optimal control framework for persistent monitoring problems is given, where the objective is to control the motion of multiple agents to minimize an uncertainty metric.

Over the last decades, model predictive control (MPC) has become the control technology of choice for controlling complex, dynamic, multi-agent systems [17, 18]. In [19], a non-cooperative distributed model predictive control algorithm for tracking constant references is proposed. The algorithm was tested by experiments performed on a quadruple tank process. In [20], a model predictive control algorithm to generate non-colliding trajectories for multiple robots is discussed. Also, an event-triggered re-planning strategy is proposed to account for disturbances. A tube-based NMPC scheme for a general class of uncertain nonlinear multi-agent systems is developed in [21]. The paper focuses on the problem of robust navigation of a multi-agent system using only local information under input constraints. In [22], a robust-MPC strategy for solving the problem of observation of multiple moving targets by cooperating mobile robots is presented. The optimization is based on linear mixed-logic dynamic models with flexible weights. An NMPC-based approach for target perception using a group of mobile robots is given in [23]. The robots also maintain a desired formation when following the target. The multi-robot collision avoidance problem in unknown environments is solved using MPC in [24]. The proposed system works based on the information-seeking theory and cooperative observations. In [25], a cooperative particle swarm optimization-based MPC scheme is proposed to solve the formation control problem of multiple nonholonomic mobile robots. The algorithm is used to find a Nash equilibrium between the multiple robots.

The following sections present solutions to some challenging problems involving multi-agent systems using nonlinear model predictive control.

1.2 Three-agent and four-agent pursuit-evasion problems

Pursuit-evasion problems are useful in several important security applications [26]. These problems consist of two agents, (i) pursuer agent (P) and (ii) evader agent (E). The objective of the pursuer is to capture the evader, while the evader wants to avoid capture. The most common application of such problems is when an aircraft tries to evade an incoming missile. A broad survey on pursuit-evasion differential games is given in [27], and several other studies can be found that analyze problems with a single pursuer-evader pair [28,29], single pursuer against multiple evaders [30,31], and multiple pursuers and a single evader [32–35]. In [28], collision cones are used for analyzing pursuit-evasion games between two objects of arbitrary shapes. A spacecraft pursuit-evasion game is formulated as a two-player zero-sum differential game in [29] and is solved using the linear quadratic differential game theory. Breakwell and Hagedorn [30] use a constant-speed coplanar model with unlimited turn rates to formulate a simple geometrical solution for the problem of point capture of two successive evaders in minimum total time. In [31], a single-pursuer, two-evader differential game is formulated with a cost functional that represents the increased cost to the pursuer when presented with multiple potentially dangerous targets. Pan et al. [32] study a game played in a convex domain with an exit through which the evader may escape. The strategy is based on the Voronoi diagrams. In [33], a reachability-based approach is adopted to deal with the pursuit-evasion differential game between one evader and multiple pursuers in the presence of dynamic environmental disturbances. A cooperative pursuit strategy using overlapping Apollonius circles around the evader is presented in [34] to capture the high-speed evader using multiple pursuers. In [35], state feedback capture strategies and an evader strategy which yields a lower bound on his time-to-capture, are devised using a geometric method for a multi-pursuer single-evader problem.

In the presence of multiple agents, cooperation among them becomes important. One such problem where there are multiple agents is the three-agent target-attacker-defender (TAD) problem. The TAD problem consists of an agent named the attacker (A), who pursues to capture the second agent known as the target (T), and the third agent, called the defender (D), tries to help the target by intercepting the attacker before it reaches the target. The target and the defender act as a team, and this cooperation enables the target to maneuver in such a way that the defender is able to intercept the attacker promptly.

The TAD problem was introduced by Isaacs in [36] and was discussed in detail by various researchers [37–43]. Diverse approaches were taken to tackle this problem, including line-of-sight (LOS) guidance [37,38], linear quadratic regulator [39,44], and optimal game theoretical solutions [42,43,45]. In [46], the TAD problem is solved using optimal control theory wherein the optimal heading angles for the target-defender team are determined, assuming that the attacker implements a conventional missile guidance law such as proportional navigation (PN) or pure pursuit guidance (PP). Another optimal control formulation with bounded controls is given in [47]. In [48], a nonlinear guidance strategy using the sliding mode control technique is proposed for various scenarios involving an attacking missile, a target aircraft, and a defender missile. In [49], the TAD problem is solved using a simple geometrical approach referred to as the triangle guidance, which was also extended to multiple attacker scenarios. The TAD solution using optimal control theory with escape regions for the target is given in [50]. Weiss et al. [51]

proposed two guidance algorithms for the TAD problem. The first one is a combined guidance algorithm for the attacker that simultaneously achieves evasion from the defender and pursuit of the target. The second is a cooperative guidance algorithm for the target-defender pair to enable target escape and the interception of the attacker by the defender. In [52], a three-agent formulation different from the classical TAD problem is presented where a navigating aircraft pursues a target and simultaneously tries to avoid an incoming bleeding-energy missile that is assigned to defend the target. In [53], a multiplayer TAD problem is discussed, where Lyapunov-based control strategies are derived for the players using approximations of the minimum and maximum functions. The solutions obtained through analysis of different agent combinations are utilized to pair defenders with attackers to maximize the number of attackers captured. In [54], the approach does not require perfect knowledge of the states as the agents estimate through the range and heading measurements using an EKF. Manyam et al. [55] presented a path-planning problem modeled as a TAD involving two cooperative agents in an adversarial environment. The second vehicle helps the first vehicle complete a mission by defending against any attackers along the path.

A natural extension of the three-agent problem is the four-agent problem. Casbeer et al. [56] introduce a four-agent game by introducing an additional defender to the TAD game proposed in [50]. Garcia et al. [57] presented an assignment problem for a multi-pursuer multi-evader differential game, where optimal assignments of pursuers to evaders are studied. Tan et al. [58] developed a state-dependent-Riccati-equation (SDRE) based approach to lure the two pursuers into collision for the two targets' survival. The approach does not present any theoretical analysis of whether the targets will survive or not.

In the two-target two-attacker (2T2A) problem, there is no need for external defenders to protect the targets. The objective of the targets is to move in such a manner that they force the attacker pursuing the other target to act as a defender. We advance the 2T2A problem formulation to use NMPC for the computation of optimal control commands. We show that the 2T2A problem can be formulated in such a way that the outcome is determined by the initial positions of the attackers and the targets. To facilitate this outcome, we analyze in detail the escape region for the targets theoretically.

The approaches presented in [37–43] assume that (i) perfect information about attacker states and guidance laws employed by it are known, by which closed-form solutions were developed, and (ii) the target is always at motion, generally with constant speed. In this work, we relax the above assumptions by using a control scheme that combines the NMPC with attacker state estimation using an EKF. The attacker positions, headings, and controls are estimated, whereas the velocities are assumed to be known.

We also consider the case in which the target has the freedom to move or not to move. This modification is significant for applications like aerial surveillance, communication relaying, etc., where some information needs to be acquired from hostile environments by hovering over the prescribed location. In such situations, the defender can protect the target from any attacking agents while allowing the target to continue its mission. The escape regions are only given for the constant speed target case in [42], which also assumes that the attacker and the defender have equal velocities. We extend the analysis for the variable target velocity case and relax the assumption of equal velocities for the attacker and the defender. Since the NMPC computes control commands based on the current state estimates, it provides the flexibility to adapt to situations where the attacker can be intelligent. This adaption is not possible in the case of open-loop optimal control formulations where the solution is predetermined [41, 45, 47, 56]. The formulations which use closed-loop solutions [58–60] linearize the system resulting in approximation errors. Our approach is based on the nonlinear model of the system. Also, the inherent constraint handling of NMPC helps to design controls that strictly adhere to the specified bounds. The NMPC also has the advantage of combining optimality with real-world implementability, whereas the conventional missile guidance laws are either sub-optimal or difficult to implement.

Now we move on from pursuit-evasion problems to a different set of problems in cooperative localization and path planning for autonomous agents used in urban air mobility. Here, instead of EKF, we introduce a moving horizon estimation (MHE) scheme for the state estimation due to its advantages explained in later paragraphs.

1.3 Cooperative Path Planning with Localization Constraints

Urban air mobility (UAM) is expected to have highly automated, cooperative, passenger, and cargo-carrying aerial vehicles in urban areas [61], and the use of autonomous aerial vehicles (AAVs) for various activities are expected to rise substantially in the near future [62]. Cargo delivery drones operate in urban canyons with high-rise buildings and other obstructions, which calls for significant localization accuracy. However, operating in such environments poses an additional challenge in localization since Global navigation satellite systems (GNSS) are unreliable in such scenarios. A solution to this problem is to use alternate localization schemes such as relative localization [63, 64], vision-based methods [65, 66], and ultra-wide-band (UWB) localization schemes [67, 68]. As the urban airspace is expected to contain a large number of AAVs, relative localization between vehicles can also be used in addition to localization with respect to known landmarks.

Cooperative path planning with localization constraints involves the following components: (i) localization – vehicles estimate their position by using relative measurements obtained with respect to other vehicles or landmarks, and (ii) cooperative path planning – localization constraints have to be considered while planning the optimal path from source to destination, and hence the agents must cooperate with each other to generate motion commands that improve the localization accuracy of the entire group while reducing the path length to reach their destination. Several works have studied the problem separately. For instance, [69–80] on cooperative localization and [81–84] on path planning. However, the collection of works on cooperative path planning with localization constraints is limited, which we will review below.

Bopardikar et al. [85] presented a graph-based probabilistic roadmap approach to tackle the path planning problem subject to localization constraints. A time-optimal path planner satisfying the covariance bounds was given in [86] using a swarm optimization technique coupled with a rabbit-carrot-based path follower. An approach to optimally place landmarks to satisfy localization constraints was proposed in [87]. The algorithm computes an optimal path for the vehicle and the locations where the landmarks should be placed. A localizability-constrained path planning method for autonomous vehicles which takes into account the laser range finder (LRF) sensor model of the vehicle, is proposed in [88] to maintain a satisfactory level of localizability throughout the path. Yang et al. [89] present a multi-objective motion planning algorithm in which the vehicle tries to balance the objectives of navigating to the waypoint and reducing its position estimate uncertainty. In [90], a nonlinear model predictive control scheme was used to find optimal paths that ensured the required localization accuracy for a group of cooperative UAVs without analysis.

Urban air mobility calls for improved localization accuracy due to its innate nature involving close structures, narrow pathways, and a large number of vehicles. Moving horizon estimation (MHE) has been suggested as an alternative to EKF for increasing the accuracy of nonlinear estimation problems [91–93]. Erunsal et al. [94] proposed an approach combining NMPC and pose-graph-MHE for 3D formation control of micro aerial vehicles with relative sensing capability. In [95], a decentralized MHE technique is proposed for networked navigation with packet dropouts.

In this thesis, we extend the work in [90] and propose a complete cooperative localization and path planning framework with moving horizon estimation (MHE) for estimating the vehicle position and a nonlinear model predictive control (NMPC) framework for cooperative path planning, and a closed formulation for covariance calculation to predicting the uncertainty. This framework uses a nonlinear vehicle model in both the controller and the estimator, which mitigates the linearization errors. The analytical expression used for the covariance calculation speeds up the computations and is derived by exploiting the relationship between the vehicle-landmark path lengths to the localization uncertainty. In most of the literature for multi-agent systems, the studies are formulated either as a control problem or an estimation problem [70, 72–74, 92]. We propose a method that combines both and looks at the multi-agent problem in a holistic sense.

Previous studies [74, 82, 83] used algorithms that were not optimal and were based on conventional path planners, whereas nonlinear model predictive control (NMPC) provides optimal paths while satisfying imposed state and input constraints. The feedback and re-planing at each step help find accurate solutions in dynamically changing environments like moving landmarks. Also, most of the works based on graph-based planning give discrete (node-to-node) paths [85]. Our design gives a continuous path, eliminating the shortcomings of former approaches. The agent connectivity is maintained by ensuring nonzero eigenvalue of the graph Laplacian in [96] and [97]. We also use a Laplacian matrix-based solution, but we calculate Laplacian matrices for each vehicle instead of forming a single matrix for the whole group. This helps in identifying connectivity at an individual level, and each vehicle can decide to maintain, lose, or gain connections depending on their covariance estimates. This is not possible in the single matrix case because a single vehicle with more connections will increase the Laplacian, even though the other vehicles may be out of the required number of connections. Another advantage of the proposed scheme is that we relax the constraint of always maintaining connections with two landmarks [87] since sometimes it might be impossible to obtain a solution in environments with a lower number of landmarks. Also, if the observability condition is violated for a short time, the vehicles would be able to re-localize if it comes in the vicinity of another landmark. So we formulate a condition in the controller that tries to increase connections with the landmarks only if the estimation covariance of the vehicle increase beyond a given threshold. This formulation helps in obtaining solutions to the path planning problem while ensuring the required localization accuracy. The works on path planning with localization constraints [86–89] consider only single vehicles and fixed landmarks in most cases, whereas our approach includes multiple vehicles, and each vehicle is considered a moving landmark. Since the NMPC computes predictions for a time window, the anticipated covariances need to be calculated for the entire prediction horizon. Existing methods of propagating covariance through an EKF-like update equation will drastically increase the computation time since it involves the calculation of Jacobians, and for long prediction horizons, this will result in a significant increase in the computation time. Hence, an approximate analytic expression was derived for finding the vehicles' estimation covariances by analyzing how the path lengths between the vehicles and the landmarks in the relative position measurement graph (RPMG) affect the covariance values.

Next, we present a learning-based control strategy by combining reinforcement learning (RL) and NMPC to leverage the advantages of both methods and eliminate their individual shortcomings. The learning-based NMPC (L-NMPC) is applied for the control problem of gimbaled camera-mounted AAVs filming race cars.

1.4 Learning-based NMPC Framework for Car Racing Cinematography Using AAVs

The problem of persistent tracking, wherein one agent tries to always maintain another mobile agent within its field of view (FOV), is an interesting problem studied by many researchers. Persistent tracking is mostly used in the domain of aerial-to-ground settings, where an autonomous aerial vehicle (AAV) tracks a ground vehicle. Choi and Kim [98] presented a guidance law for target tracking using AAV for persistent monitoring with the use of a monocular-vision sensor. In [99], AAV rendezvous and standoff tracking guidance laws against a moving target are presented using a differential geometry approach. A control strategy based on lateral guidance law was proposed in [100] to visually track a ground target with a fixed-wing AAV. The approach aims at reaching a given target view angle, which is determined on the basis of simple geometric considerations. In [101], a framework for moving vehicle detection, tracking, and geolocating is proposed, which uses a monocular camera, a GPS receiver, and inertial measurement units (IMUs). A solution to the problem of ground target tracking using an AAV with control input constraints is given in [102]. In order to achieve precise target tracking, a saturated heading rate controller based on a guidance vector field is proposed.

Since the last decade, AAVs have been used for a wide range of applications, with aerial videography being the most popular. Despite being a topic of interest, the research on filming a moving object, considering realistic constraints and obstacle avoidance, is inadequate. Bonatti et al. [103] proposed a complete system for real-time aerial cinematography that combines a vision-based target estimation and a 3D signed-distance mapping for occlusion estimation. A framework for the AAV to maintain a horizontal circular orbit about the target with a predefined radius is proposed in [104]. In [105], Quintero and Hespanha designed optimization-based strategies using game theory for a pair of fixed-wing AAVs and showed that the visibility could be improved by using a gimbaled camera. The author assumed that AAVs could only fly at a fixed altitude in an obstacle-free environment. Optimization-based methods such as the greedy method and informed tree search were explored with the presence of obstacles and visibility constraints by Theodorakopoulos and Lacroix [106]. However, the approach is restricted to 2D environments only. Vision-based air-to-ground target tracking problem was solved with stochastically optimized guidance law in urban environments while maximizing the navigation accuracy in [107]. Altan and Hacıoglu [108], and Mali et al. [109] proposed a model predictive control strategy for a target tracking AAV in an obstacle-free environment. The approach presented in [109] was advanced in [110] by including obstacles in the environment, visibility constraints due to the limited field of view of the camera, and visibility obstruction due to the obstacles. These constraints increase the complexity of the multi-objective optimization problem and make the tuning of NMPC weights a tedious task.

Iterative learning control (ILC) is the control strategy that allows the controller to learn from the previous iteration and increase the future performance of the system while rejecting periodic disturbances. ILC can be applied to systems that repeat certain tasks over time, and each repetition is known as an “iteration” or a “trial” where the system starts at the same initial conditions. ILC is exhaustively studied in literature [111–113]. ILC follows a model-free control strategy that inherently has limitations on constraint handling. Combining ILC with a model-based approach such as nonlinear model predictive control (NMPC) can compensate for the unavailability of an accurate model of the system or unmeasured disturbances and increases the performance of the system while respecting the imposed control and state constraints. The concept of integrating the ILC with MPC has been explored in [114], where the author proposed a control technique called batch-MPC (BMPC) for a time-varying MIMO system. The effectiveness of the approach was shown through experimental results. In [115], Lee and Lee proved that the tracking error of the BMPC converges to zero as the number of iterations increases. In [116], a nonlinear model predictive controller based on iterative learning control was proposed, where the NMPC is designed for disturbance rejection and the ILC to minimize the errors occurring at each iteration. The authors proved that the steady-state tracking error converges to zero as the number of iterations goes to infinity. Artificial intelligence techniques such as model-free reinforcement learning (RL) are considered one of the best options to learn from the previous iterations [117]. The authors proposed an approach where RL was used to learn the length of the prediction horizon of MPC and show the increase in resulting performance. In [118], an algorithm that combines methods from RL and MPC is given for building energy management. A formulation combining system identification with RL-based MPC is proposed in [119, 120]. The approach learns the parameters of the MPC using RL in order to optimize performance and fits the observed model using system identification. In [121], a learning-based MPC scheme is proposed for adaptive optimal control of discrete-time nonlinear systems under stochastic disturbances. In [122], the use of RL to tune the MPC scheme, which guarantees stability and safety, is proposed. In the works [123] and [124], authors used the RL technique to tune the weights of the MPC objective function. An iterative learning control guided RL scheme for batch processes is given in [125].

This combined scheme has the capability to optimize the policy network faster and also improves the robustness of the control system.

We use the idea behind iterative learning control to develop an RL-based NMPC scheme. The learning part is done by RL, and the control part is done using NMPC. Deviating from the previous studies of [117, 123, 124], which consider non-repetitive tasks, we are interested in the repetitive task of filming a circuit race of autonomous robot-race cars with the use of gimbaled fixed-wing and quadrotor autonomous aerial vehicles (AAVs). We extend the formulation given in [110] for this purpose by combining a reinforcement learning approach to auto-tune the weights to get the optimal performance. We use the repetitive nature of the race to train the RL-agent (AAV) online, and the control system performance increases over time and converges to the optimal value. Previous studies on the learning-based MPC used offline training to tune the MPC parameters [123, 124]. Also, the RL was used to learn a constant set of weights, which will then be used for the entire duration of the actual mission. We propose to use time-varying weights, which will give the perfect balance for the cost function, and we learn these weights on the fly, i.e., the training takes place online in the actual mission, and after each iteration, the performance of the controller increases, and the weights converges to the optimal value over time. We identify this advancement as the main contribution of this work.

1.5 Contributions

The main contributions of this thesis are

Three-agent and four-agent pursuit-evasion problems

1. NMPC formulation for the TAD and 2T2A problems with relaxed assumptions on the attacker states and guidance laws. We do not assume complete knowledge of the state information of the attackers.

2. A variable velocity target case for the TAD problem where the target moves only when necessary. Previous studies only considered either a stationary target or a moving target.

3. Theoretical analysis of the escape regions for the proposed three-agent problem for the constant speed target and the variable velocity target and for the four-agent 2T2A problem.

4. Validation of the real-world implementability of the NMPC scheme through hardware experiments

conducted using ground rovers.

Cooperative localization and path planning problem

5. A complete framework for control and estimation of multi-vehicle cooperative path planning problem with localization constraints using NMPC and MHE. The vehicle states are estimated using MHE based on the relative bearing measurements with respect to the nearby landmarks and vehicles. The paths from source locations to goal locations for the vehicles are found in real-time by the NMPC, considering the constraint that the estimation covariances should not go above a selected threshold.

6. An analysis was conducted to find the relation of path lengths between vehicles and landmarks on the estimation covariance. Based on this analysis, an approximate closed-form analytical expression for computing the localization error covariance was derived.

7. Evaluation of the proposed joint cooperative path planning with localization constraints framework through numerical simulations for different configurations of landmarks, vehicles, horizon lengths, and sensing ranges.

8. Comparison of cooperative and non-cooperative vehicles, and MHE-based estimation against EKF-based framework.

Learning-based NMPC tracking problem

9. A learning-based NMPC scheme considering a 3D path with obstacle avoidance for the tracking and filming of race cars using gimbaled camera-mounted AAVs.

10. The NMPC cost function contains adaptive time-varying weights, whereas the previous formulations used constant weights.

11. The weights of the NMPC cost function are tuned using an RL scheme instead of manual tuning.

12. Previous studies used the weights learned offline. We propose an online learning approach where the weights are learned and improved from each iteration of the race.

1.6 Organization of the Thesis

The proposed nonlinear model predictive control strategy for the active defense of the target in a TAD problem is presented in Chapter 2. Cooperative control commands for the target-defender pair are computed using NMPC. The attacker states are estimated using an EKF. An analysis for finding the escape regions for the target and a comparison against the CLOS and A-CLOS formulations are given.

The pursuit-evasion framework is extended to four agents in Chapter 3. In this chapter, we propose a cooperative strategy based on NMPC for the active defense of the targets in a two–targets two–attackers (2T2A) problem. The theoretical analysis using the Apollonius circles given in Chapter 2 is extended to determine the escape region for the targets.

In Chapter 4, we move on to the domain of localization and path planning and propose a nonlinear model predictive control scheme combined with moving horizon estimation (MHE) to aid the cooperative localization of a group of AAVs in transit. The controller uses an approximate analytical expression for calculating the expected covariance of the vehicles, which is derived using theoretical analysis conducted using the observability graphs.

In Chapter 5, we try to improve the performance of the NMPC scheme by combining it with the reinforcement learning technique. The proposed learning-based NMPC controller is used for the tracking and filming of race cars using gimbaled camera-mounted AAVs.

Finally, in Chapter 6, we present the conclusions obtained from the results, the limitations, and the future directions of the work.

Chapter 2

NMPC Framework For Cooperative Three-Agent Target Defense Game

2.1 Introduction

Consider a scenario shown in Fig. 2.1 where a quadcopter is hovering above an enemy territory to capture images. During this process, any attack towards the quadcopter can be intercepted by the defender which is fired from nearby allies so that the quadcopter gets enough time to complete the mission. Here, the target is stationary when performing the mission and will move only when necessary. The condition in which the target moves is explained later in this chapter in Sec. 2.3.1. This three-agent problem consisting of an attacker and a cooperating target-defender team has several applications like aerial surveillance, communication relaying, etc., where some information needs to be acquired in hostile environments at prescribed locations. In such situations, the defender tries to defend the target from attacking agents.

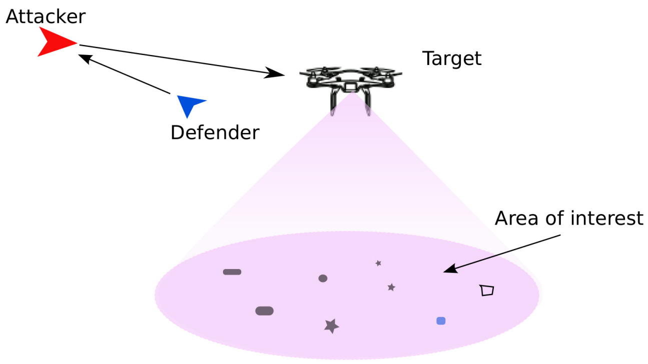


Figure 2.1: Target-Attacker-Defender scenario

2.2 Problem formulation

Consider an attacker A pursuing a target T while the defender D aims to intercept the attacker before A intercepts T as shown in Fig. 2.2. The attacker acts individually, whereas the target and the defender cooperate to form a team. Our objective is to find the optimal control commands for the target-defender team so that the target escapes and the attacker is intercepted. Computing these inputs is not trivial since the attacker states need to be estimated, and the turn rates of the target-defender team are constrained. The game is formulated in a 2D Cartesian space, assuming that the altitude of the agents remains constant throughout the engagement. All the variables are defined with respect to the inertial reference shown in Fig. 2.2. Several assumptions are considered in this article, which are listed below.

2.2.1 Assumptions

Assumption 1. *The agents have point mass, and hence we consider the kinematic equations only.*

Assumption 2. *The target-defender team does not have information about the attacker states. They are estimated, whereas the attacker's velocity is assumed to be known.*

Assumption 3. *The attacker always follows the target.*

Assumption 4. *The discrete process and measurement noises $q(k)$ and $\mu(k)$ are additive, and zero mean white Gaussian.*

Assumption 5. *The target-defender team has the capability to measure the ranges and the LOS angles defined in the inertial frame shown in Fig. 2.2.*

2.2.2 System model

The equations of motion for the agents are given as [126]

$$\dot{x}_A(t) = v_A \cos\alpha_A(t), \tag{2.1}$$

$$\dot{y}_A(t) = v_A \sin\alpha_A(t), \tag{2.2}$$

$$\dot{x}_T(t) = v_T(t) \cos\alpha_T(t), \tag{2.3}$$

$$\dot{y}_T(t) = v_T(t) \sin\alpha_T(t), \tag{2.4}$$

$$\dot{x}_D(t) = v_D \cos\alpha_D(t), \tag{2.5}$$

$$\dot{y}_D(t) = v_D \sin\alpha_D(t), \tag{2.6}$$

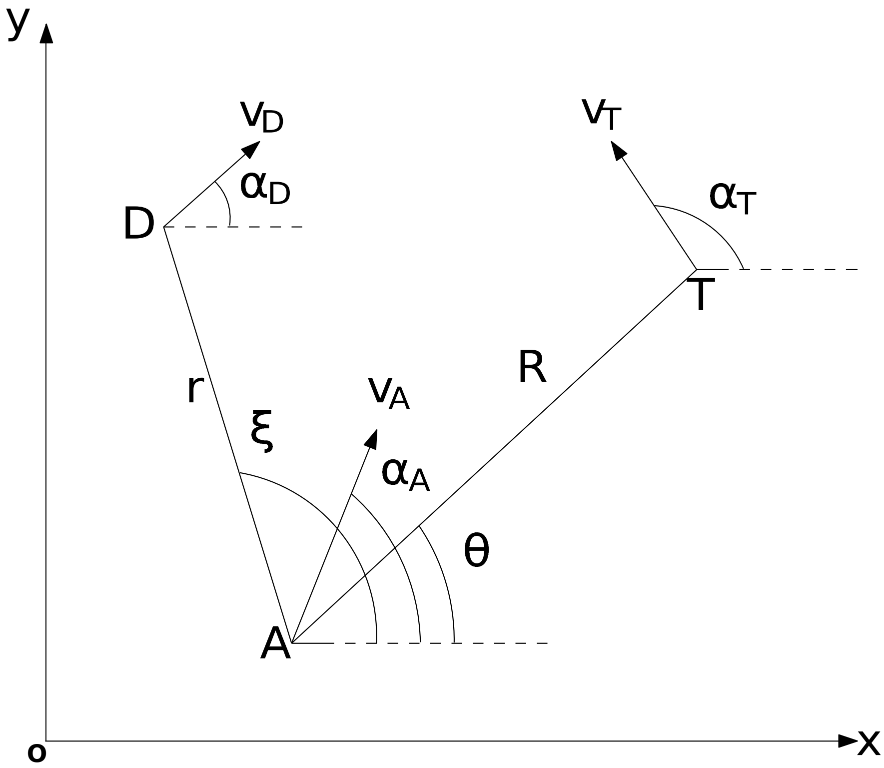


Figure 2.2: Attacker-Target-Defender engagement geometry for a three-agent problem.

where $x_A(t), y_A(t), x_T(t), y_T(t), x_D(t), y_D(t)$ are the x and y positions of the attacker, the target, and the defender respectively. $v_A,\ v_T\left(t\right)$ and v_D are the velocities of the attacker, the target, and the defender, respectively. $v_A,\ v_D$ are assumed to be constant, and $v_T\left(t\right)$ is bounded by $v_T\left(t\right) \in [0, \bar{v}_T]$. Similarly, $\alpha_A\left(t\right),\ \alpha_T\left(t\right)$ and $\alpha_D\left(t\right)$ are the heading angles of the attacker, the target, and the defender, respectively. All the states, heading angles of the agents, and the velocity of the target are changing with respect to time t, and the notation (t) is omitted in the rest of the document for simplicity. Previous studies [41, 46] assume that the attacker uses a predefined guidance law. However, an intelligent attacker can use unknown guidance laws, and hence, we assume that the exact attacker guidance law is not known to the target-defender team.

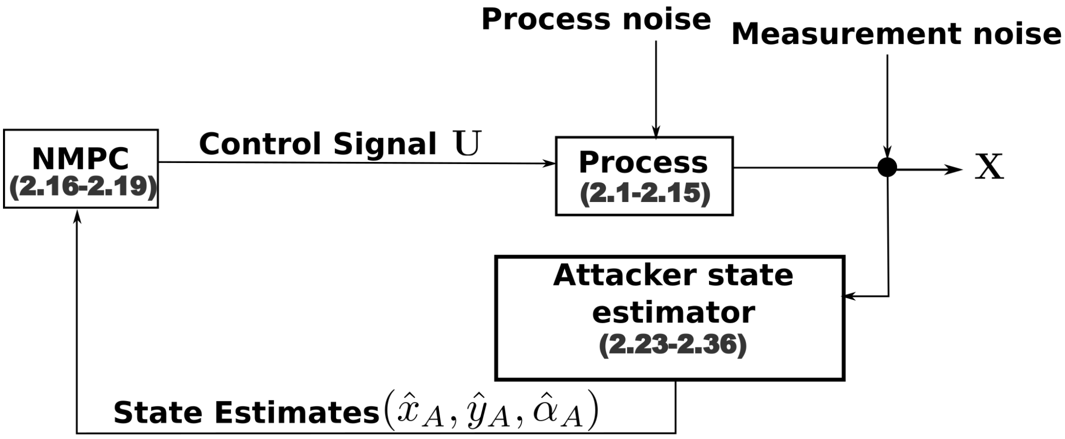


Figure 2.3: NMPC scheme for determining the control commands for the three-agent problem.

2.2.3 Engagement Geometry

Consider the three-agent engagement geometry shown in Fig. 2.2. The relative motion of the three agents can be modeled as [126]

$$
\begin{aligned}
v_R &= \dot{R} = v_T \cos(\alpha_T - \theta) - v_A \cos(\alpha_A - \theta), && (2.7)\\
v_\theta &= R\dot{\theta} = v_T \sin(\alpha_T - \theta) - v_A \sin(\alpha_A - \theta), && (2.8)\\
v_r &= \dot{r} = v_D \cos(\alpha_D - \xi) - v_A \cos(\alpha_A - \xi), && (2.9)\\
v_\xi &= r\dot{\xi} = v_D \sin(\alpha_D - \xi) - v_A \sin(\alpha_A - \xi), && (2.10)\\
R &= \sqrt{(x_T - x_A)^2 + (y_T - y_A)^2}, && (2.11)\\
\theta &= \arctan\left(\frac{y_T - y_A}{x_T - x_A}\right), && (2.12)\\
r &= \sqrt{(x_D - x_A)^2 + (y_D - y_A)^2}, && (2.13)\\
\xi &= \arctan\left(\frac{y_D - y_A}{x_D - x_A}\right), && (2.14)\\
\dot{\alpha}_A &= N\dot{\theta}, && (2.15)
\end{aligned}
$$

where θ is the line-of-sight (LOS) angle between the target and the attacker, ξ is the LOS angle between the defender and the attacker, R is the distance between the attacker and the target, and r is the distance between the defender and the attacker. v_R and v_θ are the components of the relative velocity along the $A - T$ LOS and perpendicular to the $A - T$ LOS. v_r and v_ξ are the components of the relative velocity along the $A - D$ LOS and perpendicular to the $A - D$ LOS. The attacker model is defined using proportional navigation (PN) guidance law with navigation constant N.

2.3 Nonlinear model predictive control design

Nonlinear model predictive control (NMPC) is based on determining optimal future actions of an agent or a system for a finite time horizon. The main component of the NMPC framework is the mathematical model of the three-agent system (2.17). For each time step t, the optimal control inputs are computed using this model for the horizon $[t, t+\tau_h]$, where τ_h is the specified look-ahead window. After applying the first value from the computed control sequence to the real system, the resulting output from the system is feed-backed to the controller, and the process is continued [127]. We propose to use NMPC to compute the cooperative control commands for the target-defender team. The attacker states are estimated with the help of measurements available to the $T-D$ team. Fig. 2.3 shows the proposed NMPC strategy, which also includes the state estimator for the attacker in the feedback loop in addition to the general NMPC structure. The NMPC block is the core component of the scheme, which contains the mathematical model of the system and a numerical optimizer. Process block represents the system involving the agents, and the estimator block contains the EKF.

2.3.1 Objective function

We propose a strategy wherein the target-defender team uses NMPC to compute the control commands so that the objective of target evasion from the attacker while the defender intercepts the attacker is achieved. We consider two types of target maneuvers: 1) constant speed target with only heading change, and 2) variable velocity target. The variable velocity target implies that the target can move or stop based on predefined conditions.

The NMPC problem can be stated as follows:

$$\begin{aligned} &\min_{u_x, u_y, \dot{\alpha}_D \in \mathcal{PC}(t, t+\tau_h)} J(X,U,t), \\ &J(X,U,t) = \int_t^{t+\tau_h} \left[\left(u_x^2 + u_y^2\right) + r + \max(0, e-R)\right] \mathrm{d}t, \end{aligned} \tag{2.16}$$

subject to:

$$\dot{X} = f(X,U,t), \tag{2.17}$$

$$U \in \left[U^-, U^+\right], \tag{2.18}$$

$$\sqrt{u_x^2 + u_y^2} \leq \bar{v}_T, \tag{2.19}$$

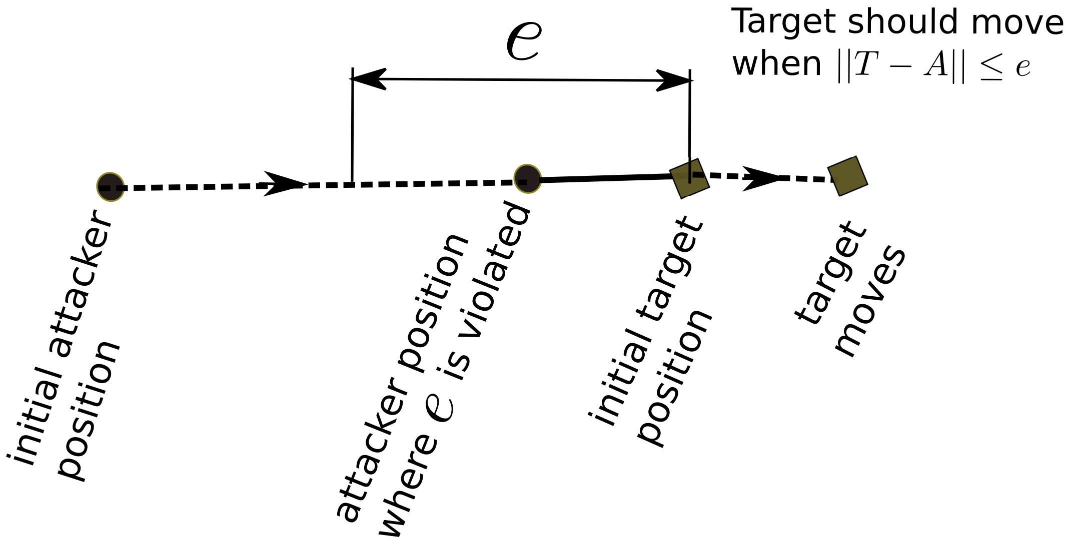


Figure 2.4: Definition of the safe distance of the target from the attacker.

where $X = \{x_A, y_A, x_T, y_T, x_D, y_D, \alpha_A, \alpha_T, \alpha_D\}$ are the states, $U = \{u_x, u_y, \dot{\alpha}_D\}$ are the control inputs, U^- and U^+ are the lower and upper bounds of U, $\mathcal{PC}(t, t+\tau_h)$ denotes the space of piece-wise continuous function defined over the time interval $[t, t+\tau_h]$, and the target velocity is bounded by $\bar{v}_T$. In NMPC, we need to apply the first value of the computed control sequence to the system. Since there is no notion of a 'first value' for a continuous signal, the assumption of piece-wise continuity for the control inputs is taken. We define a threshold, e, which is the safe distance of the target from the attacker as shown in Fig. 2.4. The target should move only if this safe distance is violated. For including the variable velocity characteristic of the target, the state equations (2.3) and (2.4) are modified as

$$\dot{x}_T = u_x, \tag{2.20}$$

$$\dot{y}_T = u_y, \tag{2.21}$$

where u_x, u_y are the velocity components of the target in x and y directions, respectively, which will also act as the control inputs for the target. The target heading angle α_T can be written in terms of the control inputs as

$$\alpha_T = \arctan\left(\frac{u_y}{u_x}\right). \tag{2.22}$$

The attacker states are unknown, and we estimate these states using an extended Kalman filter (EKF).

2.3.2 Estimation of the attacker states using EKF

An extended Kalman filter (EKF) is used to estimate the attacker's position and course angle. The EKF is formulated as [128],

Model

$$X_A(k) = f_A\left(X_A(k-1), k\right) + q(k), \tag{2.23}$$

$$z(k) = h\left(X_A(k), k\right) + \mu(k), \tag{2.24}$$

Prediction

$$\begin{aligned} X_A(k|k-1) = f_A(X_A(k-1|k-1), k)\Delta t \\ + X_A(k-1|k-1), \end{aligned} \tag{2.25}$$

$$P(k|k-1) = \nabla F_{X_A} P(k-1|k-1) \nabla F'_{X_A} + Q, \tag{2.26}$$

$$z(k|k-1) = h(X_A(k|k-1)), \tag{2.27}$$

Update

$$X_A(k|k) = X_A(k|k-1) + K(k)\nu(k), \tag{2.28}$$

$$P(k|k) = P(k|k-1) - K(k)S(k)K'(k), \tag{2.29}$$

$$\nu(k) = z(k) - z(k|k-1), \tag{2.30}$$

$$K(k) = P(k|k-1)\nabla H'_{X_A} S^{-1}(k), \tag{2.31}$$

$$S(k) = \nabla H_{X_A} P(k|k-1) \nabla H'_{X_A} + \Sigma, \tag{2.32}$$

where, $X_A(k)$ and $z(k)$ represent the attacker state model and measurement model respectively, and $P(k)$ is the covariance matrix. $q(k)$ and $\mu(k)$ are the process and measurement noises with covariance Q and Σ respectively. $\nu(k)$ is called as the innovation parameter, and $K(k)$ is the Kalman gain. The attacker states $\{x_A, y_A, \alpha_A\}$ and control a_A need to be estimated, hence the prediction model is defined as:

$$f_A = \begin{bmatrix} \dot{x}_A \\ \dot{y}_A \\ \dot{\alpha}_A \\ \dot{a}_A \end{bmatrix} = \begin{bmatrix} v_A \cos\alpha_A \\ v_A \sin\alpha_A \\ \frac{a_A}{v_A} \\ -a_A \end{bmatrix}, \tag{2.33}$$

and the Jacobian of f_A is

$$\nabla F_{X_A} = \begin{bmatrix} 0 & 0 & -v_A \sin \alpha_A & 0 \\ 0 & 0 & v_A \cos \alpha_A & 0 \\ 0 & 0 & 0 & \frac{1}{v_A} \\ 0 & 0 & 0 & -1 \end{bmatrix}. \tag{2.34}$$

We model the attacker control input similar to [129, 130], where estimated target acceleration was treated as an unknown input to the nonlinear 2D missile-target engagement system. According to Assumption 5, the measurement model for R, r, θ and ξ is given as

$$h = \begin{bmatrix} \sqrt{(x_T - x_A)^2 + (y_T - y_A)^2} \\ \sqrt{(x_D - x_A)^2 + (y_D - y_A)^2} \\ \tan^{-1}\left(\frac{y_T - y_A}{x_T - x_A}\right) \\ \tan^{-1}\left(\frac{y_D - y_A}{x_D - x_A}\right) \end{bmatrix}. \tag{2.35}$$

The Jacobian of the measurement model is given by

$$\nabla H_{X_A} = \begin{bmatrix} \frac{-(x_T - x_A)}{\sqrt{(x_T - x_A)^2 + (y_T - y_A)^2}} & \frac{-(y_T - y_A)}{\sqrt{(x_T - x_A)^2 + (y_T - y_A)^2}} & 0 & 0 \\ \frac{-(x_D - x_A)}{\sqrt{(x_D - x_A)^2 + (y_D - y_A)^2}} & \frac{-(y_D - y_A)}{\sqrt{(x_D - x_A)^2 + (y_D - y_A)^2}} & 0 & 0 \\ \frac{y_T - y_A}{(x_T - x_A)^2 + (y_T - y_A)^2} & \frac{-(x_T - x_A)}{(x_T - x_A)^2 + (y_T - y_A)^2} & 0 & 0 \\ \frac{y_D - y_A}{(x_D - x_A)^2 + (y_D - y_A)^2} & \frac{-(x_D - x_A)}{(x_D - x_A)^2 + (y_D - y_A)^2} & 0 & 0 \end{bmatrix}. \tag{2.36}$$

Note that if the EKF estimates become unbounded, the NMPC solution will get unbounded as well. A solution to keep the estimation errors bounded is to add extra constraints to the NMPC formulation to ensure the covariance of the EKF is bounded by satisfying the observability conditions [90, 131, 132]. However, it will 'degrade' the control to obtain more accurate state estimates, and a balance between the different objectives would be required. Here, we assume that the system is always observable and the measurements are available irrespective of the positions of the agents (no limitations on the sensor range). Hence, we do not provide any additional constraints to ensure system stability.

2.4 Escape region

The ability to capture or escape depends on the initial conditions of the engagement geometry. We present an analysis based on the Apollonius circle concept to determine the escape and capture zones for the target-defender team. For the analysis, we assume that the agents are initially pointing in the right direction, and they take optimal straight-line trajectories. Initially, we will assume that the attacker–defender agents have equal speed and then relax with the assumption to show how the zones are modified.

2.4.1 Assumptions

Assumption 6. *The target is slower than the attacker. Otherwise, the target will always evade the attacker, and the defender's role will become insignificant.*

Assumption 7. *The speed of the defender is greater than or equal to the speed of the target. This is a valid assumption since, in real-world scenarios, the target is usually a slow-moving aircraft and the defender is a missile.*

2.4.2 Constant speed target

We consider a modified reference frame as shown in Fig. 2.5, where T, A, D represent the positions of the target, the attacker, and the defender, respectively. The $x-$axis is defined as the line joining the attacker and the defender, A and D. The $y-$axis is defined as the perpendicular bisector of the line segment $\overline{AD}$.

Definition 1. *Apollonius circle: the concept of the Apollonius circle in pursuit-evasion games was initially proposed by Isaacs [36]. Consider a pursuer and an evader with the motion kinematics similar to the one defined in* (2.1),(2.2) *in the Cartesian plane. The Apollonius circle for the pursuer-evader engagement is the locus of points that take equal time for the evader and the pursuer to reach.*

Lemma 1. *The Apollonius circle for the modified reference frame of the $A - D$ engagement geometry is the $y-$axis [36].*

Proof. The modified reference frame for the $A - D$ engagement is shown in Fig. 2.5. The center of the Apollonius circle formed by $A - D$ is

$$(x_c, y_c)_{AD} = \left(\frac{x_D - \gamma_{AD}^2 x_A}{1 - \gamma_{AD}^2}, \frac{y_D - \gamma_{AD}^2 y_A}{1 - \gamma_{AD}^2} \right), \tag{2.37}$$

where γ_{AD} is the speed ratio of the defender to the attacker, $\gamma_{AD} = \frac{v_D}{v_A}$. Since the $x-$axis is defined as

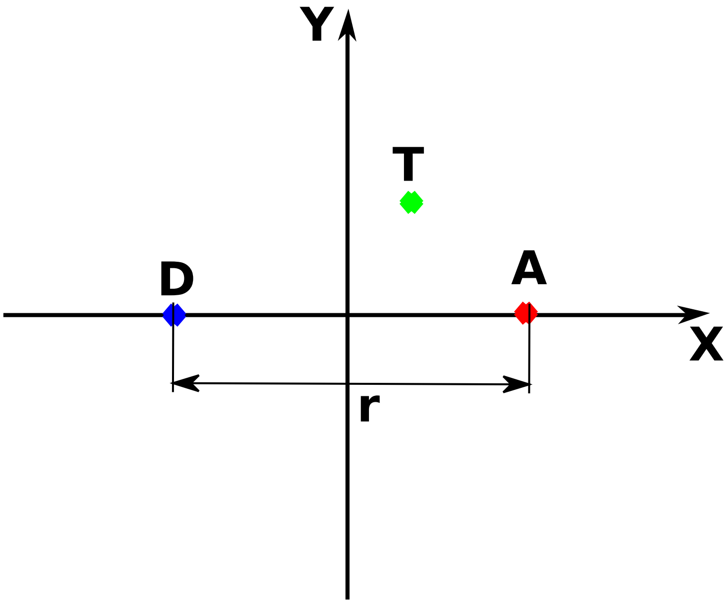


Figure 2.5: Modified TAD reference frame.

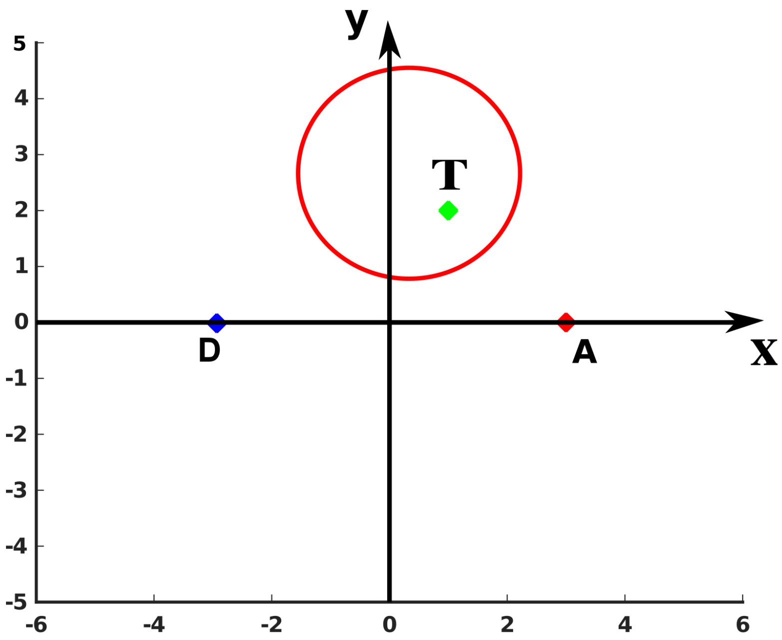


Figure 2.6: Example Apollonius circle for $A - T$ engagement.

the line joining A and D, y_A and y_D will always be zero. Therefore, equation (2.37) becomes

$$(x_c, y_c)_{AD} = \left(\frac{x_D - \gamma_{AD}^2 x_A}{1 - \gamma_{AD}^2}, 0\right). \tag{2.38}$$

The radius of the Apollonius circle for $A - D$ engagement geometry is given by

$$r_{AD} = \frac{\gamma_{AD}\left(x_D - x_A\right)}{1 - \gamma_{AD}^2}. \tag{2.39}$$

Since the attacker and the defender have equal speed, $\gamma_{AD} = 1$. According to equations (2.38)-(2.39), the Apollonius circle for the $A - D$ case will become the $y-$axis in Fig. 2.5. □

Lemma 2. *The defender would be able to intercept the attacker only if the $A - T$ circle intercepts the $y-$axis.*

Proof. Consider the Apollonius circle of $A - T$ engagement. The center and radius of the Apollonius circle for $A - T$ engagement geometry are given as

$$(x_c, y_c)_{AT} = \left(\frac{x_T - \gamma_{AT}^2 x_A}{1 - \gamma_{AT}^2}, \frac{y_T}{1 - \gamma_{AT}^2}\right), \tag{2.40}$$

and

$$r_{AT} = \frac{\gamma_{AT}\sqrt{\left(x_T - x_A\right)^2 + y_T^2}}{1 - \gamma_{AT}^2}. \tag{2.41}$$

Fig. 2.6 represents the Apollonius circle for the $A - T$ engagement. The circle represents the points at which the attacker and the target can reach simultaneously, leading to the target capture. Similarly, the $A - D$ circle represents the points the attacker and the defender can reach simultaneously. If the $A - D$ circle does not have an intersection point with the $A - T$ circle, it means that the attacker can move on to points the defender can not reach and safely capture the target. If there is a common intersection point between $A - T$ and $A - D$ circles, then the target can lure the attacker to that common point, and the defender would be able to intercept the attacker. Since the $y-$axis represents the $A - D$ Apollonius circle, the defender would be able to intercept the attacker only if the $A - T$ circle intercepts the $y-$axis. □

It is possible to map the target escape zone by examining if the condition given in Lemma. 2 is satisfied or not. In the following theorem, we find the expression for the locus of points that divides the Cartesian plane into the target escape and capture zones.

Theorem 1. *The escape region for the target can be represented by a curve given by*

$$\frac{x^2}{\gamma_{AT}^2 x_A^2} - \frac{y^2}{\left(1-\gamma_{AT}^2\right) x_A^2} = 1, \tag{2.42}$$

that divides the Cartesian plane into two zones, Z_e and Z_c, where Z_e is the escape zone for the target and Z_c is the capture zone for the target [42].

Proof. The defender would be able to intercept the attacker before it captures the target only if the $A-T$ Apollonius circle intercepts the $A-D$ Apollonius circle, which is the $y-$axis, according to Lemma. 1 and Lemma. 2. The radius of the $A-T$ circle is given by (2.41), and the $x-$coordinate of its center is

$$x_{c_{AT}} = \frac{1}{1-\gamma_{AT}^2}\left(x_T - \gamma_{AT}^2 x_A\right). \tag{2.43}$$

For the $A-T$ circle to intercept the $y-$axis, r_{AT} should be greater than $x_{c_{AT}}$.

$$\frac{\gamma_{AT}\sqrt{\left(x_T-x_A\right)^2+y_T^2}}{1-\gamma_{AT}^2} > \frac{1}{1-\gamma_{AT}^2}\left(x_T - \gamma_{AT}^2 x_A\right), \tag{2.44}$$

$$x_T - \gamma_{AT}^2 x_A < \gamma_{AT}\sqrt{\left(x_T-x_A\right)^2+y_T^2}, \tag{2.45}$$

$$\frac{x_A^2}{\left(\frac{x_T}{\gamma_{AT}}\right)^2} + \frac{y_T^2}{\left(\frac{\sqrt{1-\gamma_{AT}^2}}{\gamma_{AT}}x_T\right)^2} > 1. \tag{2.46}$$

To obtain the curve that defines the escape zone, we equate (2.46) to 1. The modified equation is

$$\frac{x_A^2}{\left(\frac{x_T}{\gamma_{AT}}\right)^2} + \frac{y_T^2}{\left(\frac{\sqrt{1-\gamma_{AT}^2}}{\gamma_{AT}}x_T\right)^2} = 1, \tag{2.47}$$

substituting x_T, y_T with x, y, we get the equation for the locus of points that divides the plane as

$$\frac{x_A^2}{\left(\frac{x}{\gamma_{AT}}\right)^2} + \frac{y^2}{\left(\frac{\sqrt{1-\gamma_{AT}^2}}{\gamma_{AT}}x\right)^2} = 1, \tag{2.48}$$

which can be rearranged into

$$\frac{x^2}{\gamma_{AT}^2 x_A^2} - \frac{y^2}{\left(1-\gamma_{AT}^2\right) x_A^2} = 1. \tag{2.49}$$

□

Fig. 2.7 shows the curve dividing the plane into two zones. If the target's initial position lies in the

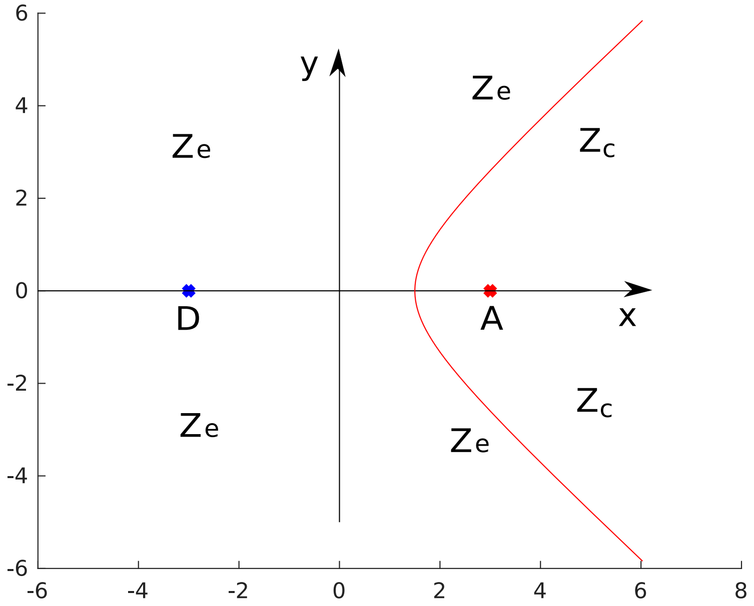


Figure 2.7: Escape zone for the constant speed target, T. Z_e is the escape zone and Z_c is the capture zone.

escape zone Z_e, the $A-T$ Apollonius circle will intercept the $y-$axis, and hence, the target can lure the attacker into crossing the $y-$axis and allow the defender to intercept it. If the target lies in the capture zone Z_c, it would be captured before crossing the $y-$axis. Fig. 2.8 shows the family of dividing curves for different values of γ_{AT}, $0 < \gamma_{AT} < 1$. It can be seen that as the speed ratio γ_{AT} increases, the area of the target escape zone increases since a target with higher velocity has more chance of escaping the attacker.

2.4.3 Stationary target

In the case of a stationary target that does not move throughout the engagement, the $A-T$ speed ratio $\gamma_{AT} = 0$. Hence, the center and radius of the $A-T$ Apollonius circle given by equations (2.40) and (2.41) can be modified as

$$(x_c, y_c)_{AT} = (x_T, y_T)\,, \tag{2.50}$$

and

$$r_{AT} = 0, \tag{2.51}$$

which means the Apollonius circle will shrink to become the target point itself. According to Lemma. 2, the target would be able to escape only if the $A-T$ Apollonius circle intercepts the $y-$axis. Hence,

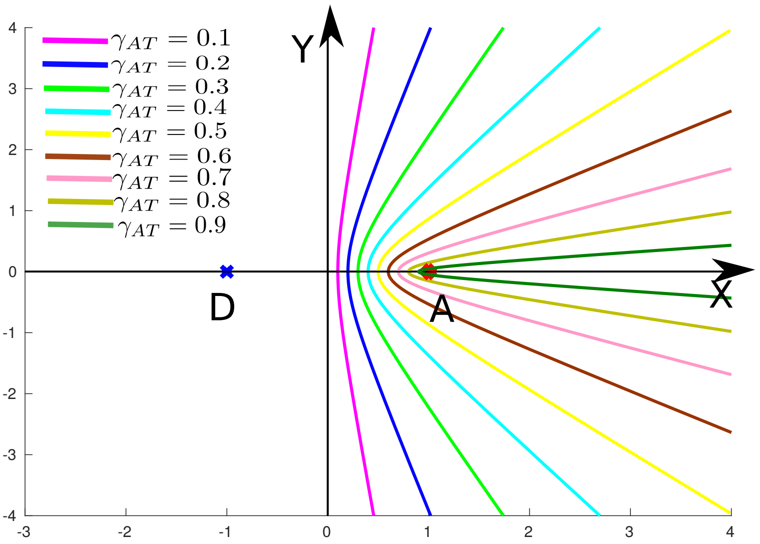


Figure 2.8: Family of dividing curves for γ_{AT} = 0.1,. . . ,0.9.

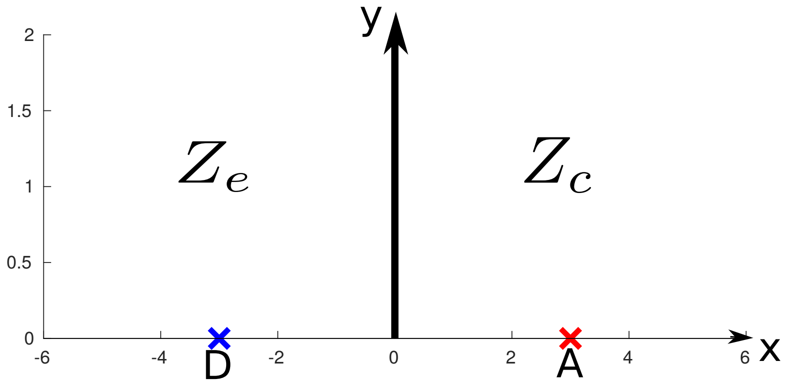


Figure 2.9: Escape zone for the stationary target, T. Z_e is the escape zone and Z_c is the capture zone.

applying the condition for the $A-T$ circle to intercept the $y-$axis, which is $r_{AT} > x_{c_{AT}}$, we get

$$0 > x_T, \tag{2.52}$$

and after substituting x for x_T and changing the inequality sign to equality for obtaining the dividing curve, we get

$$x = 0, \tag{2.53}$$

which is the equation for the $y-$axis. Hence, for the stationary target case, the curve that divides the space into the escape and the capture zones is the $y-$axis itself, as shown in Fig. 2.9.

2.4.4 Variable target velocity

For analyzing this case, consider Fig. 2.10a. The target, attacker, and defender initial positions are given as (x_A, y_A), (x_D, y_D), and (x_T, y_T). The target is in motion only when the attacker violates the safe distance e. Let (x_A^e, y_A^e) and (x_D^e, y_D^e) be the coordinates of the attacker and the defender when $||T - A|| = e$. We now create a new reference frame $x' - y'$ as shown in the figure, where

$$\phi = \arctan\left(\frac{y_A^e - y_D^e}{x_A^e - x_D^e}\right), \text{and} \tag{2.54}$$

$$x_0, y_0 = \left(\frac{x_A^e + x_D^e}{2}, \frac{y_A^e + y_D^e}{2}\right). \tag{2.55}$$

The coordinates from the $x - y$ frame can be transformed into $x' - y'$ frame as follows

$$x_A' = (x_A^e - x_0)\cos\phi + (y_A^e - y_0)\sin\phi, \tag{2.56}$$

$$y_A' = (y_A^e - y_0)\cos\phi - (x_A^e - x_0)\sin\phi, \tag{2.57}$$

$$x_D' = (x_D^e - x_0)\cos\phi + (y_D^e - y_0)\sin\phi, \tag{2.58}$$

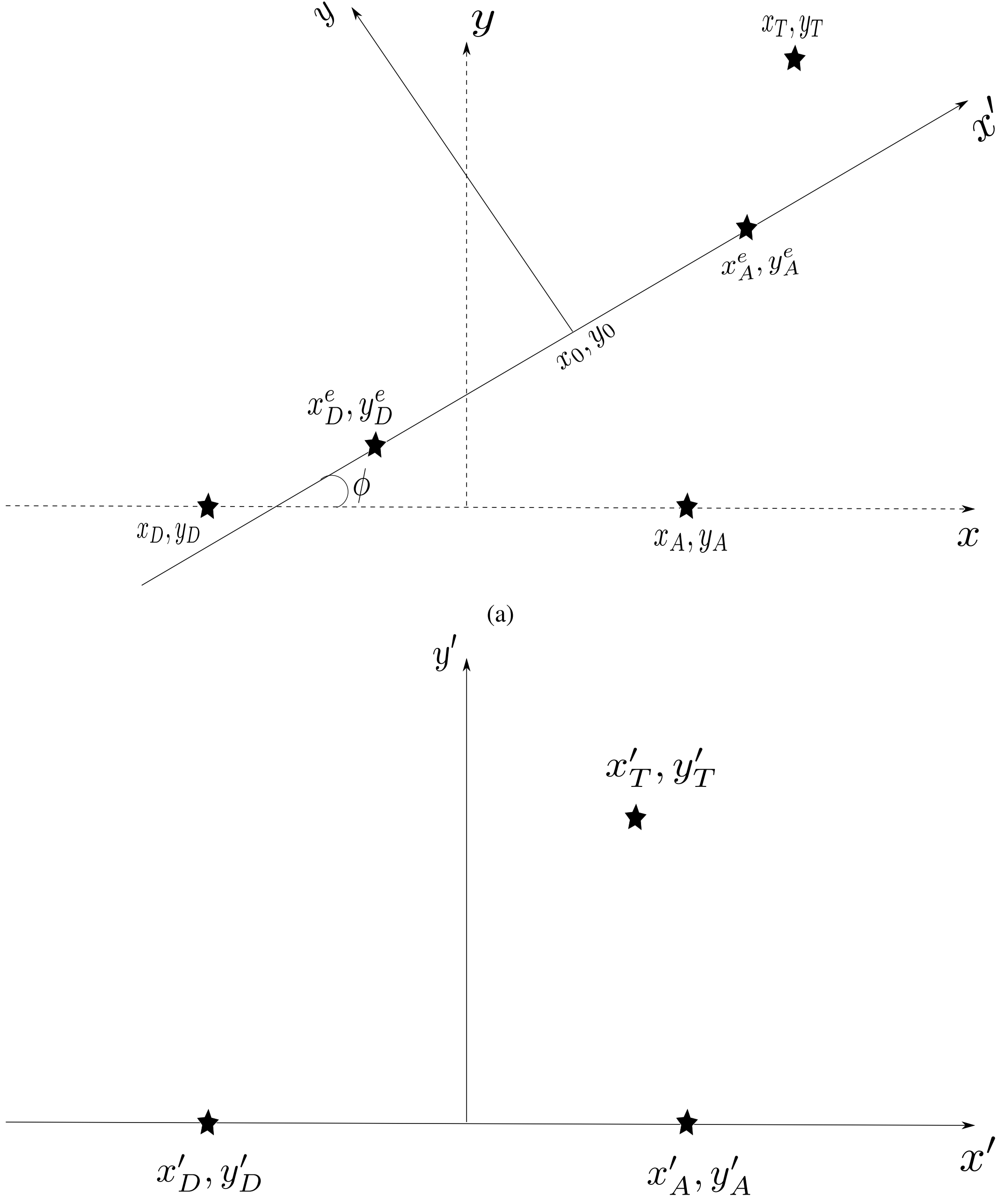
$$y_D' = (y_D^e - y_0)\cos\phi - (x_D^e - x_0)\sin\phi, \tag{2.59}$$

$$x_T' = (x_T - x_0)\cos\phi + (y_T - y_0)\sin\phi, \tag{2.60}$$

$$y_T' = (y_T - y_0)\cos\phi - (x_T - x_0)\sin\phi. \tag{2.61}$$

Fig. 2.10b shows the agent representations in the re-defined $x' - y'$ frame.

Consider an example case where the initial conditions are selected as $(x_A, y_A) = (0.5, 0)$, $(x_D, y_D) = (-0.5, 0)$, $\gamma_{AT} = 0.5$, and $e = \frac{1}{2}R$. From equation (2.46), the escape zone for the target is given in Fig. 2.11. The variable target velocity case can be subdivided into three sub-cases, (i) if the target is always moving with the maximum allowed speed as given in Sec. 2.4.2. In this case, the dashed curve represents the division of the plane into the escape and the capture zones. (ii) If the target is always stationary as given in Sec. 2.4.3, the division will be represented by the $y-$axis. The target escape zone will be left of the $y-$axis, and the capture zone will be on the right side of the $y-$axis. This is due to the fact that the interception of the attacker by the defender can only occur at the $y-$axis, and if the target does not move, it has to be on the left side of the $y-$axis in order to lure the attacker into crossing the $y-$axis so that the defender can intercept it. (iii) If the target is initially at rest and starts moving with maximum velocity after the safe distance e is violated. For this case, the solid curve represents the division of the space into the two zones, the target escape zone, Z_e', and the target capture zone, Z_c'. After the target



(a)

(b)

Figure 2.10: Configurations in a new reference frame for the variable velocity target. (a) Modified reference frame (figure not to scale). (b) Agent representations in the new $x' - y'$ frame.

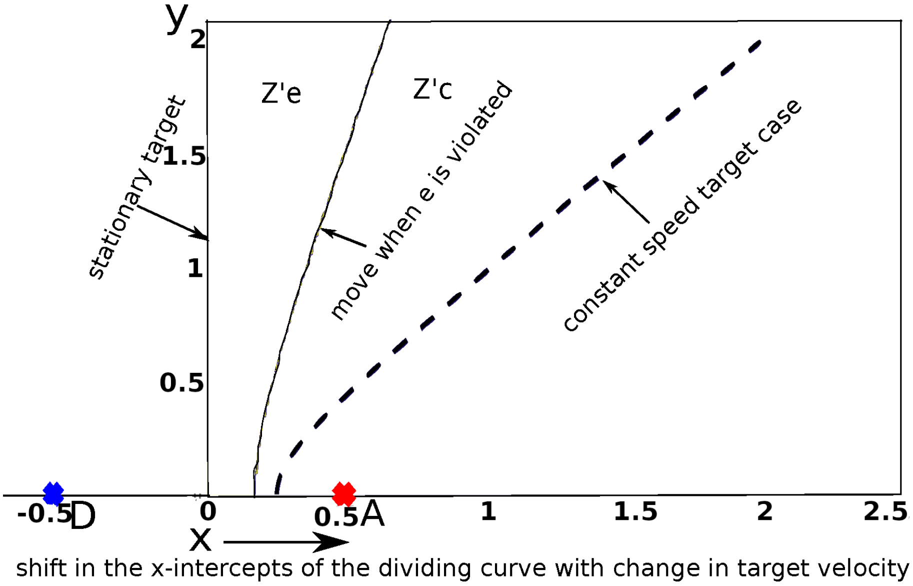


Figure 2.11: Escape zone for the variable velocity target, T. The dashed curve represents the constant speed case, the solid curve the variable velocity case, and the $y-$axis represents the boundary for the stationary target.

starts moving, the engagement is similar to the constant speed target case given in Sec. 2.4.2, and the conditions given in Lemma. 2 can be used to map the escape region for the target.

Deriving an analytical expression for this solid curve is very difficult, unlike the constant speed case, since at each initial target position, a new reference frame needs to be formed, which depends on the instantaneous position of the defender (x_D^e, y_D^e), which is the solution of a differential equation. Hence, the escape and the capture zone are obtained through numerical analysis for the variable target velocity case. For each initial position of the target in the $x - y$ frame, the coordinates for the origin and the rotation of the $x' - y'$ frame are calculated, and the agents are represented in the $x' - y'$ frame. After that, the condition (2.46) is used in the $x' - y'$ frame to check if the target will escape or not.

2.4.5 Different speeds for attacker and defender

Now, we relax the assumption that the attacker and the defender have equal speed and perform the analysis for cases where $\gamma_{AD} > 0, \gamma_{AD} \neq 1$.

2.4.5.1 Constant speed target case

Extending Lemma 2 for general cases, we can see that the curve that separates the escape region from the capture region is the locus of points where the $A-T$ Apollonius circle is tangential to the $A-D$ Apollonius circle. This condition can be written as

$$d_c = r_{AD} + r_{AT}, \tag{2.62}$$

where d_c is the distance between the centers of the $A-D$ and $A-T$ Apollonius circles, r_{AD} is the radius of the $A-D$ circle, and r_{AT} is the radius of the $A-T$ circle respectively. Now, we substitute and expand the terms in equation (2.62) as follows

$$\sqrt{(x_{c_{AD}} - x_{c_{AT}})^2 + (y_{c_{AD}} - y_{c_{AT}})^2} = \frac{\gamma_{AD}(x_D - x_A)}{1-\gamma_{AD}^2} + \frac{\gamma_{AT}\sqrt{(x_T - x_A)^2 + y_T^2}}{1-\gamma_{AT}^2}, \tag{2.63}$$

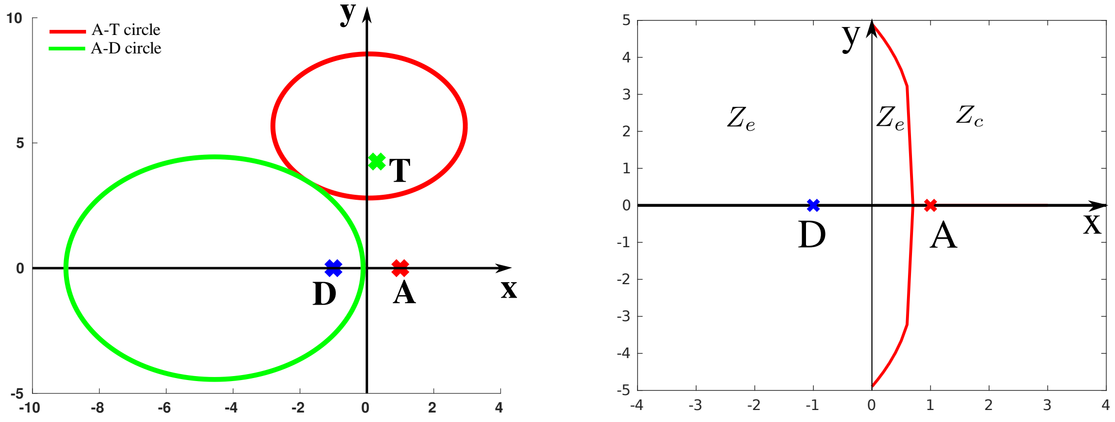
$$\begin{aligned}\left(\frac{x_D - \gamma_{AD}^2 x_A}{1-\gamma_{AD}^2} - \frac{x_T - \gamma_{AT}^2 x_A}{1-\gamma_{AT}^2}\right)^2 + \left(\frac{y_T}{1-\gamma_{AT}^2}\right)^2 = \\ \left(\frac{\gamma_{AD}(x_D - x_A)}{1-\gamma_{AD}^2} + \frac{\gamma_{AT}\sqrt{(x_T - x_A)^2 + y_T^2}}{1-\gamma_{AT}^2}\right)^2.\end{aligned} \tag{2.64}$$

After some algebraic manipulations and substituting x, y for x_T, y_T, and $x_D = -x_A$, we get the following quartic equation

$$\begin{aligned}c_3^2 y^4 + (2c_2c_3x^2 + 2c_3c_4x + 2c_3c_0 - c_1^2)y^2 + c_2^2x^4 + c_4^2x^2 + c_0^2 + 2c_4c_0x \\ + 2c_2c_4x^3 + 2c_2c_0x^2 - c_1^2x^2 + 2c_1^2x_Ax - c_1^2x_A^2 = 0,\end{aligned} \tag{2.65}$$

where

$$\begin{aligned}c_0 &= x_A^2 - 2\gamma_{AD}^2x_A^2 + \gamma_{AD}^4x_A^2 - 5\gamma_{AT}^2x_A^2 + 8\gamma_{AT}^2\gamma_{AD}^2x_A^2 - \\ &\quad \gamma_{AD}^4\gamma_{AT}^2x_A^2 + 4\gamma_{AT}^4x_A^2 - 4\gamma_{AT}^4\gamma_{AD}^2x_A^2 - 2\gamma_{AD}^2\gamma_{AT}^2x_A, \\ c_1 &= -4\gamma_{AT}\gamma_{AD}x_A + 4\gamma_{AT}^3\gamma_{AD}x_A + 4\gamma_{AT}\gamma_{AD}^3x_A - \\ &\quad 4\gamma_{AD}^3\gamma_{AT}^3x_A, \\ c_2 &= 1 - \gamma_{AT}^2 - 2\gamma_{AD}^2 + 2\gamma_{AT}^2\gamma_{AT}^2 + \gamma_{AD}^4 - \gamma_{AD}^4\gamma_{AT}^2, \\ c_3 &= 1 - \gamma_{AT}^2 - 2\gamma_{AD}^2 + 2\gamma_{AD}^2\gamma_{AT}^2 + \gamma_{AD}^4 - \gamma_{AD}^4\gamma_{AT}^2, \\ c_4 &= 2x_A - 2\gamma_{AT}^2x_A - 2\gamma_{AD}^4x_A + 2\gamma_{AT}^2\gamma_{AD}^4x_A.\end{aligned}$$



(a) (b)

Figure 2.12: Example plots of the escape region for a constant speed target when $\gamma_{AD} < 1$. (a) Apollonius circles of $A-D$ and $A-T$ engagements at the target escape boundary. (b) The curve that divides the plane into escape and capture regions for the target.

The solutions for the equation (2.65) is given by

$$y_1^2 = \frac{-b+\sqrt{b^2-4ac}}{2a}, \tag{2.66}$$

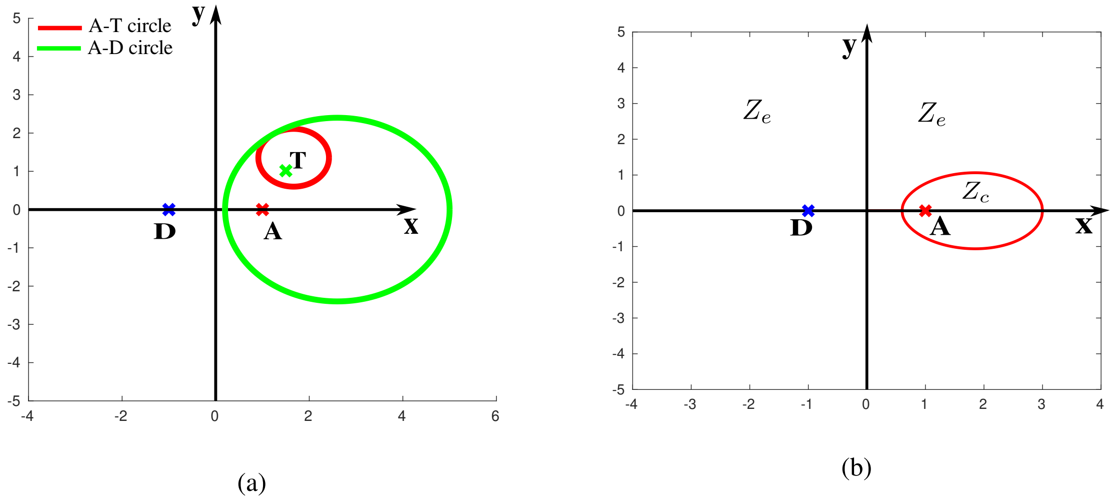
$$y_2^2 = \frac{-b-\sqrt{b^2-4ac}}{2a}, \tag{2.67}$$

where

$$\begin{aligned} a &= c_3^2, \\ b &= 2c_2c_3x^2 + 2c_3c_4x + 2c_3c_0 - c_1^2, \\ c &= c_2^2x^4 + c_4^2x^2 + c_0^2 + 2c_4c_0x + 2c_2c_4x^3 + 2c_2c_0x^2 \\ &\quad - c_1^2x^2 + 2c_1^2x_Ax - c_1^2x_A^2. \end{aligned}$$

When $\gamma_{AD} < 1$, the $A-T$ circle will be outer-tangential to the $A-D$ circle at the escape boundary of the target as shown in Fig. 2.12a, and the solution to be used is given by equation (2.66). An example curve that divides the $x-y$ plane into the target escape and capture zones is shown in Fig. 2.12b.

When $\gamma_{AD} > 1$, the $A-T$ circle will be inner-tangential to the $A-D$ circle at the escape boundary of the target as shown in Fig. 2.13a, and the solution to be used is given by equation (2.67). An example curve that divides the $x-y$ plane into the target escape and capture zones is shown in Fig. 2.13b. It can be seen that the target capture zone is reduced to a closed space beyond which the target escapes. This phenomenon is due to the speed advantage of the defender over the attacker.



(a)

(b)

Figure 2.13: Example plots of the escape region for a constant speed target when $\gamma_{AD} > 1$. (a) Apollonius circles of $A - D$ and $A - T$ engagements at the target escape boundary. (b) The curve that divides the plane into escape and capture regions for the target.

2.4.5.2 Stationary target case

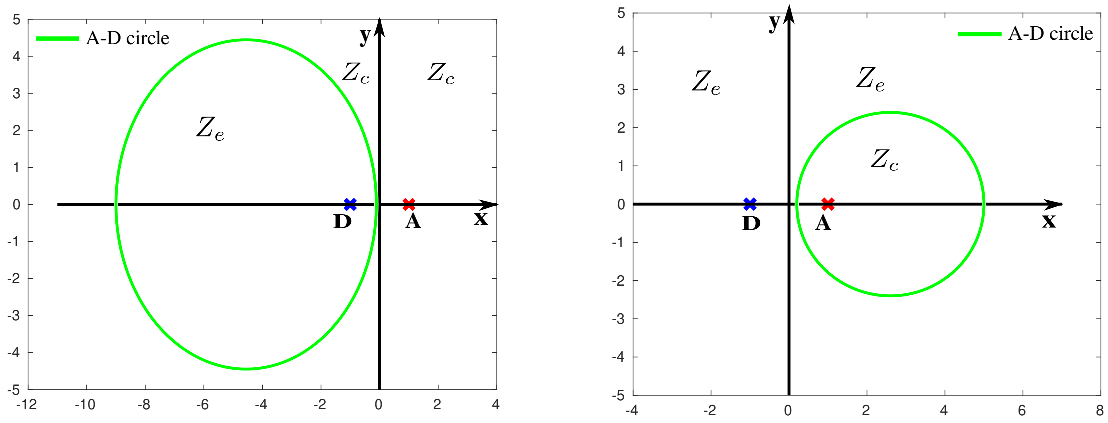
Since the $A - T$ Apollonius circle is the target point itself as given by equations (2.50) and (2.51), the curve that divides the plane into the target escape and capture zones will be the $A - D$ Apollonius circle. When $\gamma_{AD} < 1$, the target escape zone will be inside the $A - D$ circle, and when $\gamma_{AD} > 1$, the target escape zone will be outside the $A - D$ circle. An example for both cases is shown in Fig. 2.14.

2.4.5.3 Variable target velocity case

This case needs to be numerically computed similar to the analysis given in Sec. 2.4.4. The only difference here is that instead of checking if the $A - T$ Apollonius circle crosses the $y-$axis, we need to check if the $A - T$ circle crosses the $A - D$ circle. An example for the curve that divides the $x - y$ plane into the target escape and capture zones is shown in Fig. 2.15.

2.4.6 Escape region based on the estimated states (stochastic escape region)

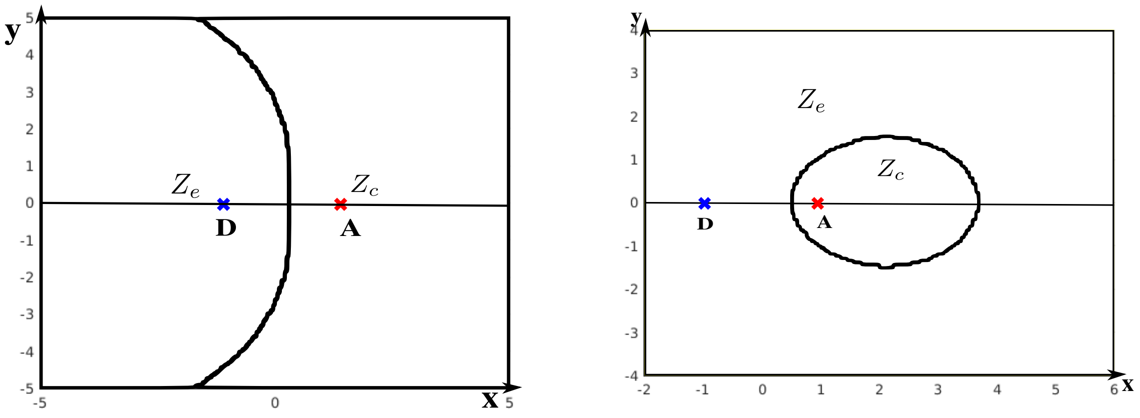
The analysis presented in Sec. 2.4.2 is based on ideal conditions with a deterministic system, and the true position values of the agents are known. Nevertheless, the analysis can easily be converted to include stochasticity and attacker state estimation. The EKF formulated in Sec. 2.3.2 allows the attacker states to be estimated precisely within the $\pm 3\sigma$ bounds. If the worst case (for the target) of -3σ is considered,



(a)

(b)

Figure 2.14: Example plots of the escape region for a stationary target when $\gamma_{AD} \neq 1$. The $A - D$ Apollonius circle divides the plane into the target escape zone, z_e and capture zone, z_c. (a) When $\gamma_{AD} < 1$ (b) when $\gamma_{AD} > 1$.



(a)

(b)

Figure 2.15: Example plots of the escape region for a variable velocity target when $\gamma_{AD} \neq 1$. (a) When $\gamma_{AD} < 1$ (b) when $\gamma_{AD} > 1$.

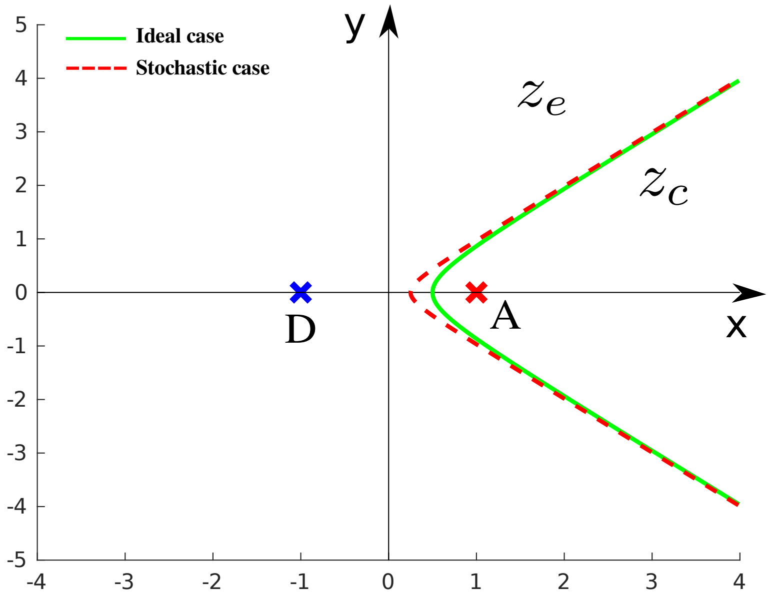


Figure 2.16: Stochastic escape zone for the constant speed target, T. Z_e is the escape zone and Z_c is the capture zone.

the dividing curve shifts a little to the left, reducing the escape zone of the target as shown in Fig. 2.16. A similar analysis can be performed on all the other target speed cases given in the thesis, and the corresponding results are not included here to avoid redundancy.

2.5 Simulation results

The efficacy of the proposed NMPC scheme is evaluated through extensive numerical simulations. The simulations were performed using MATLAB R2017b and VirtualArena [133] on an Ubuntu 16.04, Intel i5 PC with 8GB RAM running at 2.4GHz. Initially, we present the results for the constant speed target case followed by the variable target velocity case.

2.5.1 Simulation setup

The look-ahead window for the NMPC was selected as 6 steps to balance the computation time and the performance, with a sampling time of $0.05\,$s, which makes the prediction window $\tau_h = 0.3\,$s for all

constant speed target			variable velocity target			
parameter	escape	capture	parameter	escape e not violated	escape e violated	capture
(x_A, y_A) (m)	(50,50)	(50,50)	(x_A, y_A) (m)	(50,0)	(50,0)	(50,0)
(x_T, y_T) (m)	(25,30)	(60,130)	(x_T, y_T) (m)	(-50,100)	(10,100)	(100,100)
(x_D, y_D) (m)	(0,0)	(0,0)	(x_D, y_D) (m)	(-50,0)	(-50,0)	(-50,0)
α_A (rad)	-2.2	0.78	α_A (rad)	1.57	3.14	1.57
α_T (rad)	-2.2	0.78	u_x (m/s)	0	0	0
α_D (rad)	0.78	0.78	u_y (m/s)	0	0	0
v_A (m/s)	4	4	α_D (rad)	0.78	0	0
v_T (m/s)	2	2	v_A (m/s)	4	4	4
v_D (m/s)	4	4	v_D (m/s)	4	4	4

Table 2.1: Initial parameters for the agents in TAD simulation with equal speed for the attacker and the defender.

simulations. The covariance matrices Q and Σ for the EKF are selected as

$$Q = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}, \Sigma = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.01 \end{bmatrix}.$$

For the simulation purpose, the attacker switches the guidance law between pure pursuit (PP) and proportional navigation (PN) at each second. This technique is opted in order to show that the attacker is intelligent and also the efficacy of the proposed NMPC scheme against unknown attacker guidance laws. Note that the attacker guidance law is not known to the target-defender team. The PP guidance law is given as [134] $a_A = -\kappa\left(\alpha_A - \theta\right)$, and the PN guidance is given as [126] $a_A = N v_A \dot{\theta}$. The navigation constant, N = 3, is taken for the PN guidance law and κ = 2 for the PP law. $R_c = 1\,\text{m}$ and $r_c = 1\,\text{m}$ are the attacker-target and the attacker-defender capture radii. The angular velocity of the defender is constrained to $-0.5 \leq \dot{\alpha}_D \leq 0.5\,\text{rad/s}$ considering the real-world implementation constraints. The maximum velocity of the target is also constrained to $\bar{v}_T \leq 2\,\text{m/s}$ to keep the $A - T$ speed ratio $\gamma_{AT} < 1$. Initial conditions of the agents are given in Table 2.1, Fig. 2.17, 2.20, 2.24, 2.27, 2.30, and 2.33.

2.5.2 Constant speed target

Given the initial engagement geometry, we can say whether the target will be captured or not using the analysis given in Sec. 2.4.2. We validate this claim through simulation for the initial configuration given in Fig. 2.17. The initial positions of the attacker and the defender are selected as $A(35, 0)$ and $D(-35, 0)$.

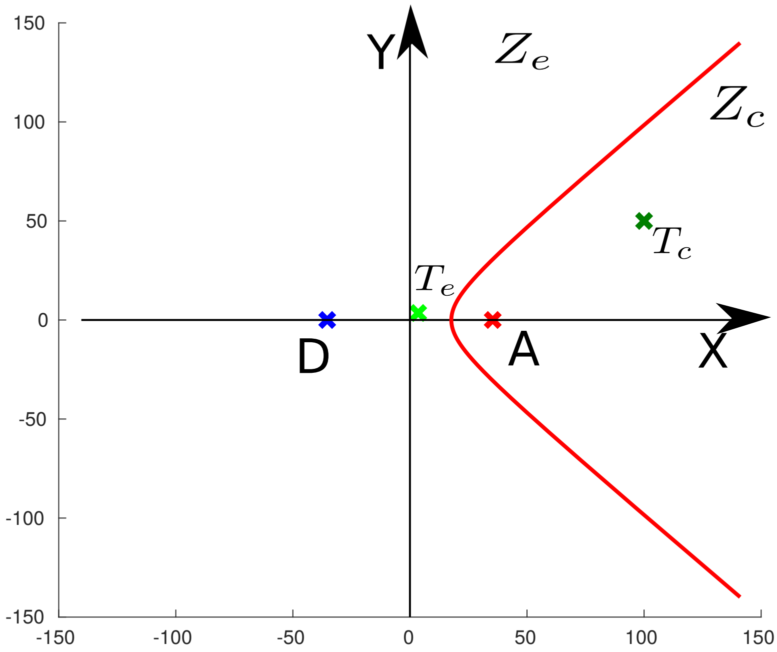
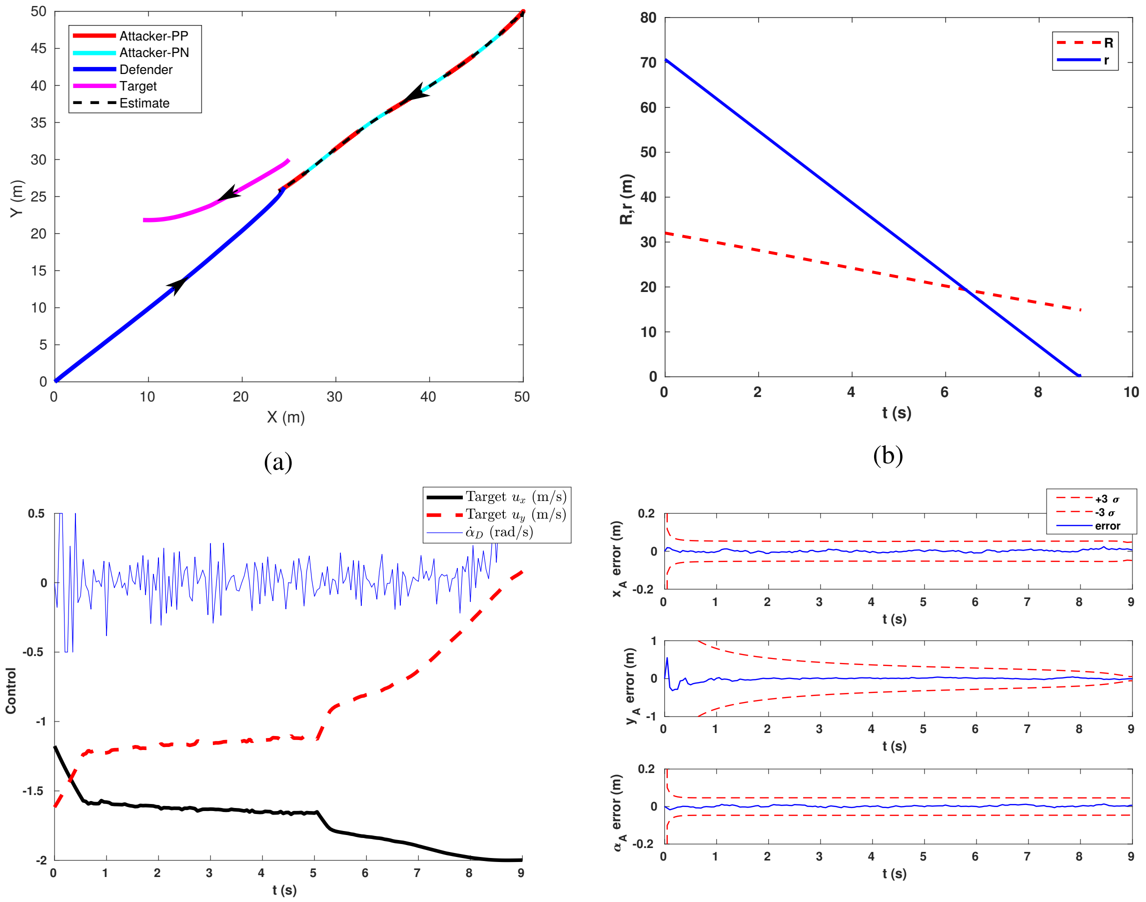


Figure 2.17: Initial agent configurations for the constant speed target case.

The $A - D$ speed ratio, $\gamma_{AD} = 1$, and the $A - T$ speed ratio, $\gamma_{AT} = 0.5$.

2.5.2.1 Target escape case

The initial position of the target is selected inside the escape zone Z_e and is represented by $T_e(3.5, 3.5)$ in Fig. 2.17. The initial conditions for the simulation are given in Table 2.1 escape column of the constant speed case. The agent trajectories are shown in Fig. 2.18a, and the evolution of the distances (R and r) between the agents are shown in Fig. 2.18b. Since the attacker guidance law is changing over time, we represent this change by using different color sequences in the plot, PP in red and PN in cyan. Even with the attacker guidance law switching from PP and PN, the defender was able to intercept the attacker and mitigate the target capture. The optimal control inputs determined by the NMPC are given in Fig. 2.18c. It can be seen that the bounds on the angular velocity of the defender were strictly followed. Fig. 2.18d shows the errors in the estimates of the attacker position and heading angle. The errors are low and stay within the 3σ bounds calculated using the corresponding covariances from the matrix P, which implies that the estimator performance is satisfactory.



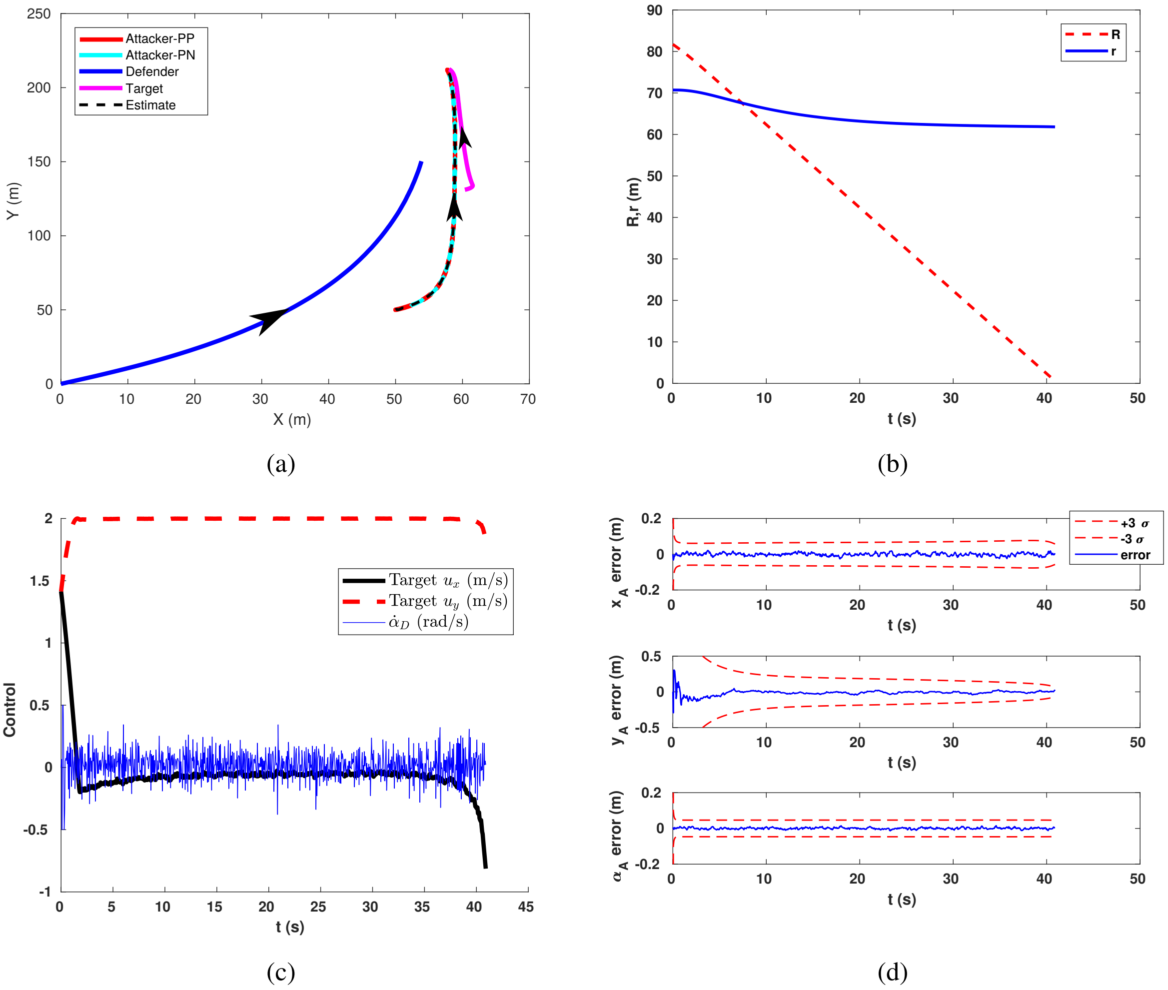
(a)

(b)

(c)

(d)

Figure 2.18: Target escape scenario for the constant speed target. (a) Trajectories of the agents. (b) Distance between the agents. (c) Optimal control inputs determined by the NMPC. (d) Error in the attacker state estimates.



(a) (b)

(c) (d)

Figure 2.19: Target capture scenario for the constant speed target. (a) Trajectories of the agents. (b) Distance between the agents. (c) Optimal control inputs determined by the NMPC. (d) Error in the attacker state estimates.

2.5.2.2 Target capture case

The initial position of the target is selected inside Z_c and is represented by $T_c(100, 50)$ in Fig. 2.17. The simulation parameters are shown in Table 2.1 capture column of the constant speed target. The agent trajectories and the evolution of the distances for this scenario are shown in Fig. 2.19a and Fig. 2.19b. The attacker captured the target before the defender could intercept the attacker. It can be seen from Fig. 2.19b that the distance between the attacker and the defender remained almost constant since they have equal velocities and are on a tail-chase engagement. The optimal control inputs and the error in attacker states are given in Fig. 2.19c and Fig. 2.19d, respectively. The target velocity stays constant at the specified value of 2 m/s throughout the engagement. The estimator performance was satisfactory since the errors never crossed the upper and lower covariance bounds.

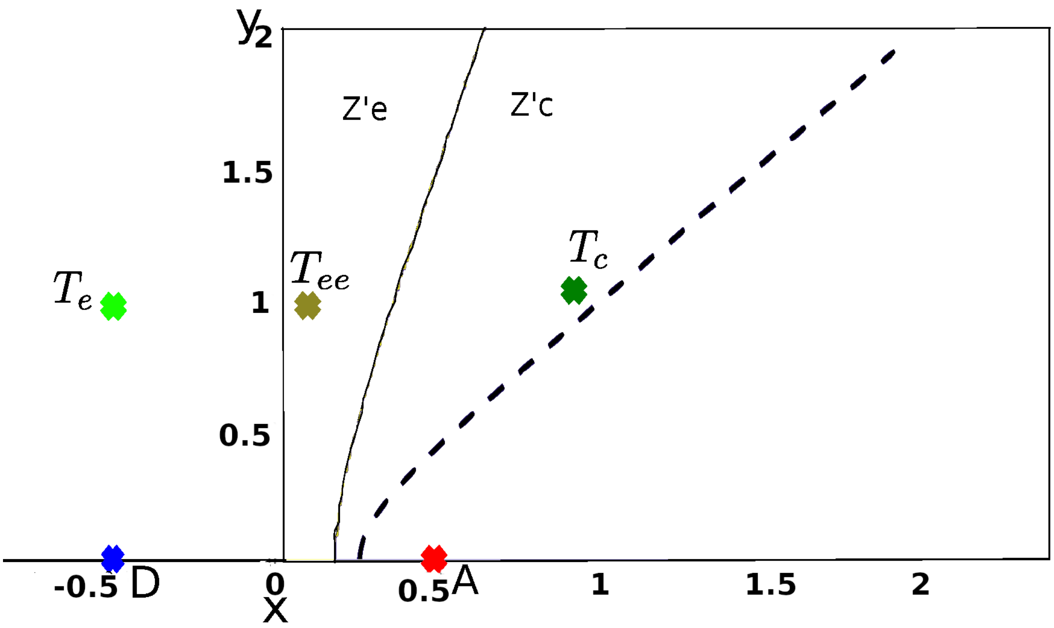


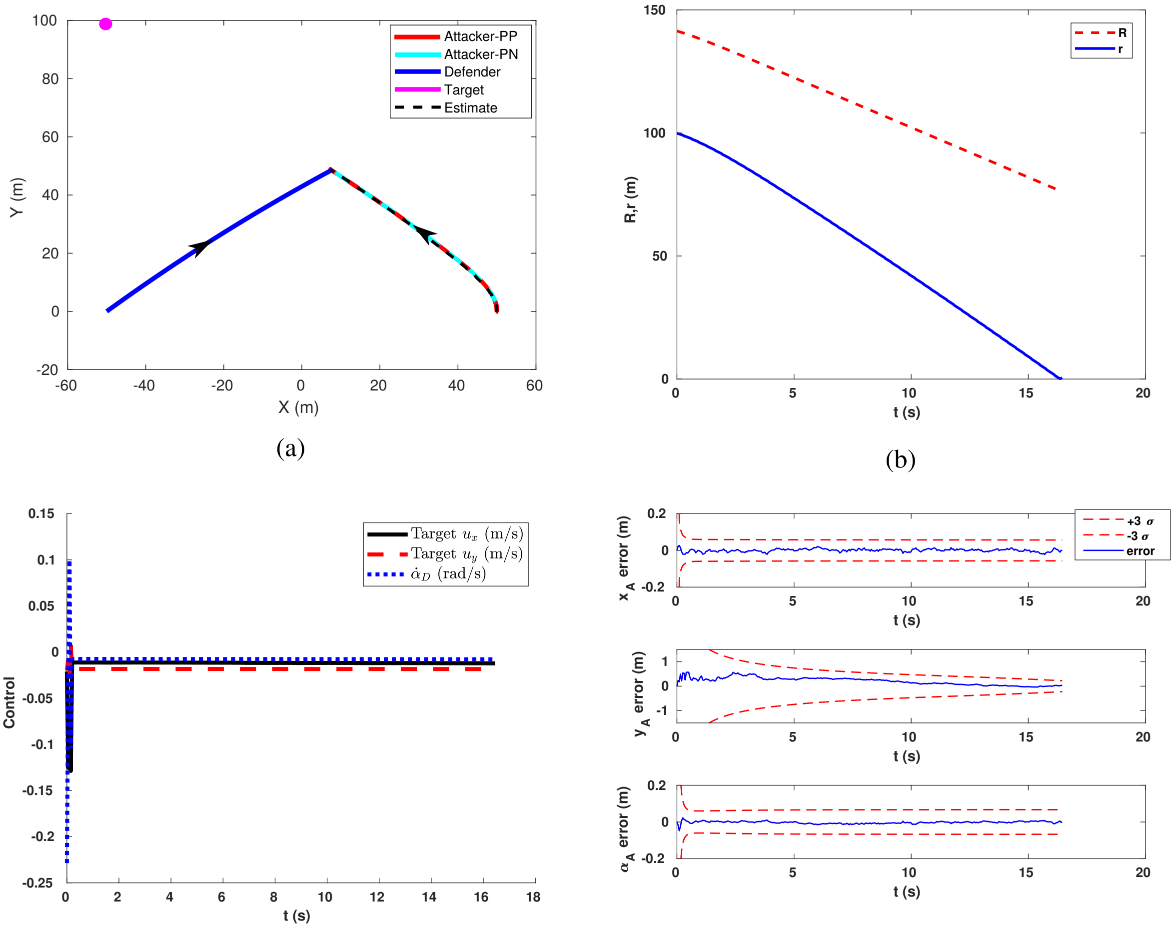
Figure 2.20: Initial agent configurations for the variable velocity target case. The dashed curve represents the constant speed case, the solid curve the variable velocity case, and the $y-$axis represents the boundary for the stationary target.

2.5.3 Variable target velocity case

The analysis given for the variable velocity target in Sec. 2.4.4 is validated using the initial configurations given in Fig. 2.20. The initial positions of the attacker and the defender are selected as $A(0.5, 0)$ and $D(-0.5, 0)$. The $A - D$ speed ratio, $\gamma_{AD} = 1$, and the $A - T$ speed ratio, $\gamma_{AT} = 0.5$. The safe distance parameter e is selected as half of the initial distance between the target and the attacker.

2.5.3.1 Target escape case where safe distance is not violated

The target escape scenario for the variable velocity target case, where the safe distance parameter e is not violated, is considered. The initial position of the target is selected inside the escape zone, Z_e', and is represented by $T_e(-0.5, 1)$ in Fig. 2.20. The agent trajectories and the evolution of the distances for this scenario are shown in Fig. 2.21a and Fig. 2.21b. It can be seen from the figures that the target is stationary since the attacker was intercepted by the defender before e is violated. It is evident that the target's control inputs would be zero for this case, which is verified in Fig. 2.21c. The defender's angular velocity is also almost zero due to the nearly straight line path taken by the agent. Fig. 2.21d shows the errors in the estimates of the attacker position and heading angle. The errors are low and stay within the 3σ bounds.



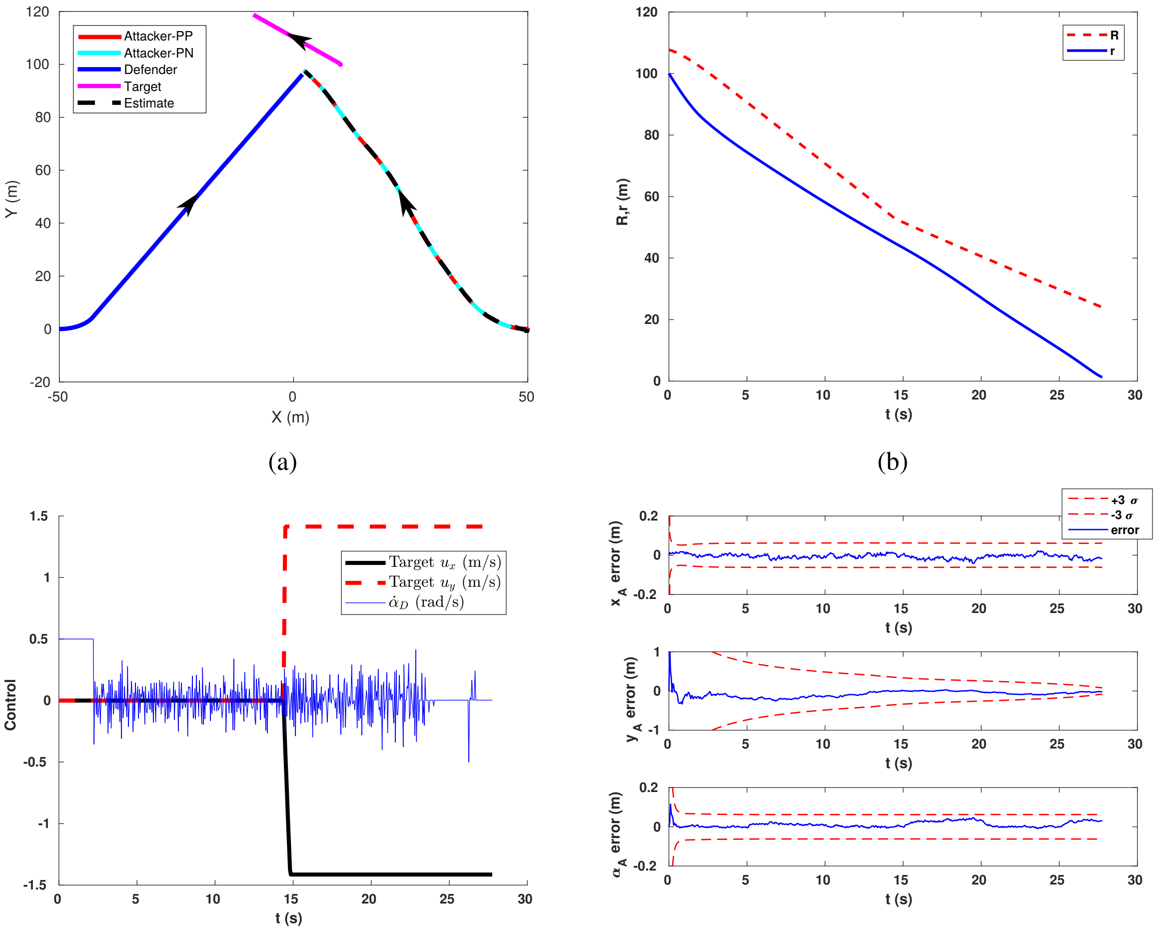
(a)

(b)

(c)

(d)

Figure 2.21: Target escape scenario for the variable velocity target where e is not violated. (a) Trajectories of the agents. (b) Distance between the agents. (c) Optimal control inputs determined by the NMPC. (d) Error in the attacker state estimates.



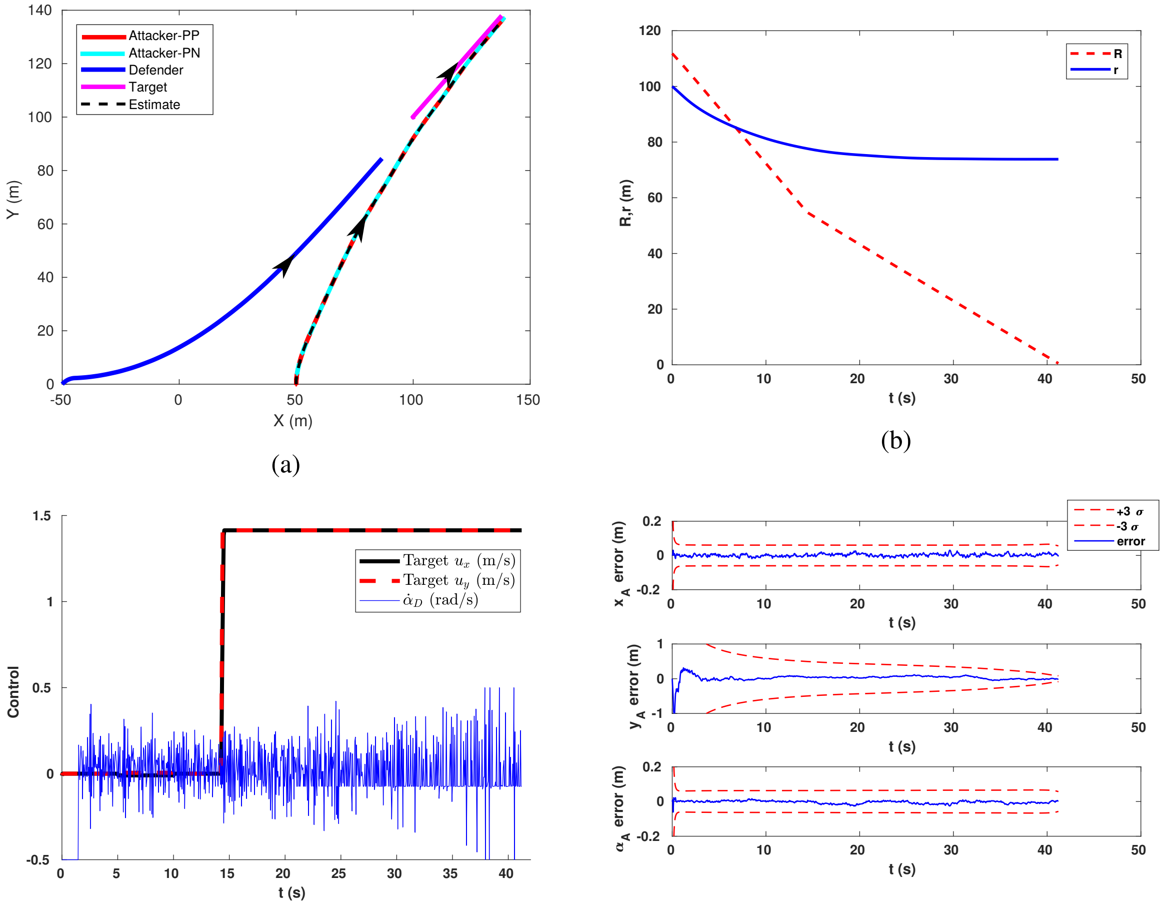
(a) (b)

(c) (d)

Figure 2.22: Target escape scenario for the variable velocity target where e is violated. (a) Trajectories of the agents. (b) Distance between the agents. (c) Optimal control inputs determined by the NMPC. (d) Error in the attacker state estimates.

2.5.3.2 Target escape case where safe distance is violated

The target escape scenario for the variable velocity target case, where the safe distance parameter e is violated, is considered here. The initial position of the target is selected inside the escape zone, Z'_e, and is represented by $T_{ee}(0.1, 1)$ in Fig. 2.20. The agent trajectories and the evolution of the distances for this scenario are shown in Fig. 2.22a and Fig. 2.22b. It can be seen from the figures that the target started moving when e was about to be violated, and the target's control inputs became nonzero, as shown in Fig. 2.22c. It can be seen from Fig. 2.22b that the slope of R was decreasing at a constant rate, and the rate decreased when the target started moving. Fig. 2.22d shows the errors in the estimates of the attacker position and heading angle, which is very low and stays within the 3σ bounds.



(a)

(b)

(c)

(d)

Figure 2.23: Target capture scenario for the variable velocity target. (a) Trajectories of the agents. (b) Distance between the agents. (c) Optimal control inputs determined by the NMPC. (d) Error in the attacker state estimates.

2.5.3.3 Target capture case

The initial position of the target is selected inside the capture zone, Z_c', and is represented by $T_c(1,1)$ in Fig. 2.20. The agent trajectories and the evolution of distances between the agents are shown in Fig. 2.23a and Fig. 2.23b, respectively. The optimal control inputs determined by the NMPC for the target and the defender are shown in Fig. 2.23c. It can be seen that the target's control inputs become nonzero only after the safe distance e was about to be violated. The target then moves with the maximum allowed velocity and tries to maintain $R > e$. The target was captured as the defender failed to intercept the attacker. Like the other simulations, the estimated errors were below the 3σ bounds. Fig. 2.23d shows the errors in estimated states, which is well below the 3σ bounds.

2.5.4 Unequal attacker–defender speed ratio case

Simulations were carried out with different configurations (constant speed and variable velocity target, $A - D$ speed ratio $\gamma_{AD} < 1$ and $\gamma_{AD} > 1$). The $A - T$ speed ratio is selected as $\gamma_{AT} = 0.5$. The safe distance parameter e is selected as half of the initial distance between the target and the attacker.

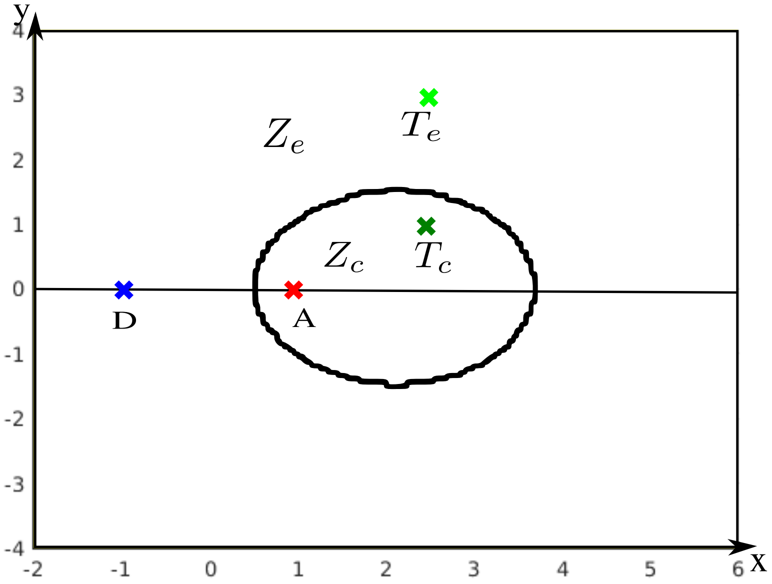


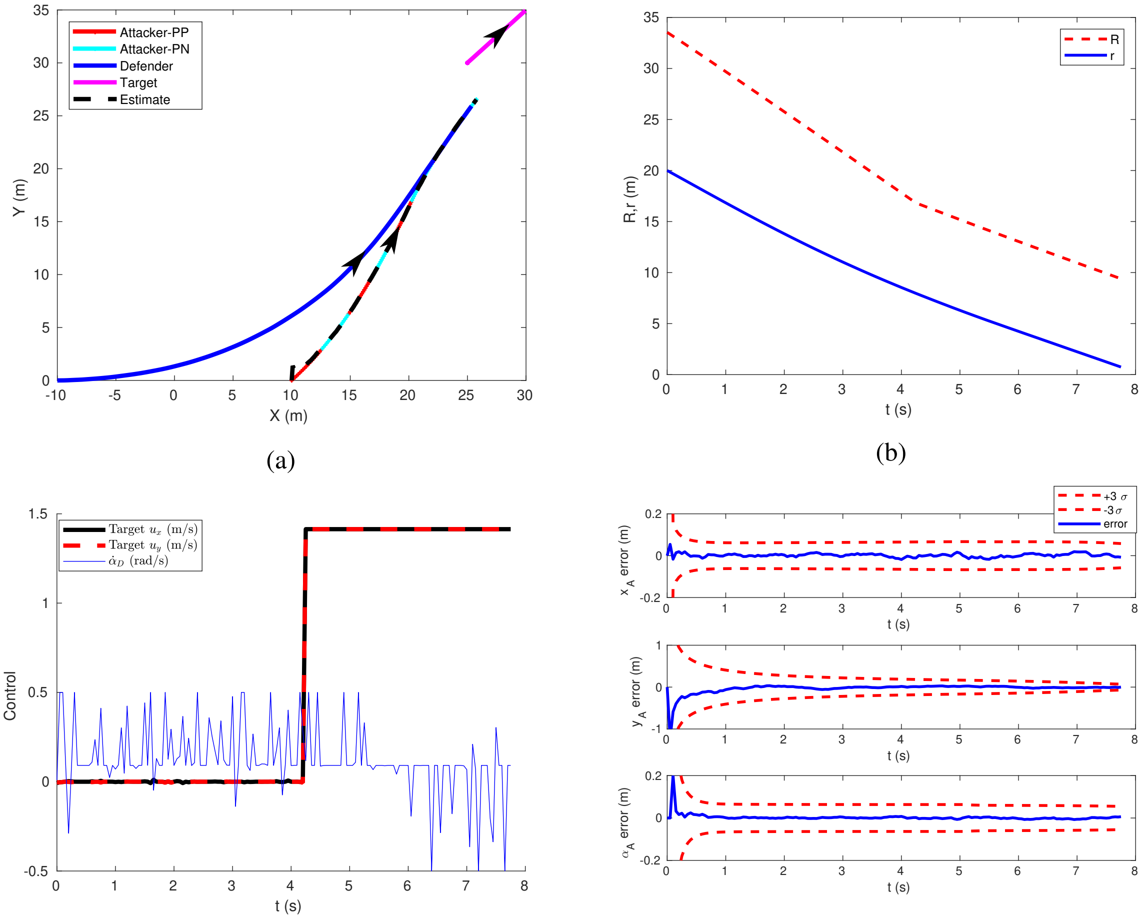
Figure 2.24: Initial agent configurations for the variable velocity target case, $\gamma_{AD} = 1.5$.

2.5.4.1 Variable velocity target, Target escape case, $\gamma_{AD} = 1.5$

The initial position of the target is selected inside the escape zone, Z_e, and is represented by $T_e(2.5, 3)$ in Fig. 2.24. The agent trajectories and the evolution of the distances for this scenario are shown in Fig. 2.25a and Fig. 2.25b. The safe distance e was violated, and hence the target's control inputs became nonzero, as shown in Fig. 2.25c. The distance between the attacker and the defender goes to zero in Fig. 2.25b, confirming the $A - D$ interception. Fig. 2.25d shows the errors in the estimates of the attacker states, which is within the acceptable tolerance range.

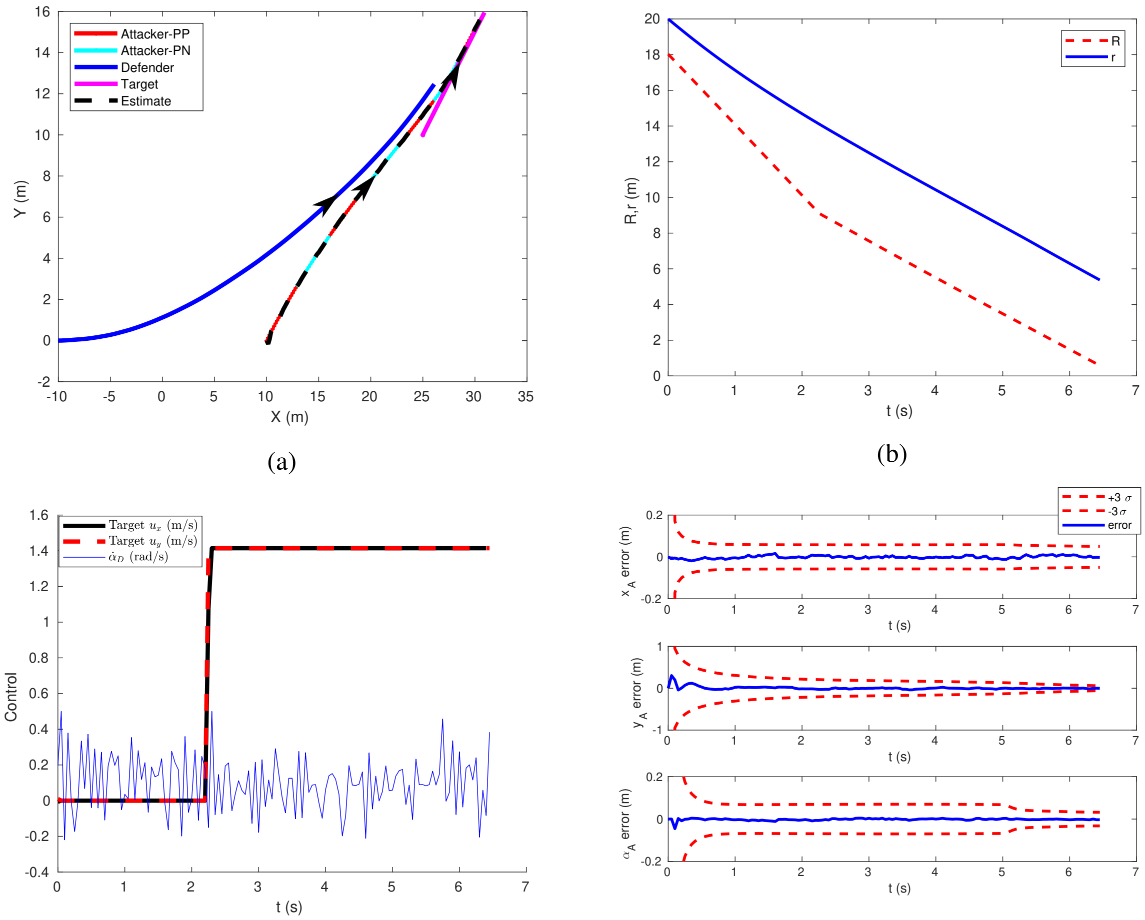
2.5.4.2 Variable velocity target, Target capture case, $\gamma_{AD} = 1.5$

The initial position of the target is selected inside the capture zone, Z_c, and is represented by $T_c(2.5, 1)$ in Fig. 2.24. The agent trajectories and the evolution of distances between the agents are shown in Fig. 2.26a and Fig. 2.26b respectively. The optimal control inputs determined by the NMPC for the target and the



(a) (b) (c) (d)

Figure 2.25: Target escape scenario for the variable velocity target, $\gamma_{AD} = 1.5$ (a) Trajectories of the agents. (b) Distance between the agents. (c) Optimal control inputs determined by the NMPC. (d) Error in the attacker state estimates.



(a) (b)

(c) (d)

Figure 2.26: Target capture scenario for the variable velocity target, $\gamma_{AD} = 1.5$. (a) Trajectories of the agents. (b) Distance between the agents. (c) Optimal control inputs determined by the NMPC. (d) Error in the attacker state estimates.

defender are shown in Fig. 2.26c. Even though the control inputs became nonzero after the safe distance e was violated, the target failed to escape since the defender was unable to intercept the attacker. The estimator performance was satisfactory, as seen in Fig. 2.26d.

2.5.4.3 Constant speed target, Target escape case, $\gamma_{AD} = 0.8$

The initial position of the target is selected inside the escape zone, Z_e, and is represented by T_e in Fig. 2.27. The agent trajectories and the evolution of the distances for this scenario are shown in Fig. 2.28a and Fig. 2.28b. The target moves at a constant speed away from the attacker. Hence, control commands are nearly constant, as shown in Fig. 2.28c. At the initial stages, some oscillations can be seen in the defender's input as it tries to align itself with the attacker's course. The distance between

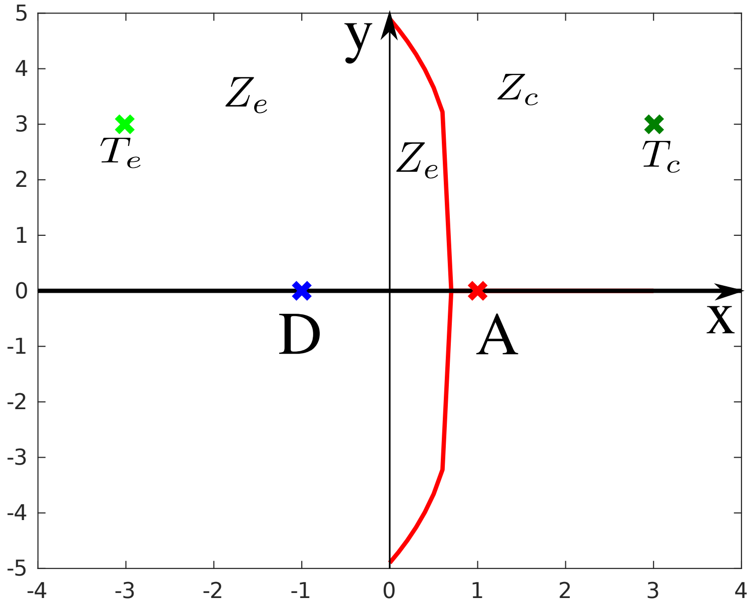
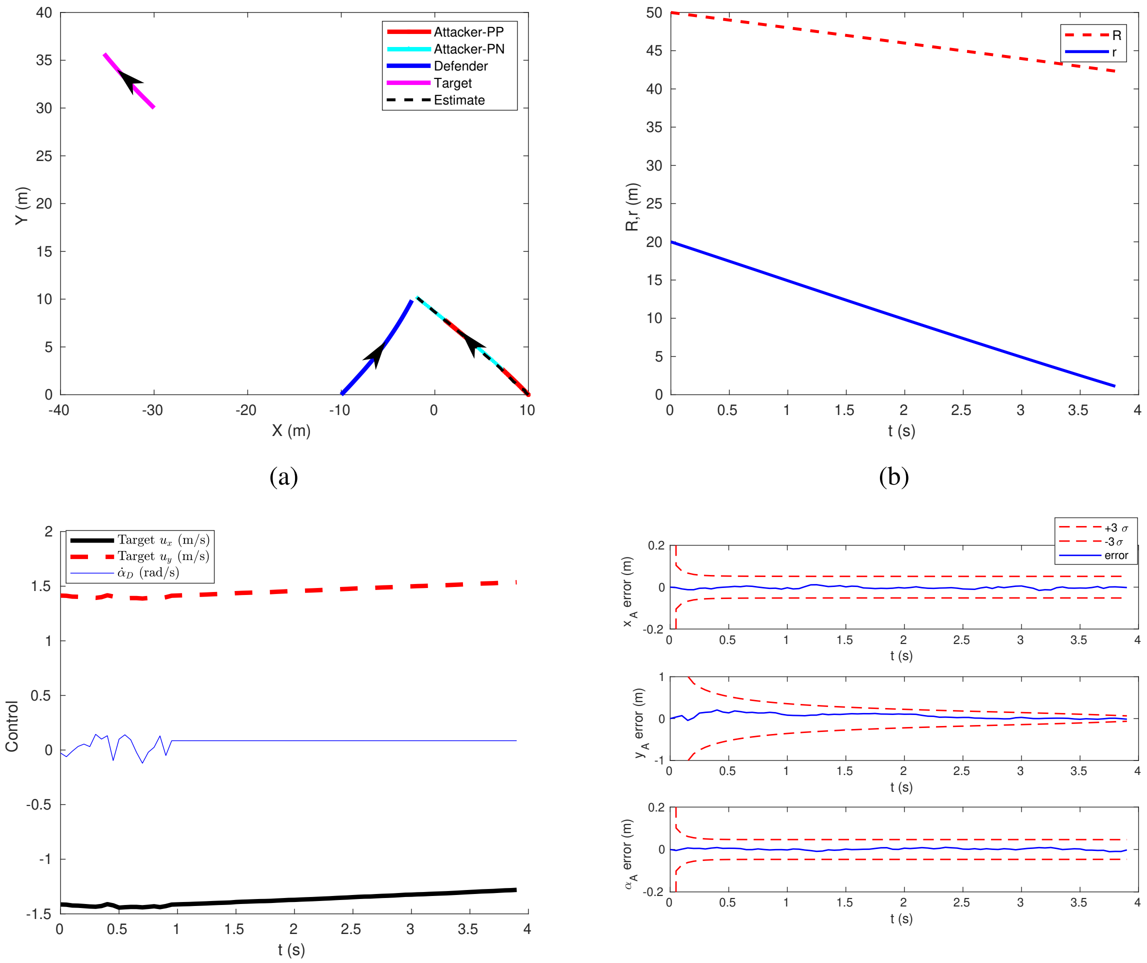


Figure 2.27: Initial agent configurations for the constant speed target case, $\gamma_{AD} = 0.8$

the attacker and the defender goes to zero, as shown in Fig. 2.28b since the attacker was intercepted. Fig. 2.28d shows the errors in the estimates of the attacker states, which is well within the acceptable tolerance range.

2.5.4.4 Constant speed target, Target capture case, $\gamma_{AD} = 0.8$

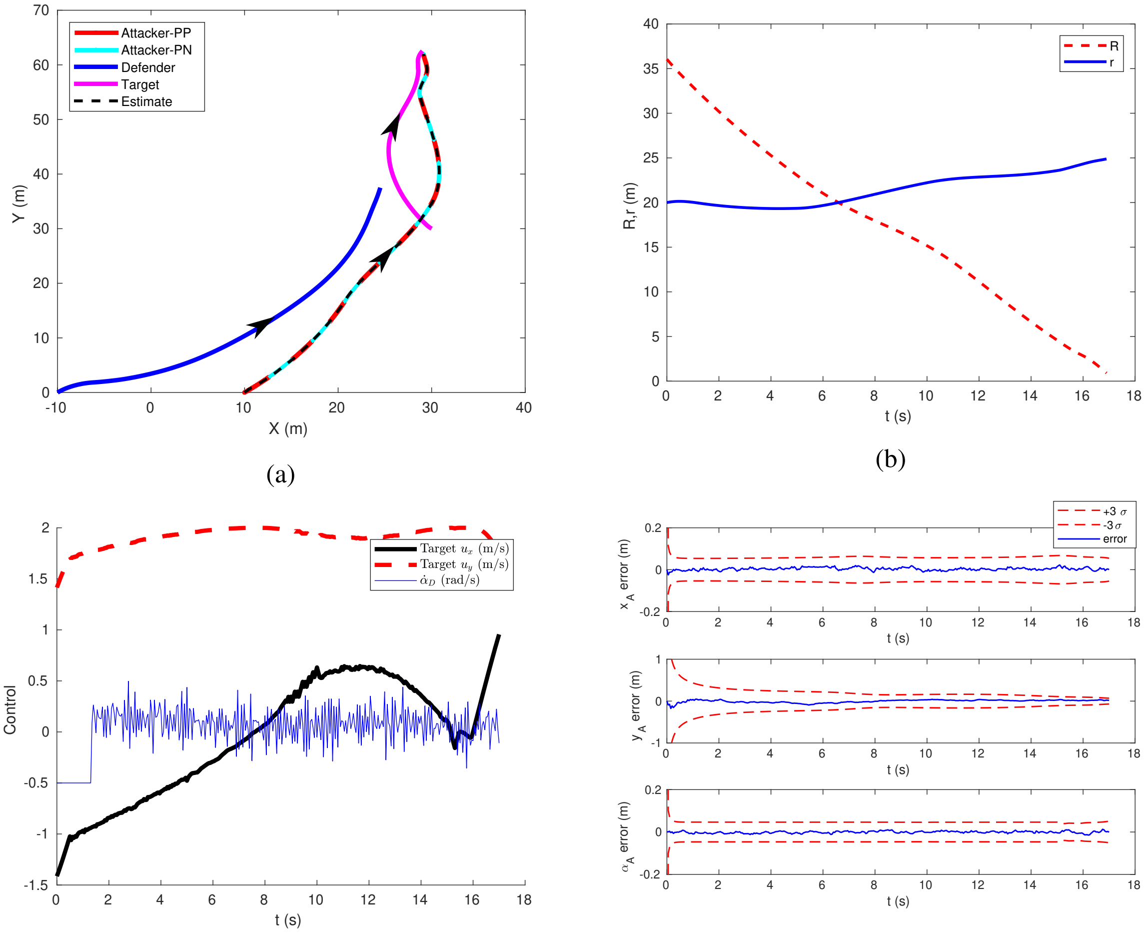
The initial position of the target is selected inside the capture zone, Z_c, and is represented by T_c in Fig. 2.27. The agent trajectories and the evolution of distances between the agents are shown in Fig. 2.29a and Fig. 2.29b respectively. The target was captured even though it tried to escape by taking a zig-zag trajectory. Since the defender was in a tail chase scenario with a higher-speed attacker, it was not able to intercept the attacker. This can be seen in Fig. 2.29b that the distance r does not decrease and is nearly constant. The optimal control inputs determined by the NMPC for the target and the defender are shown in Fig. 2.29c. Identifiable changes can be seen in the target controls due to its zig-zag trajectory. The estimator performance was satisfactory, as seen in Fig. 2.29d, and the failure of the defender was not due to estimation errors.



(a) (b)

(c) (d)

Figure 2.28: Target escape scenario for the constant speed target, $\gamma_{AD} = 0.8$. (a) Trajectories of the agents. (b) Distance between the agents. (c) Optimal control inputs determined by the NMPC. (d) Error in the attacker state estimates.



(a)

(b)

(c)

(d)

Figure 2.29: Target capture scenario for the constant speed target, $\gamma_{AD} = 0.8$. (a) Trajectories of the agents. (b) Distance between the agents. (c) Optimal control inputs determined by the NMPC. (d) Error in the attacker state estimates.

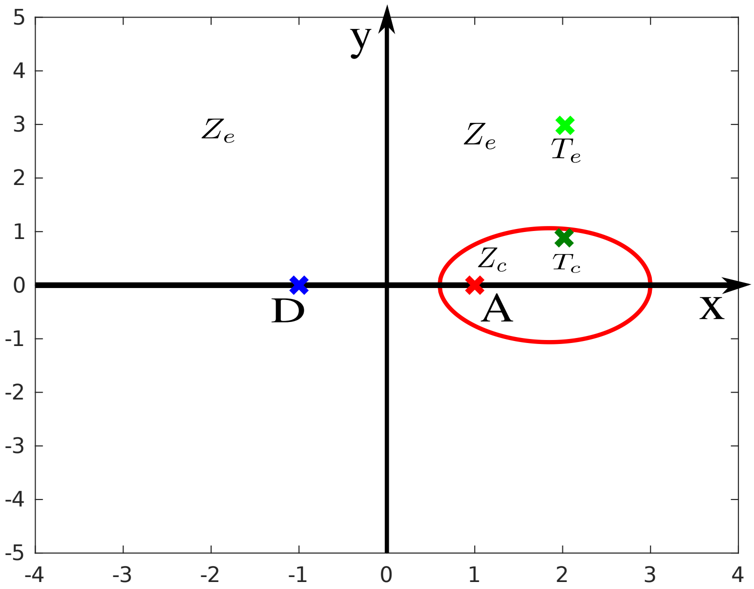


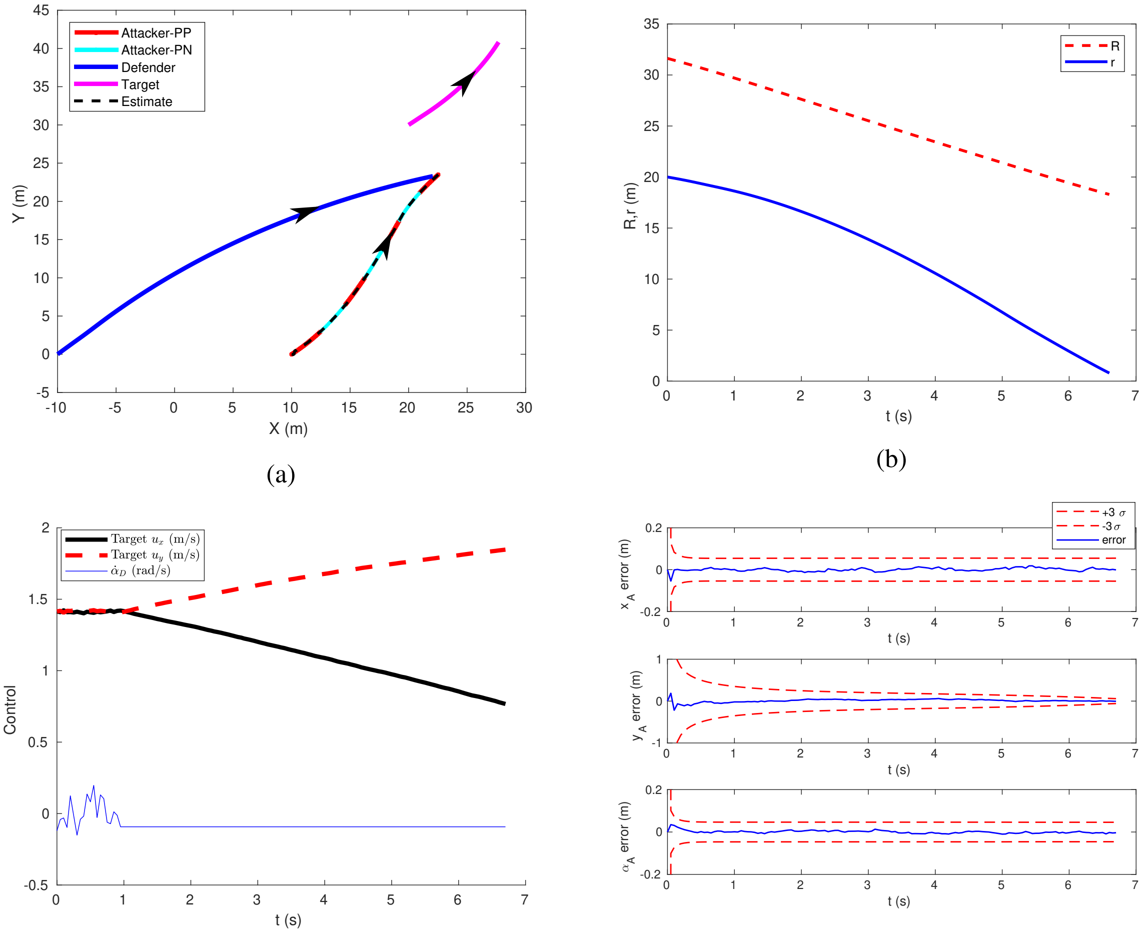
Figure 2.30: Initial agent configurations for the constant speed target case, $\gamma_{AD} = 1.5$

2.5.4.5 Constant speed target, Target escape case, $\gamma_{AD} = 1.5$

The initial position of the target is selected inside the escape zone, Z_e and is represented by T_e in Fig. 2.30. The agent trajectories and the evolution of the distances for this scenario are shown in Fig. 2.31a and Fig. 2.31b. The target initially moves away from the attacker and then takes a slight left turn to make the interception of the attacker easy for the defender. Hence, we can see a change in control commands after the initial stage, as shown in Fig. 2.31c. The distance between the attacker and the defender goes to zero, as shown in Fig. 2.31b confirming the interception of the attacker. Fig. 2.31d shows the errors in the estimates of the attacker states, which is bounded by σ.

2.5.4.6 Constant speed target, Target capture case, $\gamma_{AD} = 1.5$

The initial position of the target is selected inside the capture zone, Z_c, and is represented by T_c in Fig. 2.30. The agent trajectories and the evolution of distances between the agents are shown in Fig. 2.32a and Fig. 2.32b respectively. The target was captured even though it tried to escape by taking an ‘S’-shaped trajectory. Even though the defender is faster than the attacker, it was not able to intercept the attacker due to the high initial separation between them. This can be seen in Fig. 2.32b that the distance r is decreasing to zero, and the defender was close to intercepting the attacker. The optimal control inputs determined by the NMPC for the target and the defender are shown in Fig. 2.32c. Intermittent changes



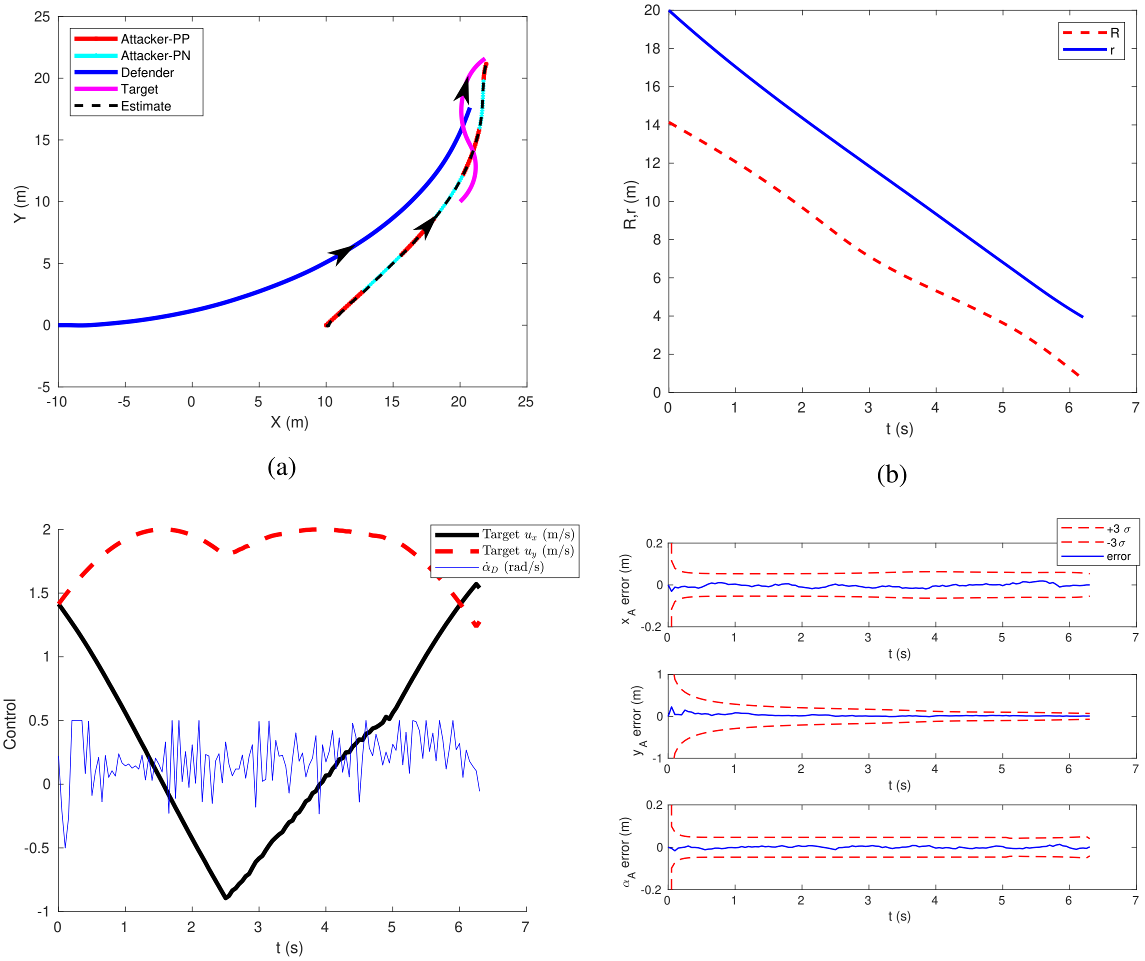
(a)

(b)

(c)

(d)

Figure 2.31: Target escape scenario for the constant speed target, $\gamma_{AD} = 1.5$. (a) Trajectories of the agents. (b) Distance between the agents. (c) Optimal control inputs determined by the NMPC. (d) Error in the attacker state estimates.



(a) (b) (c) (d)

Figure 2.32: Target capture scenario for the constant speed target, $\gamma_{AD} = 1.5$. (a) Trajectories of the agents. (b) Distance between the agents. (c) Optimal control inputs determined by the NMPC. (d) Error in the attacker state estimates.

in the target controls are due to the 'S'-shaped trajectory taken by the target. The estimator performance was satisfactory in this case also, and the errors are shown in Fig. 2.32d.

2.5.4.7 Variable velocity target, Target escape case, $\gamma_{AD} = 0.8$

The initial position of the target is selected inside the escape zone, Z_e, and is represented by T_e in Fig. 2.33. The agent trajectories and the evolution of the distances for this scenario are shown in Fig. 2.34a and Fig. 2.34b. The attacker was intercepted by the defender before it could breach the safe radius. Hence, the target stayed in its position to perform its mission. Since the safe distance e was not violated, the target's control inputs stayed near zero, as shown in Fig. 2.34c. The distance between the

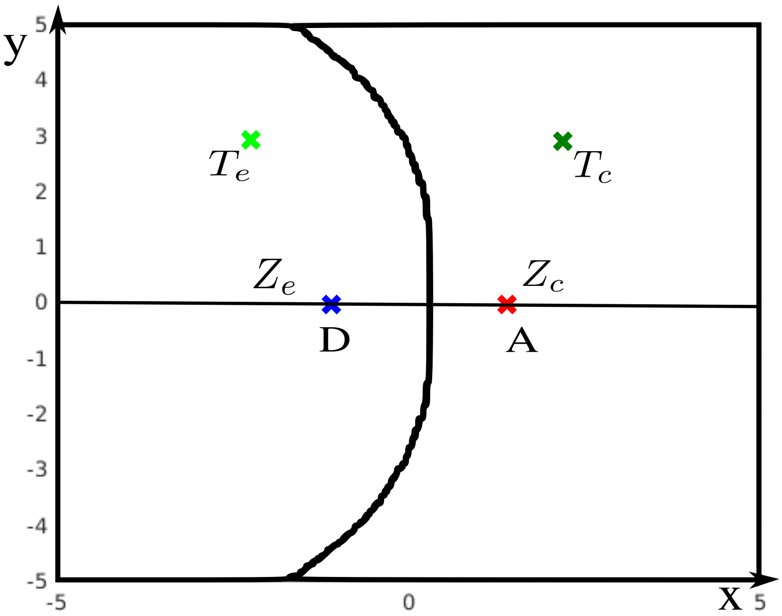
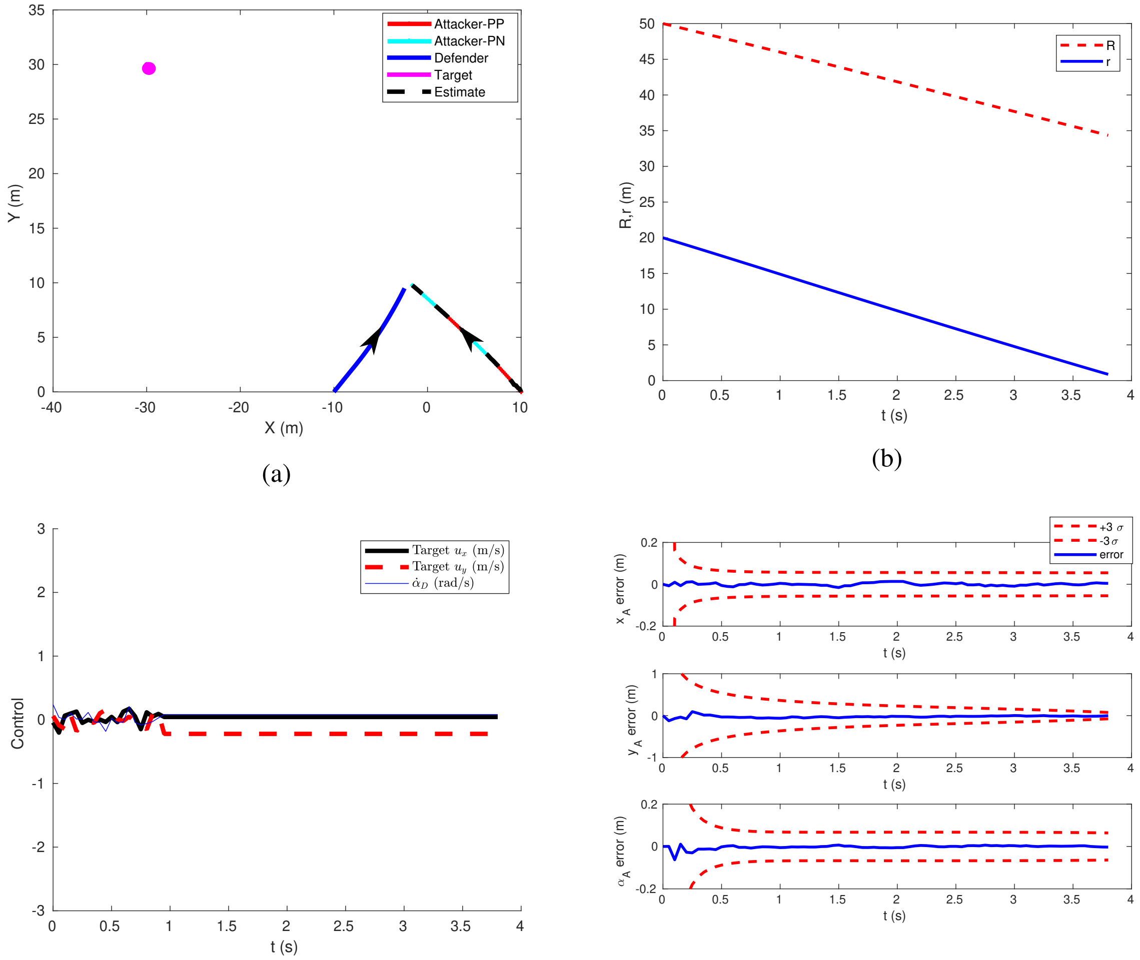


Figure 2.33: Initial agent configurations for the variable velocity target case, $\gamma_{AD} = 0.8$.

attacker and the defender goes to zero in Fig. 2.34b, confirming the $A - D$ interception. The distance R was reduced at a constant rate since the target was stationary. Fig. 2.34d shows the errors in the estimates of the attacker states, which is within the $\pm\sigma$ bounds.

2.5.4.8 Variable velocity target, Target capture case, $\gamma_{AD} = 0.8$

The initial position of the target is selected inside the capture zone, Z_c, and is represented by T_c in Fig. 2.33. The agent trajectories and the evolution of distances between the agents are shown in Fig. 2.35a and Fig. 2.35b respectively. The target started moving when the safe distance was violated. It can be seen that there is a change in slope in the distance plot R due to the target motion. The defender could not intercept the attacker since it was chasing an agent with a higher velocity. This is evident from Fig. 2.35b since the value of r is almost constant. The optimal control inputs determined by the NMPC for the target and the defender are shown in Fig. 2.35c. The control values for the target rose to the maximum when the safe distance was violated. The estimator performance was satisfactory in this case also, as seen in Fig. 2.35d.



(a) (b) (c) (d)

Figure 2.34: Target escape scenario for the variable velocity target, $\gamma_{AD} = 0.8$ (a) Trajectories of the agents. (b) Distance between the agents. (c) Optimal control inputs determined by the NMPC. (d) Error in the attacker state estimates.

Comparison			
parameter	NMPC	A-CLOS	CLOS
(x_A, y_A) (m)	(0,5000)	(0,5000)	(0,5000)
(x_T, y_T) (m)	(0,0)	(0,0)	(0,0)
(x_D, y_D) (m)	(0,0)	(0,0)	(0,0)
α_A (rad)	-1.57	-1.57	-1.57
α_T (rad)	0	0	0
α_D (rad)	1.57	1.57	1.57
v_A (m/s)	600	600	600
v_T (m/s)	200	200	200
v_D (m/s)	600	600	600

Table 2.2: Initial parameters for the agents for the comparison with the A-CLOS and CLOS guidance laws.

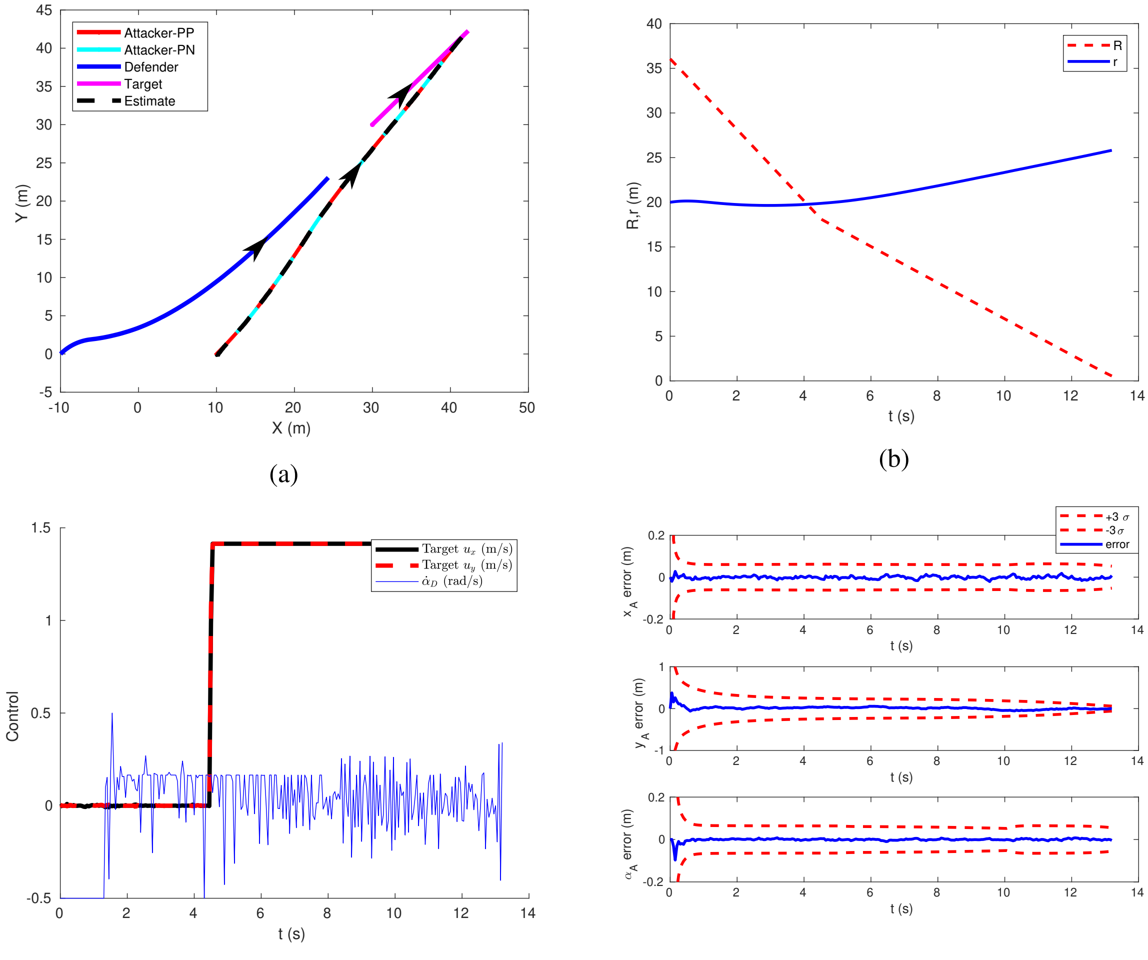


Figure 2.35: Target capture scenario for the variable velocity target, $\gamma_{AD} = 0.8$. (a) Trajectories of the agents. (b) Distance between the agents. (c) Optimal control inputs determined by the NMPC. (d) Error in the attacker state estimates.

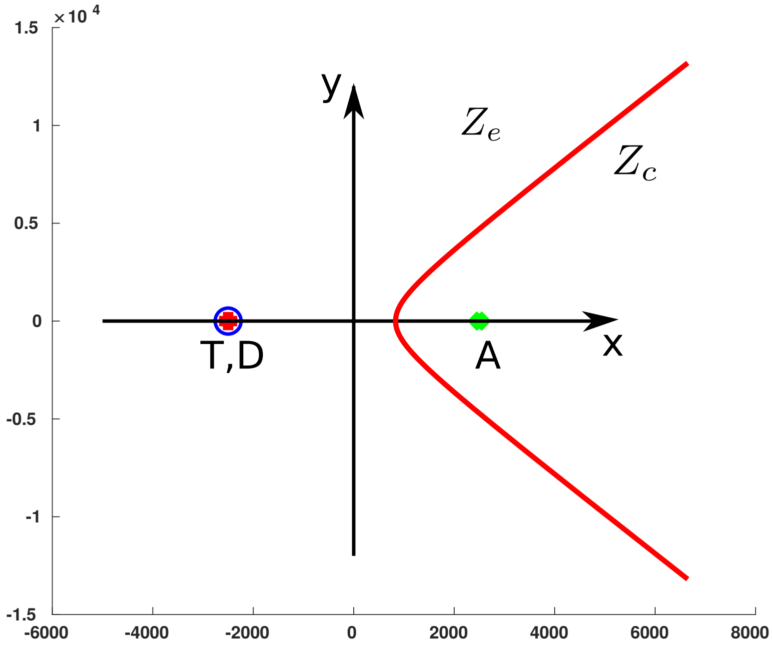


Figure 2.36: Initial agent configurations for the comparison of CLOS, A-CLOS, and NMPC.

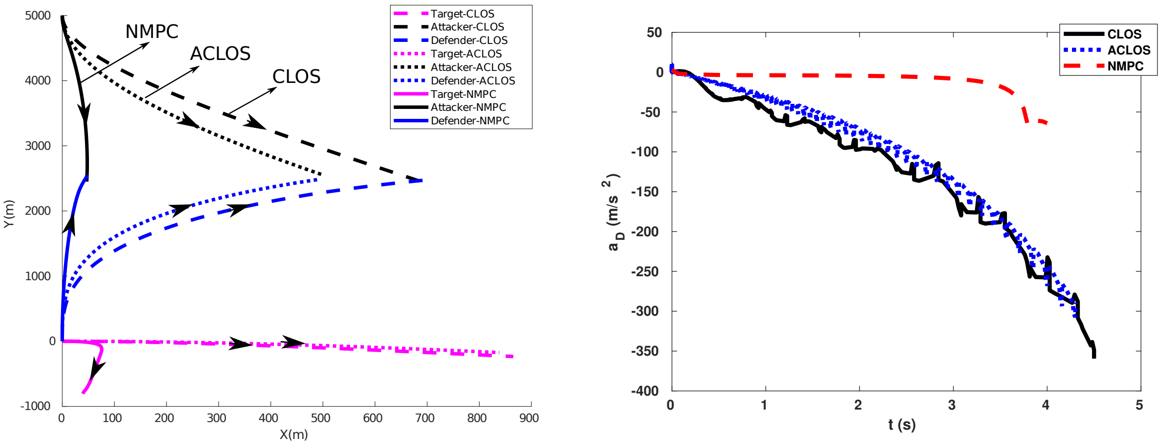
Parameter	NMPC	A-CLOS	CLOS
interception time (s)	4.0	4.3	4.5
avg. control effort (m/s^2)	350.97	3.38e+03	1.11e+04
avg. computation time / iteration	0.0374	0.0018	0.0038

Table 2.3: Performance comparison between NMPC, A-CLOS, and CLOS.

2.5.5 Comparison with CLOS and A-CLOS guidance laws

The proposed NMPC scheme for the three-agent pursuit-evasion problem was compared with existing solutions which use a command to line-of-sight (CLOS) guidance [37] and a modified CLOS guidance, called A-CLOS law [38], for the calculation of control input for the defender. The three-agent system was simulated with the same initial conditions as given in Table 2.2 and was compared with the solutions given by the CLOS and the A-CLOS guidance laws.

Velocities of the agents were selected as $v_T = 200\,\text{m/s}$, $v_A = 600\,\text{m/s}$ and $v_D = 600\,\text{m/s}$ respectively to match the parameters given in [37]. The initial positions of the agents are shown in Fig. 2.36. The target and the defender start from the same initial position, and the target moves with constant speed. The initial position of the target lies in the escape zone Z_e. The agent trajectories for the CLOS, A-CLOSG, and the NMPC for the selected scenario are given in Fig. 2.37a. The computed control input for the defender, which is the lateral acceleration, is given in Fig. 2.37b for all the approaches. The simulation



(a) (b)

Figure 2.37: Performance comparison of the NMPC, CLOS, and the A-CLOS. (a) Agent trajectories with CLOS, A-CLOS, and NMPC. (b) Control effort comparison for the CLOS, A-CLOS, and NMPC.

results show that the NMPC outperformed the CLOS and A-CLOS with lower interception time and average control effort, as seen from Table 2.3. Even though the NMPC has a higher computation time compared to the CLOS and A-CLOS, it should be considered as a trade-off between computation time and efficiency. Nevertheless, the proposed NMPC scheme is real-time implementable according to our experience [54].

2.5.6 Experimental results

The efficacy of the proposed NMPC scheme was validated through implementation on ground rovers. Three ground rovers were used as the target, the attacker, and the defender. Initially, Software-In-The-Loop (SITL) simulations were performed, which mimic real-world conditions. Each rover has a Pixhawk autopilot, a GPS receiver, and a Raspberry Pi onboard computer. GPS was used for localization, and the onboard computer was used for connecting the rover to the local network. The experimental setup uses Robotic Operating System (ROS) in a single-master-multiple-slave configuration, where a Ground Control Station (GCS) is the ROS master, and each rover is configured as a ROS slave. MAVROS was used as the inter-vehicle communication framework. The experimental setup is shown in Fig. 2.38.

The velocities of the agents were selected as $v_A = 0.4$ m/s, $\bar{v}_T = 0.3$ m/s, and $v_D = 0.5$ m/s. The control inputs were selected as linear and angular velocity commands. ROS-MATLAB toolkit was used for interfacing ROS and MATLAB. The ROS node, written in MATLAB, publishes command velocities and subscribes to feedback topics from all the agents. The command velocities are computed in real-

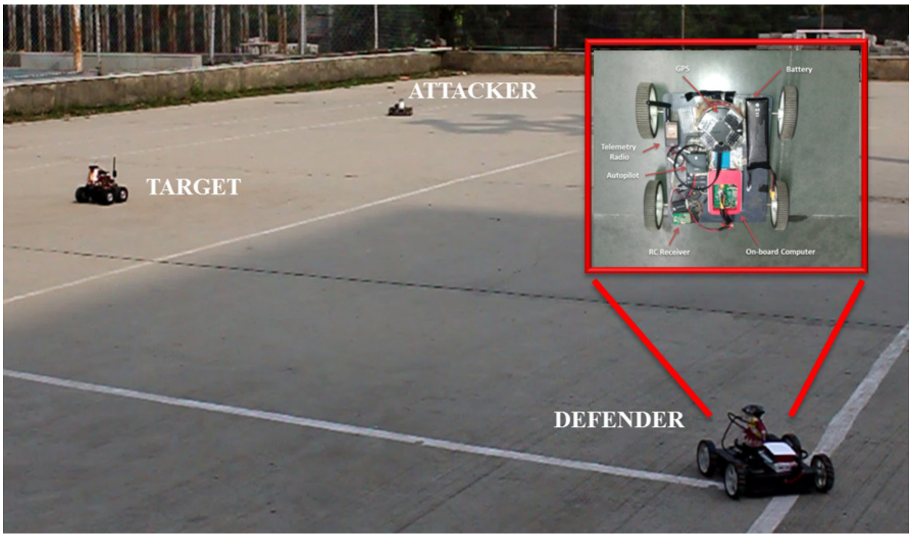
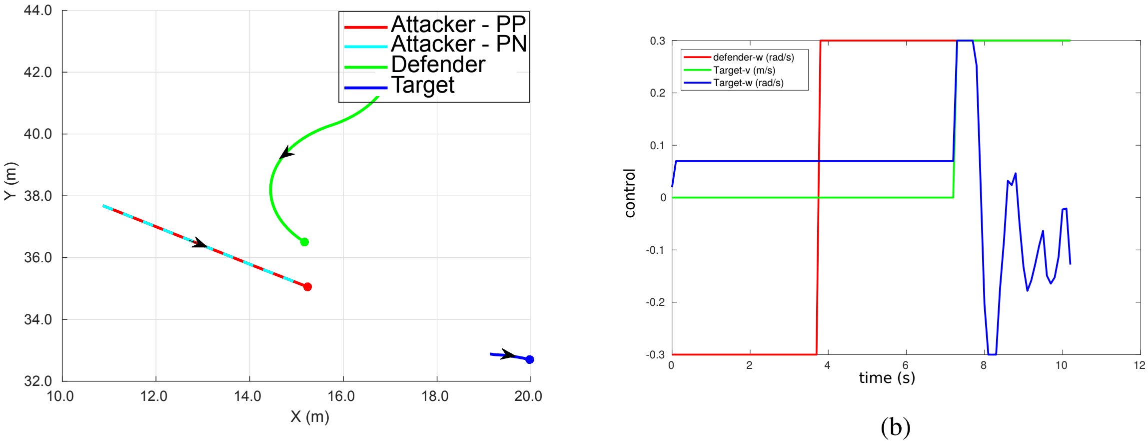


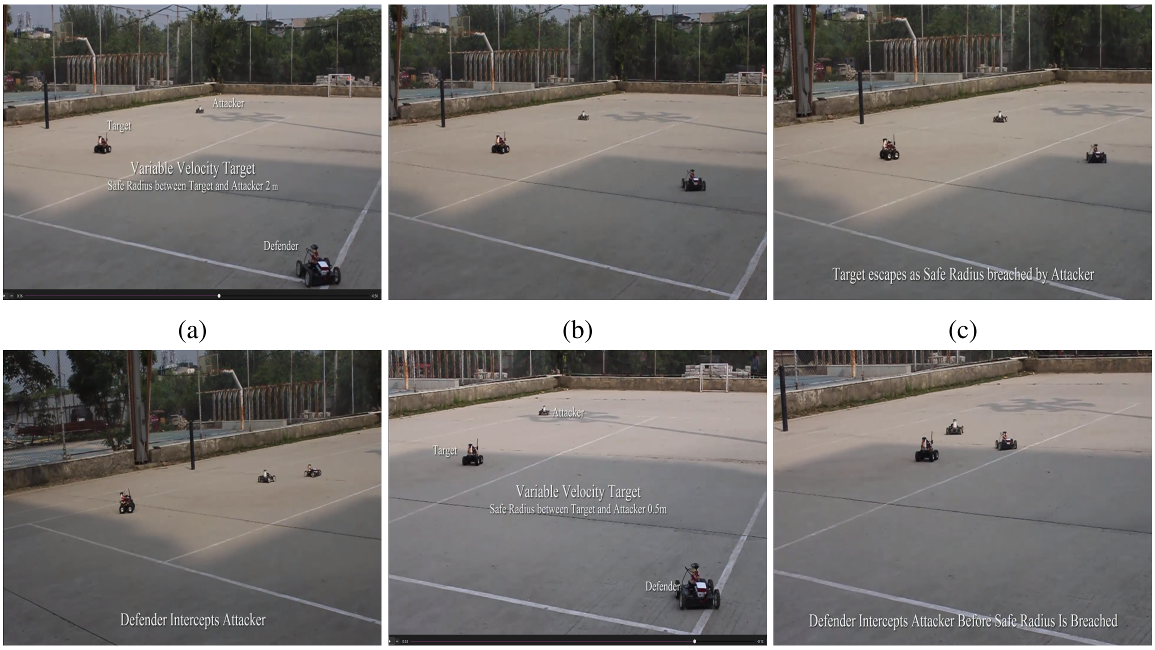
Figure 2.38: Experimental setup



(a)

(b)

Figure 2.39: Experimental results. (a) Rover trajectories (b) Rover control inputs.



(a) (b) (c)

(d) (e) (f)

Figure 2.40: Snapshots from the experiment. (a) Initial setup with a safe radius of 2 m (b) Rovers moving (c) Target started moving as the safe radius was breached (d) Defender intercepts the attacker (e) Initial setup with a safe radius of 0.5 m (f) Defender intercepts the attacker before the safe radius was violated.

time using NMPC. Multiple scenarios were tested to evaluate the performance of the proposed strategy, and one selected experimental result is given in Fig. 2.39. The rover trajectories shown in Fig. 2.39a represent the variable velocity target case. The target stays still initially and moves only when the safe distance is about to be breached. The control inputs for the agents are shown in Fig. 2.39b. Defender input rises after some time to change its course to align with the attacker. The target velocity became non-zero when the safe distance was violated, and the target started moving.

Some snapshots from the experiments are discussed next. Fig. 2.40a shows the initial setup for variable velocity target case with a safe radius of 2 m. Fig. 2.40b shows the rovers in motion. The target stays stationary. The attacker is moving toward the target, and the defender is moving toward the attacker. In Fig. 2.40c, we can see that the safe distance was violated, and the target started moving to escape from the attacker. Fig. 2.40d shows the final stage where the defender intercepted the attacker, and the target escaped. Next, the setup is changed with a smaller safe radius of 0.5 m, as shown in Fig. 2.40e. In this case, the defender intercepted the attacker before the safe distance was violated. Hence, the target stayed stationary throughout the experiment, as shown in Fig. 2.40f. A video of the experimental results can be found at `https://youtu.be/IuvKP1vifh0.`

2.6 Conclusions

A nonlinear model predictive control strategy was proposed for the active defense of the target in a TAD game. The formulation involves computing control commands for a cooperative target-defender pair against an individually acting attacker. The state information of the attacker was assumed to be unknown and was estimated using an EKF. An analysis using the Apollonius circles was conducted to determine the escape regions for the target, and the same was verified using simulations conducted for various scenarios. The nonlinear online feedback scheme designed using the NMPC was found to be effective in achieving the objectives while respecting the imposed constraints. The performance of the NMPC strategy was compared against CLOS and A-CLOS, and the results showed that the NMPC outperformed both the CLOS and A-CLOS based strategies. We have also experimentally demonstrated the efficacy and real-time implementability of the proposed approach using three rovers outdoors.

The main limitation of the proposed NMPC scheme is the increase in the computational burden for longer prediction horizons. Although reducing the horizon helps in real-time implementation, it makes the solution sub-optimal. Also, we have yet to define any clear rule for selecting the safe distance parameter, e. More analysis is required to find the best value for e under different scenarios. The current work has limitations due to assumption (1), which can be relaxed, and a more realistic analysis can be performed. Also, it is possible to broaden the framework to three dimensions taking the terrain map into account and including obstacles in urban environments. The major challenge in formulating differential games in 3D compared to the 2D case is the tractability of analytic solutions. The use of intersection spheres is required instead of Apollonius circles. Also, extra control inputs are required since each agent's azimuth and elevations need to be controlled. This makes the dynamics more nonlinear, computationally intensive to optimize, and hard to find the escape regions for the agents in the game. The recent works [135, 136] show promising results on how to tackle these issues.

Chapter 3

NMPC-Based Cooperative Strategy To Lure Two Attackers Into Collision By Two Targets

3.1 Introduction

Here we consider a pursuit-evasion problem involving a pair of attackers $(A1, A2)$ pursuing a pair of targets $(T1, T2)$ as shown in Fig. 3.1. We call this the two–targets two–attackers (2T2A) problem, where the objective of the target pair is to maneuver in such a way that the attackers collide with each other, ensuring the survival of the targets. The target pair is cooperative, whereas the attackers act individually. This type of scenario often occurs in the real world when heat-seeker missiles are fired to intercept the targets (an airplane or helicopter). The missiles simply follow the heat source (target) and generally do not cooperate with other missiles, whereas the targets can cooperate and devise a plan to escape.

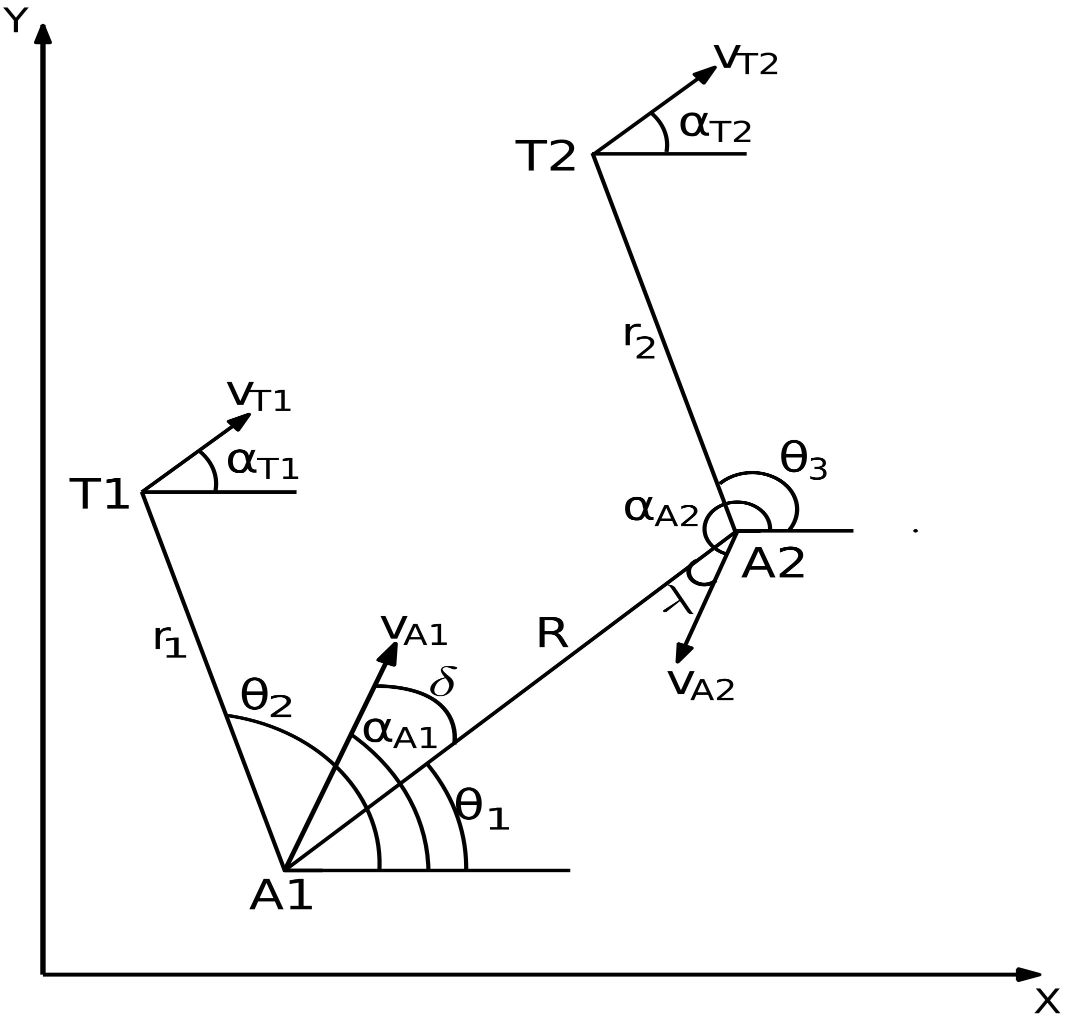


Figure 3.1: Four-agent engagement geometry. $A1$ and $A2$ are the attackers, $T1$ and $T2$ are the targets.

3.2 Problem formulation

The engagement geometry of the four agents is shown in Fig. 3.1. The equations of motion of the four agents can be written as [58, 126]

$$\dot{x}_i(t) = v_i \cos \alpha_i(t), \tag{3.1}$$

$$\dot{y}_i(t) = v_i \sin \alpha_i(t), \tag{3.2}$$

where $i = \{T1, T2, A1, A2\}$, $v_{T1},\ v_{T2}$ are the velocities of the targets, $v_{A1},\ v_{A2}$ are the velocities of the attackers, $\alpha_{T1}(t),\ \alpha_{T2}(t)$ are the heading angles of the respective targets, and $\alpha_{A1}(t),\ \alpha_{A2}(t)$ are the heading angles of the respective attackers. The velocities of the attackers and the targets are assumed to be constant. All the states and heading angles of the agents are changing with respect to time t, and the notation (t) is omitted in the rest of the document for readability. The equations governing the relative motion of the four agents can be written as [58, 126]

$$\dot{R} = v_{A2} \cos(\alpha_{A2} - \theta_1) - v_{A1} \cos(\alpha_{A1} - \theta_1), \tag{3.3}$$

$$\dot{r}_1 = v_{T1} \cos(\alpha_{T1} - \theta_2) - v_{A1} \cos(\alpha_{A1} - \theta_2), \tag{3.4}$$

$$\dot{r}_2 = v_{T2} \cos(\alpha_{T2} - \theta_3) - v_{A2} \cos(\alpha_{A2} - \theta_3), \tag{3.5}$$

$$\dot{\theta}_1 = \frac{1}{R}\left(v_{A2} \sin(\alpha_{A2} - \theta_1) - v_{A1} \sin(\alpha_{A1} - \theta_1)\right), \tag{3.6}$$

$$\dot{\theta}_2 = \frac{1}{r_1}\left(v_{T1} \sin(\alpha_{T1} - \theta_2) - v_{A1} \sin(\alpha_{A1} - \theta_2)\right), \tag{3.7}$$

$$\dot{\theta}_3 = \frac{1}{r_2}\left(v_{T2} \sin(\alpha_{T2} - \theta_3) - v_{A2} \sin(\alpha_{A2} - \theta_3)\right), \tag{3.8}$$

$$\dot{\alpha}_{A1} = -\kappa(\alpha_{A1} - \theta_2)/v_{A1}, \tag{3.9}$$

$$\dot{\alpha}_{A2} = -\kappa(\alpha_{A2} - \theta_3)/v_{A2}, \tag{3.10}$$

$$\dot{\alpha}_{T1} = u_1, \tag{3.11}$$

$$\dot{\alpha}_{T2} = u_2, \tag{3.12}$$

where $\theta_1,\ \theta_2,\ \theta_3$ are the line-of-sight (LOS) angles between $A1-A2$, $T1-A1$, and $T2-A2$ respectively. R is the distance between the attackers $A1$ and $A2$, r_1 is the distance between $A1$ and $T1$ while r_2 is the distance between $A2$ and $T2$. The attacker model is defined using pure-pursuit (PP) guidance law with proportionality constant κ. u_1 and u_2 are the control inputs of $T1$ and $T2$. The angles δ and λ are

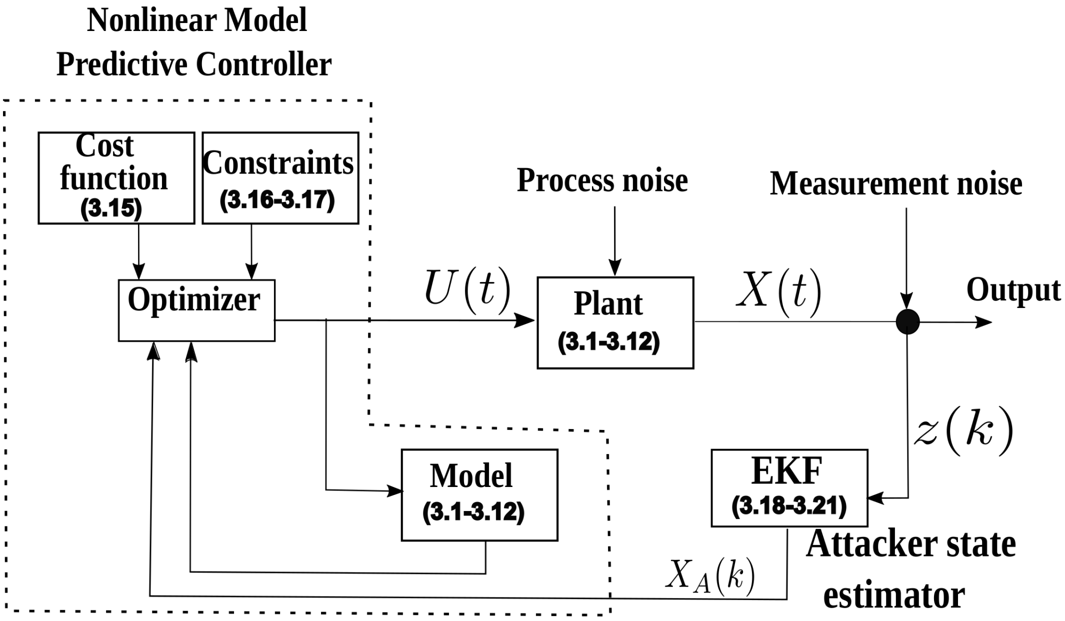


Figure 3.2: Block diagram of the proposed NMPC scheme with EKF for the 2T2A problem.

defined as [58]

$$\delta = \alpha_{A1} - \theta_1, \tag{3.13}$$

$$\lambda = \pi + (\alpha_{A2} - \theta_1). \tag{3.14}$$

We will now formulate the NMPC framework for solving the 2T2A problem using the defined state model and the equations of motion.

3.3 NMPC formulation

We propose a strategy wherein the target team uses an NMPC scheme to compute their control commands so as to lure the attackers into a collision course. Fig. 3.2 shows the structure of the proposed NMPC scheme. The main components of the controller are a mathematical model of the system (3.1)-(3.12), a nonlinear convex optimizer, a cost function, and physical constraints on the states and controls. The plant represents the four-agent system, and the EKF estimates the attacker states. The receding horizon approach combined with state feedback helps in re-planning the control sequence to counter the uncertainties involved in real-world systems.

To make the attackers collide, we need to minimize the distance between them (R) and the angles

δ, λ to keep their velocity vectors on the $A1 - A2$ line-of-sight (LOS). The distances r_1 and r_2 should be maximized so that the targets can evade the attackers. The objective function for the NMPC can be formulated as follows:

$$\min_{u_1, u_2 \in \mathcal{PC}(t, t+\tau_h)} J,$$
$$J = \int_t^{t+\tau_h} \left(w_1 R^2 + w_2 \delta^2 + w_3 \lambda^2 - w_4 r_1^2 - w_5 r_2^2 \right) \mathrm{d}t, \tag{3.15}$$

subject to:

$$\dot{X} = f(X, U, t), \tag{3.16}$$
$$U \in \left[U^-, U^+ \right], \tag{3.17}$$

where $X = \{x_{T1}, y_{T1}, x_{T2}, y_{T2}, x_{A1}, y_{A1}, x_{A2}, y_{A2}, R, r_1, r_2, \theta_1, \theta_2, \theta_3, \alpha_{A1}, \alpha_{A2}, \alpha_{T1}, \alpha_{T2}\}$ are the states, and $U = \{u_1, u_2\}$ are the control commands computed for the targets $T1$ and $T2$. The U^- and U^+ are the lower and upper bounds of U, $\mathcal{PC}(t, t + \tau_h)$ denotes the space of piece-wise continuous function defined over the time interval $[t, t + \tau_h]$, and $w_{i,i=1...5}$ are the weights. The weights should be tuned to ensure zero miss distance between the attackers. If the values of w_2, w_3 are too high compared to the weight w_1, it will result in non-zero miss distance, whereas too low values for w_2, w_3 will increase the risk of attackers capturing the targets. The values of w_4, w_5 should not be too high since the primary objective is to make the attackers collide. Since the attacker states are unknown to the target pair, they are estimated using an EKF.

Extended Kalman Filter

The structure of the EKF is defined similarly to Sec. 2.3.2. The attacker states $\{x_{A1}, y_{A1}, \alpha_{A1}, x_{A2}, y_{A2}, \alpha_{A2}\}$ and controls $\{a_{A1}, a_{A2}\}$ need to be estimated, hence the estimation model is represented with dynamics [129, 130]

$$f_A = \begin{bmatrix} \dot{x}_{A1} \\ \dot{y}_{A1} \\ \dot{x}_{A2} \\ \dot{y}_{A2} \\ \dot{\alpha}_{A1} \\ \dot{\alpha}_{A2} \\ \dot{a}_{A1} \\ \dot{a}_{A2} \end{bmatrix} = \begin{bmatrix} v_{A1}\cos\alpha_{A1} \\ v_{A1}\sin\alpha_{A1} \\ v_{A2}\cos\alpha_{A2} \\ v_{A2}\sin\alpha_{A2} \\ \frac{a_{A1}}{v_{A1}} \\ \frac{a_{A2}}{v_{A2}} \\ -a_{A1} \\ -a_{A2} \end{bmatrix}, \tag{3.18}$$

and the Jacobian of f_A is $\nabla F_{X_A} = [\mathbf{0}_{8\times 4}\; F_{8\times 4}]$, where

$$F = \begin{bmatrix} -v_{A1}\sin\alpha_{A1} & 0 & 0 & 0 \\ v_{A1}\cos\alpha_{A1} & 0 & 0 & 0 \\ 0 & -v_{A2}\sin\alpha_{A2} & 0 & 0 \\ 0 & v_{A2}\cos\alpha_{A2} & 0 & 0 \\ 0 & 0 & \frac{1}{v_{A1}} & 0 \\ 0 & 0 & 0 & \frac{1}{v_{A2}} \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}. \tag{3.19}$$

We assume that the quantities $R, r_1, r_2, \theta_1, \theta_2$ and θ_3 can be measured and the measurement model is given as

$$h = \begin{bmatrix} \sqrt{(x_{T1}-x_{A1})^2 + (y_{T1}-y_{A1})^2} \\ \sqrt{(x_{T2}-x_{A2})^2 + (y_{T2}-y_{A2})^2} \\ \sqrt{(x_{A2}-x_{A1})^2 + (y_{A2}-y_{A1})^2} \\ \tan^{-1}\left(\frac{y_{T1}-y_{A1}}{x_{T1}-x_{A1}}\right) \\ \tan^{-1}\left(\frac{y_{T2}-y_{A2}}{x_{T2}-x_{A2}}\right) \\ \tan^{-1}\left(\frac{y_{A2}-y_{A1}}{x_{A2}-x_{A1}}\right) \end{bmatrix}. \tag{3.20}$$

The Jacobian of the measurement model is given by $\nabla H_{X_A} = [H_{6\times 4}\ \mathbf{0}_{6\times 4}]$, where

$$H = \begin{bmatrix} \frac{-(x_{T1}-x_{A1})}{r_1} & \frac{-(y_{T1}-y_{A1})}{r_1} & 0 & 0 \\ 0 & 0 & \frac{-(x_{T2}-x_{A2})}{r_2} & \frac{-(y_{T2}-y_{A2})}{r_2} \\ \frac{-(x_{A2}-x_{A1})}{R} & \frac{-(y_{A2}-y_{A1})}{R} & \frac{x_{A2}-x_{A1}}{R} & \frac{y_{A2}-y_{A1}}{R} \\ \frac{y_{T1}-y_{A1}}{r_1^2} & \frac{-(x_{T1}-x_{A1})}{r_1^2} & 0 & 0 \\ 0 & 0 & \frac{y_{T2}-y_{A2}}{r_2^2} & \frac{-(x_{T2}-x_{A2})}{r_2^2} \\ \frac{y_{A2}-y_{A1}}{R^2} & \frac{-(x_{A2}-x_{A1})}{R^2} & \frac{-(y_{A2}-y_{A1})}{R^2} & \frac{x_{A2}-x_{A1}}{R^2} \end{bmatrix}. \tag{3.21}$$

The NMPC scheme for the 2T2A problem formulated so far does not give us the answer to the question of whether the targets would be captured or not, given the initial positions of the attackers and the targets. In reality, the formulation would give optimal trajectories for the target survival only in a subset of the complete problem, where the targets are guaranteed to be successful. In the next section, we derive the conditions that would help us determine the answer to the escape problem.

3.4 Escape region

We now analyze the 2T2A problem to see if the outcome of the target capture or escape could be determined by the initial positions of the agents. The NMPC control law defined in Sec. 3.3 is independent of the results derived in this section. We use the concept of Apollonius circles to determine the escape region for the targets in the Cartesian plane subject to the following assumptions.

Assumption 8. *The attackers are identical with equal speeds.*

Assumption 9. *The targets have equal speed and are slower than the attackers. Otherwise, the targets can always evade the attackers.*

Assumption 10. *The attackers are assumed to pursue the targets closer to them at time* $t = 0$*. i.e.,* $x_{T1}(t = 0) > 0$ *and* $x_{T2}(t = 0) < 0$*. Once the attacker-target pair is assigned, they remain the same for all future times.*

To make the analysis easier, we modify the engagement geometry reference frame, as shown in Fig. 3.3a. The $x-$axis is defined as the line joining the coordinates of the attackers, $A1$ and $A2$, and the $y-$axis is defined as the perpendicular bisector of the line segment $\overline{A1A2}$, where $A1, A2, T1, T2$ represent the positions of the attackers and the targets respectively.

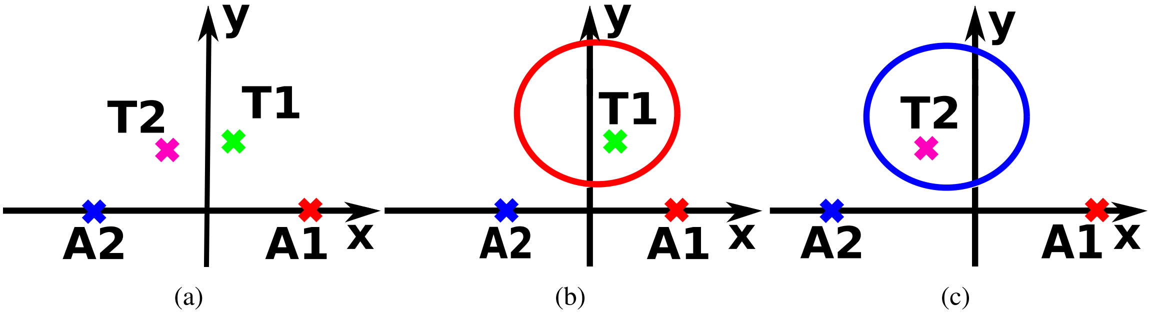


Figure 3.3: Modified reference frame considering the line joining $A1$ and $A2$ as the x-axis.

Lemma 3. *The locus of points where the attackers* $A1$ *and* $A2$ *can reach simultaneously is represented by the* $y-$*axis in the modified reference frame.*

Proof. Based on Assumption 8, the locus of points where the two agents can reach simultaneously is the orthogonal bisector of the line segment joining the agents [36]. Since $y-$axis is defined as the orthogonal bisector of the line segment $\overline{A1A2}$, it is the locus of points where the attackers $A1$ and $A2$ can reach simultaneously. ☐

Fig. 3.3b represents an example Apollonius circle for the $A1-T1$ engagement, and Fig. 3.3c for the $A2-T2$ engagement. Apollonius circles represent the locus of points where the two agents can reach simultaneously. For example, in Fig. 3.3b, the interior of the Apollonius circle is the set of positions that $T1$ can reach before $A1$, the points on the circle can be reached by $A1$ and $T1$ simultaneously, and $A1$ reaches the exterior positions before $T1$. The Apollonius circle for the $A1-T1$ engagement can be constructed using the center and radius given by [36]

$$(x_c, y_c)_{A1T1} = \left(\frac{x_{T1} - \gamma_{A1T1}^2 x_{A1}}{1-\gamma_{A1T1}^2}, \frac{y_{T1}}{1-\gamma_{A1T1}^2} \right), \tag{3.22}$$

and

$$d_{A1T1} = \frac{\gamma_{A1T1}\sqrt{(x_{T1}-x_{A1})^2 + y_{T1}^2}}{1-\gamma_{A1T1}^2}, \tag{3.23}$$

where γ_{A1T1} is the speed ratio defined by $\gamma_{A1T1} = \frac{v_{T1}}{v_{A1}}$, and similarly the Apollonius circle for the $A2-T2$ engagement is constructed using the center and radius given by

$$(x_c, y_c)_{A2T2} = \left(\frac{x_{T2} - \gamma_{A2T2}^2 x_{A2}}{1-\gamma_{A2T2}^2}, \frac{y_{T2}}{1-\gamma_{A2T2}^2} \right), \tag{3.24}$$

and

$$d_{A2T2} = \frac{\gamma_{A2T2}\sqrt{(x_{T2} - x_{A2})^2 + y_{T2}^2}}{1 - \gamma_{A2T2}^2}, \tag{3.25}$$

where γ_{A2T2} is the speed ratio defined by $\gamma_{A2T2} = \frac{v_{T2}}{v_{A2}}$. Now, let us state the theorem for finding the target escape region for the 2T2A problem.

Theorem 2. *The escape region for the targets exists if and only if the two Apollonius circles defined by* $A1 - T1$ *and* $A2 - T2$ *engagement have a common interception point with the* $y-$*axis. i.e., the target pair will be able to successfully lure the attackers into a collision if and only if the following conditions are satisfied:*

1. $Y_1 \cap Y \neq \emptyset$,
2. $Y_2 \cap Y \neq \emptyset$,
3. $Y_1 \cap Y_2 \neq \emptyset$,

where Y *is the set of all points on the* $y-$*axis,* Y_1 *is the set of all points on the* $A1 - T1$ *Apollonius circle,* Y_2 *is the set of all points on the* $A2 - T2$ *Apollonius circle, and* $\emptyset$ *is the null set.*

Proof. For finding the escape region for the targets, the 2T2A problem is divided into two sub-problems containing three agents each. More information on the three-agent problem is given in [54]. These new sub-problems will contain (i) attackers A1–A2 and target T1 and (ii) attackers A1–A2 and target T2.

First, consider the $A1 - A2 - T1$ problem. The $A1 - T1$ circle represents the points at which $A1$ and $T1$ can reach simultaneously, and the $y-$axis represents the points at which $A1$ and $A2$ can reach simultaneously. According to Lemma 3, $A2$ will collide with the $A1$ only if the $A1 - T1$ Apollonius circle intercepts the $y-$axis. If $Y_1 \cap Y = \emptyset$, then $T1$ will be captured by $A1$ before it can cross the $y-$axis and escape. Therefore $1)$ $Y_1 \cap Y \neq \emptyset$ must be satisfied.

Next, consider the $A1 - A2 - T2$ problem. Similarly to the previous case, we can see that if $Y_2 \cap Y = \emptyset$, then $T2$ will be captured by $A2$ before it can cross the $y-$axis and escape. Therefore 2) $Y_2 \cap Y \neq \emptyset$ must be satisfied.

Finally, we consider the complete $A1 - A2 - T1 - T2$ problem. Since each target depends on the other attacker for survival, the $A1 - T1$ and $A2 - T2$ circles should intercept each other on the $y-$ axis for $A1 - A2$ collision. Otherwise, if $Y_1 \cap Y_2 = \emptyset$, even if conditions 1 and 2 are satisfied, $A1$ and $A2$

will cross the $y-$axis at different points, and hence they will not collide. Therefore, for the attackers to collide and the targets to escape, conditions 1, 2, and 3 should be simultaneously satisfied. ☐

Remark 1. *Theorem 2 is devised assuming optimal play by the agents. Also, it does not consider the headings of the targets and the attackers. The escape region would be much larger if the attackers were initially pointing away from the targets and had bounded yaw rates.*

From Fig. 3.4a, the equation of the $A1 - T1$ Apollonius circle can be written as

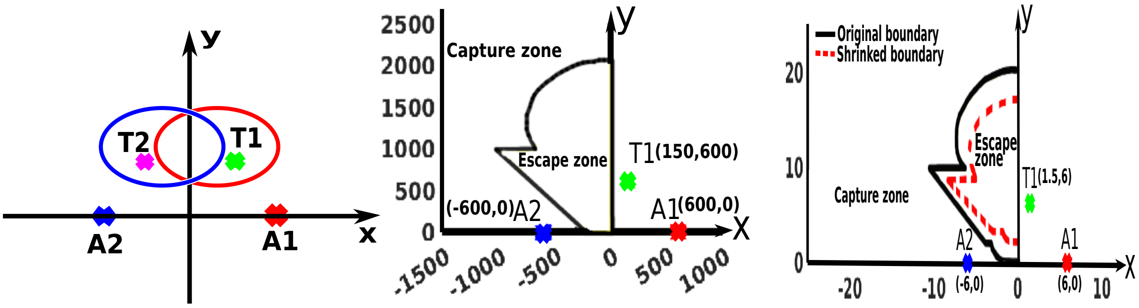
$$\left(x - \left(\frac{x_{T1} - \gamma_{A1T1}^2 x_{A1}}{1 - \gamma_{A1T1}^2}\right)\right)^2 + \left(y - \left(\frac{y_{T1} - \gamma_{A1T1}^2 y_{A1}}{1 - \gamma_{A1T1}^2}\right)\right)^2 = \left(\frac{\gamma_{A1T1}\sqrt{(x_{T1} - x_{A1})^2 + (y_{T1} - y_{A1})^2}}{1 - \gamma_{A1T1}^2}\right)^2. \quad (3.26)$$

Since $A1$ lies on the $x-$axis, $y_{A1} = 0$. For finding the intersection points of $A1 - T1$ circle with the $y-$axis, we put $x = 0$, and the following expression is obtained.

$$y^2 - 2y\left(\frac{y_{T1}}{1 - \gamma_{A1T1}^2}\right) + \left(\frac{x_{T1} - \gamma_{A1T1}^2 x_{A1}}{1 - \gamma_{A1T1}^2}\right)^2 + \left(\frac{y_{T1}}{1 - \gamma_{A1T1}^2}\right)^2 - \frac{\gamma_{A1T1}^2\left((x_{T1} - x_{A1})^2 + y_{T1}^2\right)}{\left(1 - \gamma_{A1T1}^2\right)^2} = 0. \quad (3.27)$$

Interception points of $A1 - T1$ Apollonius circle with the $y-$axis are the solutions of this quadratic equation (3.27) and are represented by $\overline{y}_1$ and $\underline{y}_1$. Similarly, the interception points of $A2 - T2$ Apollonius circle with the $y-$axis can be represented using $\overline{y}_2$ and $\underline{y}_2$. The escape region for the targets can be mapped by verifying the conditions given in Theorem 2 by analyzing the positions of $y-$intercepts: $\overline{y}_1, \underline{y}_1, \overline{y}_2$, and $\underline{y}_2$.

Consider an example agent configuration shown in Fig. 3.4a. The initial positions of the agents and the escape region for the targets are shown in Fig. 3.4b. The agent speed ratios were taken as $\gamma_{A1T1} = \gamma_{A2T2} = 0.5$ and the initial position of $T2$ was varied between $-1500 < x_{T2} < 0$ and $0 < y_{T2} < 2500$. At each selected initial point of $T2$, the conditions stated in Theorem 2 was checked by comparing the $y-$intercepts: $\overline{y}_1, \underline{y}_1, \overline{y}_2$, and $\underline{y}_2$, and finally, the dividing curve was drawn. If the $T2$'s initial position lies inside the escape region, then the targets would escape, and if it lies outside of the escape region, the targets would be captured.



(a) (b) (c)

Figure 3.4: (a) Apollonius circles for the 2T2A problem. (b) Escape region for the targets. (c) Stochastic escape zone for the targets. The dashed red line shows the change in the escape region due to uncertainty in attacker positions.

Stochastic escape region

The above analysis assumes the true position of the attackers. If the attacker state estimation is considered, the escape region will shrink or expand based on the estimation error. This change can be quantified using $\pm 3\sigma$ bounds of the EKF. For example, consider an initial configuration given in Fig. 3.4c with the initial position of $T2$ varied between $-20 < x_{T2} < 0$ and $0 < y_{T2} < 25$. Taking a conservative approach with $-3\sigma = 5\,\mathrm{m}$ for the attackers and solving equation (3.27), we obtain a reduced escape region as shown in Fig. 3.4c. The initial positions of the agents were selected as $x_{A1} = 6, y_{A1} = 0, x_{A2} = -6, y_{A2} = 0, x_{T1} = 1.5, y_{T1} = 6$. It is difficult to obtain deterministic escape regions while taking the state estimation into account, and it is a topic for future research.

3.5 Results and Discussion

3.5.1 Simulation setting

The simulations were carried out using CasADi(3.5.5)-Python(3.8) [137] on an Ubuntu 20.04 system with Intel i5-7200U CPU and 8GB RAM. The horizon, τ_h is 2.5 s for all simulations (prediction window of 50 steps with a sampling time of 0.05 s). The continuous kinematics (3.15),(3.16) were discretized and integrated using the RK4 method. The optimal control problem was converted to an optimization problem using the multiple-shooting method and was solved using the interior point optimization (IPOPT) library available in CasADi. The following options were used in the optimizer max_iter : 2000, $acceptable\_tol$: $1e^{-8}$, $acceptable\_obj\_change\_tol$: $1e^{-6}$, $warm\_start\_init\_point$: yes. The average computation time taken for each NMPC re-planing step was 0.1 s. The state covariance matrix

Q and the measurement covariance matrix Σ for the EKF are selected as

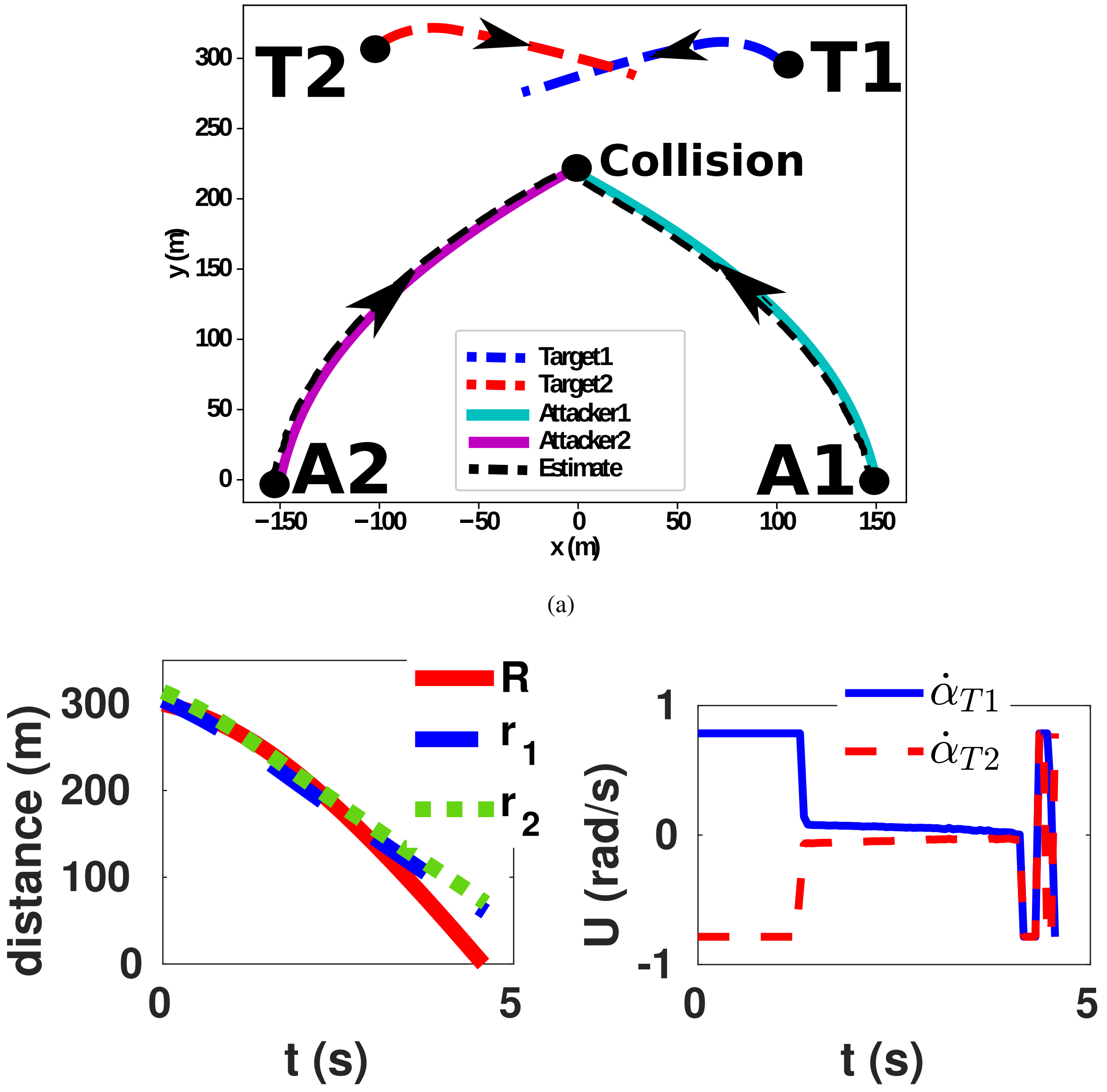
$$Q = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix},$$

$$\Sigma = \begin{bmatrix} 0.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 \end{bmatrix}.$$

The capture radii is $R_c = 1\,\text{m}$, $r_{1_c} = 1\,\text{m}$, and $r_{2_c} = 1\,\text{m}$. The weights were selected as $w_1 = 30, w_2 = w_3 = 60, w_4 = w_5 = 1$. Due to practical considerations on the turn rate, the angular velocities are constrained to $-0.785 \leq \{\dot{\alpha}_{T1}, \dot{\alpha}_{T2}\} \leq 0.785\,\text{rad/s}$. The velocities of the agents are taken as $v_{A1}, v_{A2} = 60\,\text{m/s}$ and $v_{T1}, v_{T2} = 30$ m/s. Proportionality constant for the attacker model, $\kappa = 3$.

3.5.2 An example scenario

We consider an engagement scenario as shown in Fig. 3.5a. The figure shows the agent trajectories for an initial configuration of $x_{T1} = 100, y_{T1} = 300, x_{T2} = -100, y_{T2} = 310, x_{A1} = 150, y_{A1} = 0, x_{A2} = -150, y_{A2} = 0$. In the simulation, we use an optimal guidance law (PN) [126] given by $a_{A1} = N v_{A1} \dot{\theta}_{A1T1}$, $a_{A2} = N v_{A2} \dot{\theta}_{A2T2}$, $N = 3$ for the attackers. The target pair was able to lure the attackers into a collision successfully even though the attackers used a guidance law (PN) different from the NMPC model (PP). The evolution of distances between the agents is shown in Fig. 3.5b. The distance between the attackers went to zero, confirming the collision. Control commands computed by NMPC for the target pair are shown in Fig. 3.5c. It can be seen that the input constraints have been satisfied. The estimator performance is shown in Fig. 3.6. The estimation errors in attacker positions are very low



(a)

(b)

(c)

Figure 3.5: An example scenario of the four-agent 2T2A problem. (a) The trajectories of the two targets and the two attackers. (b) Distance between different agents. (c) Control profile of the target pair.

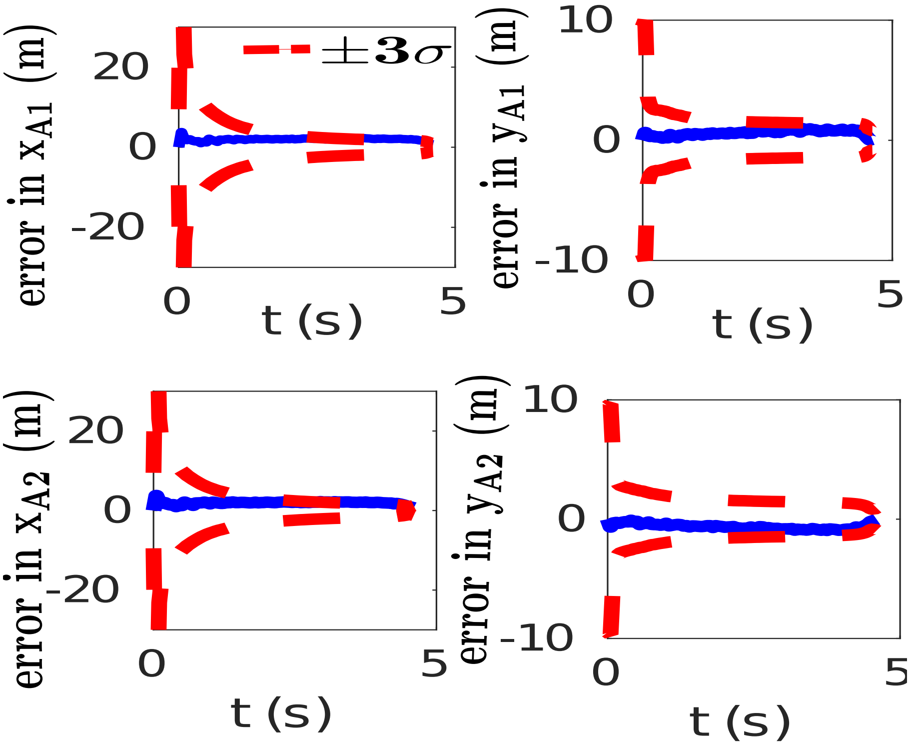


Figure 3.6: Estimation errors of the attacker states in 2T2A problem.

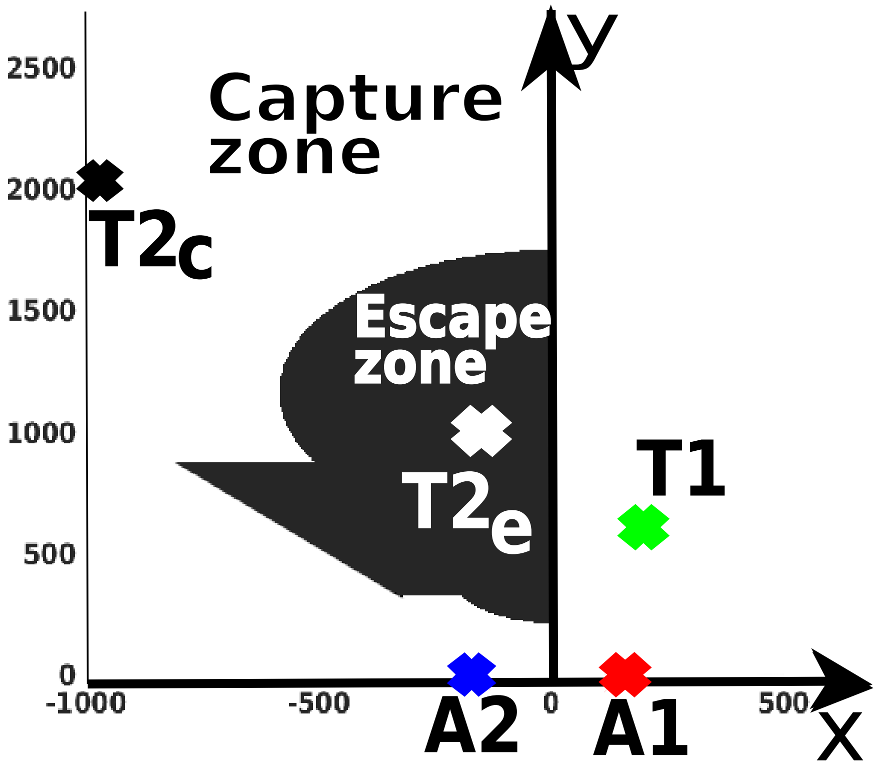
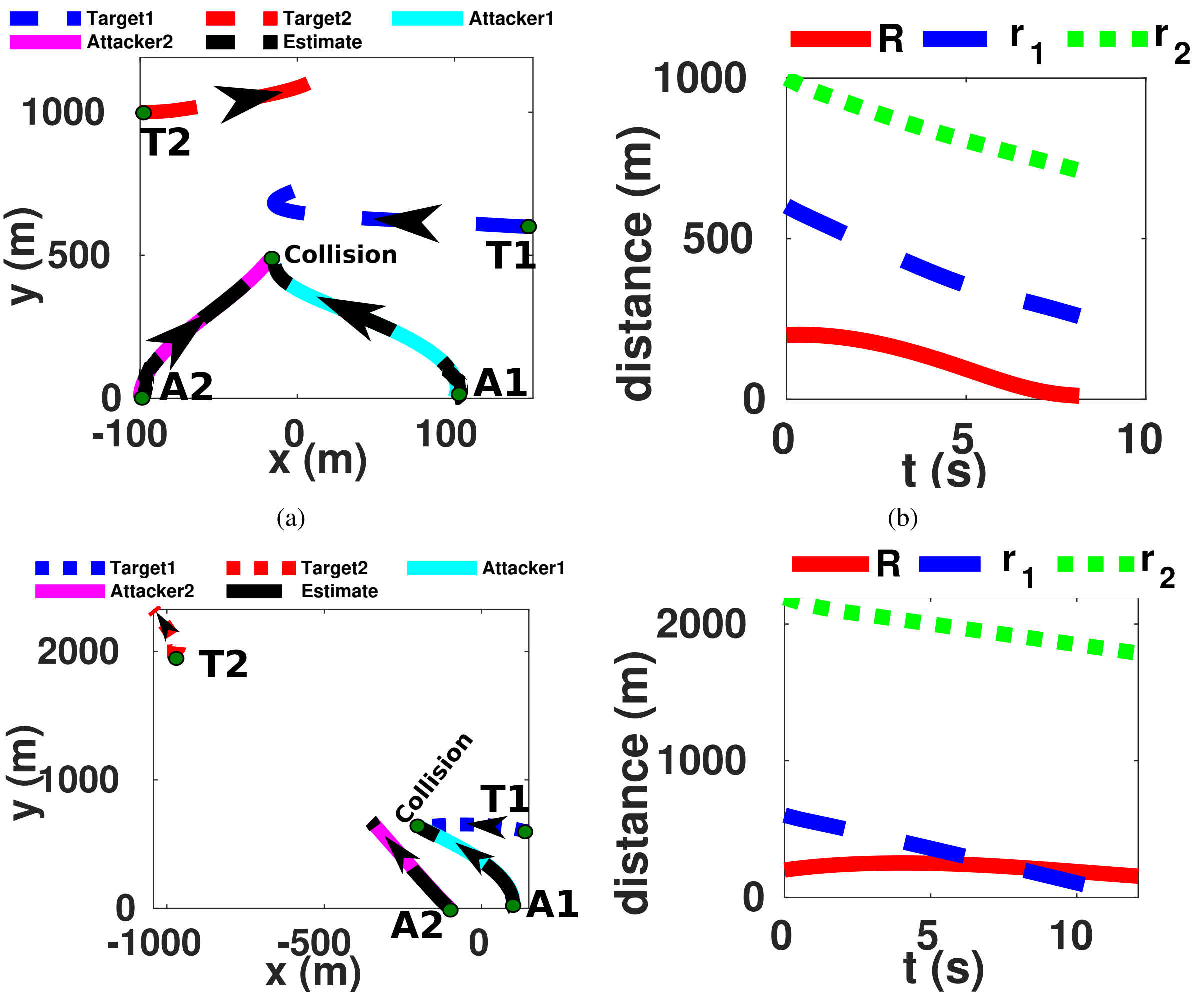


Figure 3.7: Initial agent configurations for the 2T2A problem in the escape map. The subscript c denotes capture and e escape.



(a)

(b)

(c)

(d)

Figure 3.8: (a) Agent trajectories for an example escape scenario. (b) Distance between the agents (escape). (c) Agent trajectories for an example capture scenario. (d) Distance between the agents (capture).

and stay within the 3σ bounds.

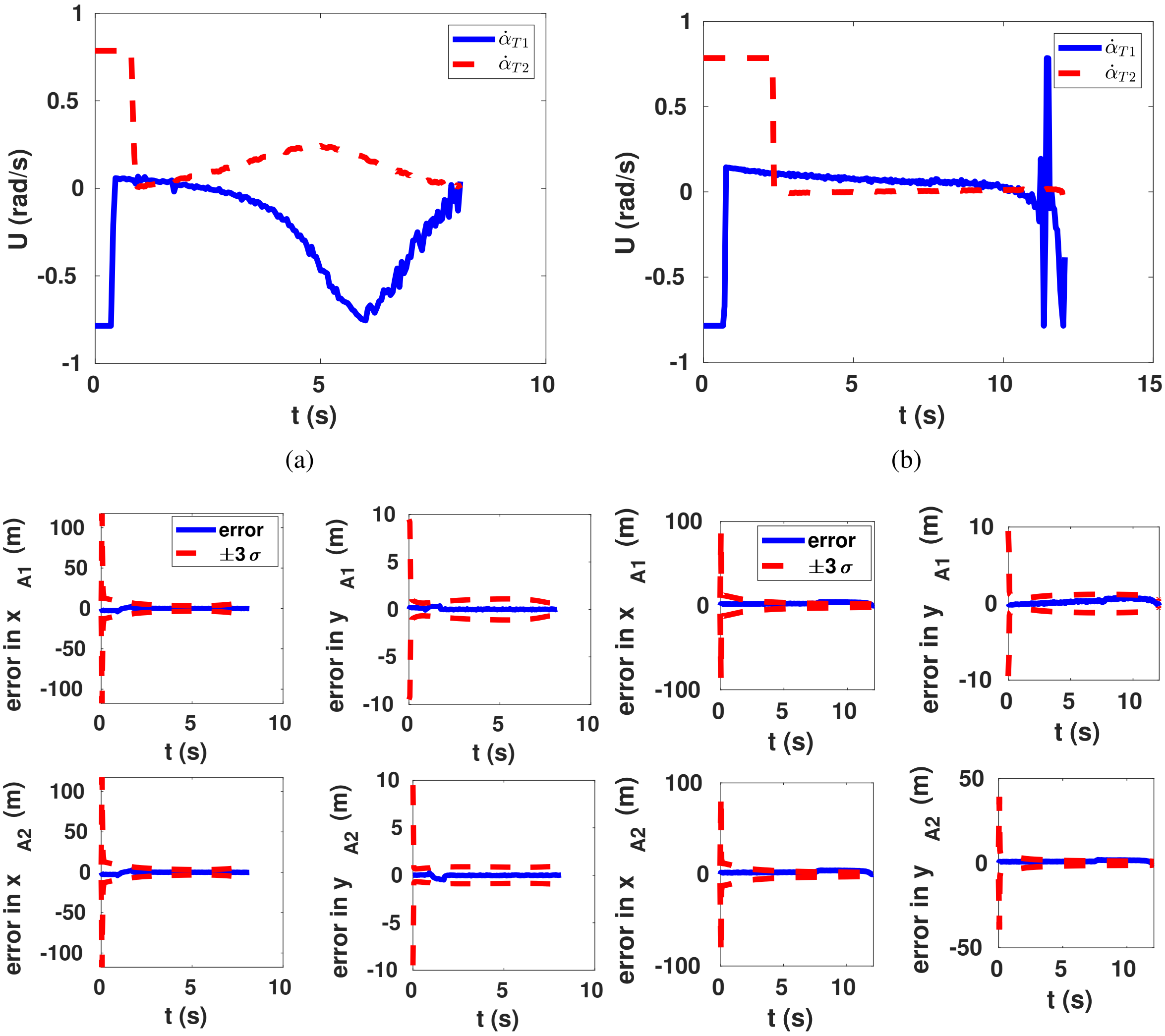
3.5.3 Examples to validate the theoretical analysis

We now show two examples based on the escape region given in Fig. 3.4b. We fix the positions for the attackers and $T1$, while the $T2$ position is changed in the simulations. First, the initial position of $T2$ is selected inside the escape region as $x_{T2} = -100, y_{T2} = 1000$ and is represented by $T2_e$ in Fig. 3.7. The agent trajectories are shown in Fig. 3.8a, where we can see the attackers colliding. Fig. 3.8b shows the distance (R) between the attackers converging to zero, while the attacker-target distances remain positive. Control commands for the target pair are shown in Fig. 3.5c. The shape of the input graph is similar for the targets since they are executing similar trajectories in opposite directions. The control values respect the defined bounds. The estimation errors in attacker positions are given in Fig. 3.9c. It can be seen that the errors in x and y positions of both the attackers are very low and is well within the σ bounds.

Next, the initial position of $T2$ is considered outside the escape region as $x_{T2} = -1000, y_{T2} = 2000$ and is represented by $T2_c$ in Fig. 3.7. Under this initial condition, Fig. 3.8c shows that one of the targets was captured, and as the engagement continues, the other target will also be captured. Fig. 3.8d shows the distance between the agents. We can see that the distance between $A1$ and $T1$, represented by r_1 goes to zero, confirming the capture of $T1$ by $A1$. Hence, the attackers won in this scenario as expected. The distance between the attackers, R is seen decreasing since the targets tried to make them on a collision course. But, it does not reach zero since one of the targets was captured by an attacker just before the attackers were about to collide. The distance r_2 is decreasing steadily, and eventually, the target T2 will also be captured by A2. Although the targets lost the game, the controller and the estimator performed satisfactorily, as seen in Fig. 3.5c and Fig. 3.9c. Some oscillations can be seen in the control command of $T1$ at the end since it desperately tries to avoid capture. The failure of the targets was not due to the estimation errors, as we can see from Fig. 3.9c. The errors are very low and are well within bounds.

3.6 Conclusions

In this work, we proposed a cooperative strategy based on NMPC for the active defense of the targets in a two–targets two–attackers (2T2A) problem. Theoretical analysis using Apollonius circles was performed to determine whether the targets would escape or not, given the initial conditions of the attackers and



(a) (b)

(c) (d)

Figure 3.9: (a) Control profile of the agents (escape). (b) Control profile of the agents (capture). (c) Estimation errors (escape). (d) Estimation errors (capture).

the targets. The efficacy of the proposed scheme was validated using numerical simulations. The results show the ability of the NMPC to determine control commands satisfying the state and control constraints. Also, the integration of the EKF with the NMPC performed well, as the estimation errors are within the 3σ bounds. The results also support the theoretical analysis.

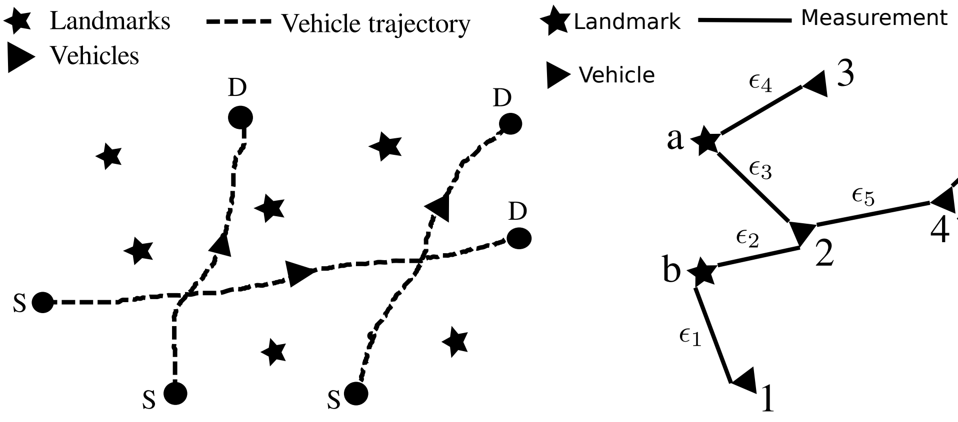
Chapter 4

Multi-AAV Cooperative Path Planning using NMPC with Localization Constraints

4.1 Introduction

We consider a scenario where a group of AAVs needs to navigate from their source location (S) to their destination (D), as shown in Fig. 4.1a. These vehicles need to transit in a GPS-denied area, and we assume that any kind of GNSS is not available. In such scenarios, known landmarks or other vehicles in the area can be used for relative localization using range or bearing measurements. This structure involving landmarks and vehicles can be modeled as a dynamic relative position measurement graph (RPMG) [138] with vehicles and landmarks as nodes and connections/measurements as edges.

Previous studies show that for cooperative localization to work, each vehicle should have a direct or indirect path to at least two known landmarks [139]. This condition is very limiting in environments with a low number of landmarks. Hence in this chapter, we find a relationship between vehicle uncertainty and path length to the landmarks and then use that relationship to formulate and solve an NMPC problem to guarantee that the covariance does not exceed a specified threshold and desired localization accuracy is achieved while performing individual missions. A moving horizon estimation (MHE) scheme is used to estimate the vehicle states.



(a) (b)

Figure 4.1: (a) Path planning scenario. (b) Relative position measurement graph with vehicles and landmarks as nodes and measurements as edges.

4.2 Problem formulation

Definition 2. *A relative position measurement graph (RPMG) for* $n_v(t)$ *vehicles with* $n_l(t)$ *landmarks is a graph* $G_{n_v}^{n_l}(t) \triangleq \{\mathcal{V}_{n_v}^{n_l}(t), \mathcal{E}_{n_v}^{n_l}(t)\}$*, where* $\mathcal{V}_{n_v}^{n_l}$ *is the node set consisting of* $n_v(t)$ *vehicle nodes and* $n_l(t)$ *landmark nodes (which makes a total of* $n_v(t) + n_l(t)$ *nodes), and* $\mathcal{E}_{n_v}^{n_l}(t)$ *is the edge set representing available relative measurements. The number of edges is denoted by* $n_e(t) = |\mathcal{E}_{n_v}^{n_l}(t)|$*. The graph is time-varying since the number of nodes will change depending on the number of connections possible due to the sensor range and availability of measurements.*

Definition 3. *A path from a vehicle node to a landmark node is a finite sequence of edges that joins a sequence of distinct vertices between them. Let* $\mathcal{G} \triangleq \{\mathcal{V}, \mathcal{E}, \phi\}$ *be a graph. A path* ϕ *from vertex* i *to vertex* j *is a sequence of edges* $\{\epsilon_1, \epsilon_2, \ldots, \epsilon_{n-1}\}$ *for which there is a sequence of distinct vertices* $\{\nu_1, \nu_2, \ldots, \nu_n\}$ *such that* $\phi \triangleq \{\nu_1, \nu_n\}$ *where* $\nu_1 = i$ *and* $\nu_n = j$*.*

An example RPMG (G_5^3 with $n_e = 7$) is shown in Fig. 4.1b. A path from the vehicle 5 to the landmark a is represented by the edge set $\phi \triangleq \{\epsilon_6, \epsilon_5, \epsilon_3\}$ which can also be represented using the vertices as $5 - 4 - 2 - a$. A graphical representation of the proposed solution using NMPC combined with MHE to tackle the cooperative localization and path planning problem is shown in Fig. 4.2. The optimal path is different from the shortest path since the latter may not satisfy localization constraints. The components of the block diagram are explained in the subsequent sections. In the first step of the control scheme, vehicle states are estimated by the MHE block using available measurements from the sensors (Sec. 4.3). The second step is the calculation of estimation covariances for the NMPC prediction window, which is accomplished by the covariance calculator block that contains the derived analytical

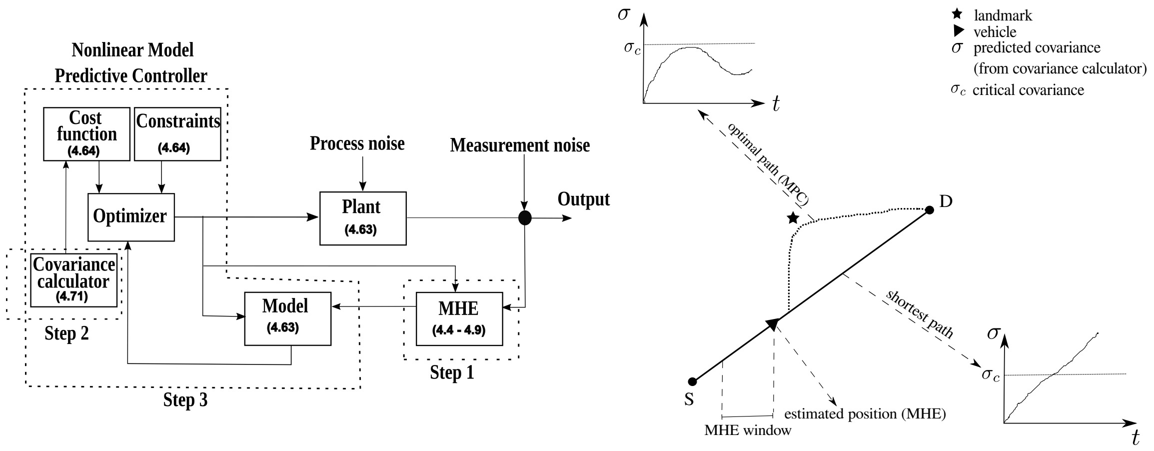


Figure 4.2: Block diagram and a graphical representation of the proposed NMPC-MHE control scheme for the localization and path planning problem.

expression (Sec. 4.4). In the third and final step, the NMPC controller computes the control actions for the vehicles (Sec. 4.5).

4.3 Moving horizon estimation

Moving horizon estimation (MHE) uses optimization techniques to determine state trajectories that best fit a series of measurements acquired over a finite time interval. It uses the exact nonlinear models of the available measurements and system dynamics for estimation. Also, there is another advantage of including the state/control constraints in the formulation, which helps in bounding the estimates. Like NMPC, MHE also has three main components, 1) an internal dynamic model of the process, 2) a history of past measurements, and 3) an optimization cost function over the estimation horizon.

The model used for estimation is given as:

$$X(k) = f\left(X(k-1), \boldsymbol{\omega}(k), k\right) + q(k), \tag{4.1}$$

$$\tilde{z}(k) = h\left(X(k), \boldsymbol{\omega}(k), k\right) + \mu(k), \tag{4.2}$$

where, $f(\cdot)$ and $h(\cdot)$ represent the nonlinear functions of states and observations, respectively. $X(k)$, $\boldsymbol{\omega}(k)$, and $\tilde{z}(k)$ are the predicted system states, controls, and measurements at the k^{th} time instant. The vectors $q(k)$ and $\mu(k)$ are the process and measurement noises which are assumed to be additive and zero mean white Gaussian noises with covariance Q and Γ respectively. $f(\cdot)$ is defined as

$$f = \begin{bmatrix} x_1(k) \\ y_1(k) \\ \psi_1(k) \\ \vdots \\ x_{n_v}(k) \\ y_{n_v}(k) \\ \psi_{n_v}(k) \end{bmatrix} = \begin{bmatrix} x_1(k-1) + T_s v \cos\psi_1(k-1) \\ y_1(k-1) + T_s v \sin\psi_1(k-1) \\ \psi_1(k-1) + T_s \omega_1(k-1) \\ \vdots \\ x_{n_v}(k-1) + T_s v \cos\psi_{n_v}(k-1) \\ y_{n_v}(k-1) + T_s v \sin\psi_{n_v}(k-1) \\ \psi_{n_v}(k-1) + T_s \omega_{n_v}(k-1) \end{bmatrix}, \tag{4.3}$$

where T_s is the sampling time used for discretization, v the linear velocity, ψ the heading angle, and ω the angular velocity.

Let m be the current time step, N_E is the estimation horizon, and we denote $\tau = m - N_E$ for simplicity. We formulate the moving horizon estimation problem

$$\begin{aligned} &\min_X J, \\ &J = \|X_\tau - \hat{X}_\tau\|^2_{P_\tau^{-1}} + \sum_{k=\tau}^{m} \|h(X_k) - z_k\|^2_{\Gamma^{-1}}, \end{aligned} \tag{4.4}$$

subject to:

$$X_{k+1} = f(X_k, \boldsymbol{\omega}_k), \tag{4.5}$$

$$X \in \left[X^-, X^+\right], \tag{4.6}$$

$$\boldsymbol{\omega} \in \left[\boldsymbol{\omega}^-, \boldsymbol{\omega}^+\right], \tag{4.7}$$

where $\hat{X}$ is the estimated states, P is the estimation covariance matrix, z_k is the actual measurement, and Γ is the measurement covariance. X^-, X^+ are the lower and upper bounds on the states, and $\boldsymbol{\omega}^-, \boldsymbol{\omega}^+$ are the lower and upper bounds of controls. The norms in the cost function are quadratic norms of the form $||x||_P = (x^T P x)^{1/2}$. Only the estimates of the current time step will be feedbacked to the NMPC controller. ie., $\hat{X}_m = X_m$. It is assumed that each vehicle can measure relative bearing to other vehicles and landmarks using a sensor (such as LiDar or radar) with a circular range, R_s. We do not consider sensor occlusions. Relative bearing from the i^{th} vehicle to the j^{th} vehicle or landmark is given by the measurement model:

$$h(X) = \tan^{-1}\left(\frac{y_j - y_i}{x_j - x_i}\right) - \psi_i. \tag{4.8}$$

The first term in (4.4) is known as the arrival cost, and it plays an important role in stabilizing the estimator. It penalizes the deviation of the first state in the moving horizon window and its previous estimate $\hat{X}_\tau$. The weighting matrix P is the covariance matrix given by [140]

$$P_{k+1} = Q + \nabla F_X(P_k - P_k \nabla H_X^T(\nabla H_X P_k \nabla H_X^T + \Gamma)^{-1} \nabla H_X P_k)\nabla F_X^T, \tag{4.9}$$

where Q is the state covariance matrix, and $\nabla F_X, \nabla H_X$ are the Jacobians of f and h. The second term in (4.4) penalizes the change in predicted measurements $h(X_k)$ from the actual measurements z_k.

Now, we look into the stability of the moving horizon estimator. The following assumptions are required to prove the stability result.

Assumption 11. *The initial state X_0 and the control input ω are such that, for any noise q, the system trajectory X lies in a compact set χ and ω in a compact set U.*

Assumption 12. *The functions f and h are C^2 functions w.r.t X on $co(\chi)$ for every $\omega \in U$, where $co(\chi)$ is the convex closure of χ.*

The observation map for a horizon $N_E + 1$ can be defined as

$$F^{N_E}(X, \omega, q) = \begin{bmatrix} h(X_\tau) \\ h \circ f^\tau(X_\tau) \\ . \\ . \\ . \\ h \circ f^{m-1} \circ \cdots f^\tau(X_\tau) \end{bmatrix}, \tag{4.10}$$

where $\circ$ is function composition. Then it is possible to re-write equation (4.2) as

$$z_\tau = F^{N_E}(X, \omega, q) + \mu_\tau, \tag{4.11}$$

and modify the cost function as

$$J_m(X_{\tau, \hat{X}_\tau}) = \|X_\tau - \hat{X}_\tau\|^2_{\mathbb{P}_\tau} + \|F^{N_E}(X_\tau, \omega_\tau, q_\tau) - z_\tau\|^2_{\mathbf{\Gamma}}, \tag{4.12}$$

where, $\mathbf{\Gamma} = I_{N_E+1} \otimes \Gamma^{-1}$, where $\otimes$ is the Kronecker product.

Now, let's consider the following remarks:

Remark 2. *System (4.1), (4.2) is said to be observable in* $N_E + 1$ *steps if there exists a* K*-function* $\phi(\cdot)$ *such that* $\phi\left(\|x_1 - x_2\|^2\right) \leq \|F^{N_E}(x_1, \omega, 0) - F^{N_E}(x_2, \omega, 0)\|^2$, $\forall x_1, x_2 \in \chi$ *and* $\forall \omega \in U^{N_E}$.

Remark 3. *If the observability matrix* $\frac{\partial F^{N_E}(X,\omega,0)}{\partial X}$ *has full rank, then the system is said to be observable in* $N_E + 1$ *steps with finite sensitivity* $1/\delta$ *if the* K*-function* $\phi(\cdot)$ *satisfies the following condition*

$$\delta = \inf_{x_1, x_2 \in \chi; x_1 \neq x_2} \frac{\phi\left(\|x_1 - x_2\|^2\right)}{\|x_1 - x_2\|^2} \geq 0. \tag{4.13}$$

Let k_f be an upper bound on the Lipschitz constant of $f(X, \omega)$ w.r.t X on χ for every $\omega \in U$ and $\mathbb{P}$ is diagonal with $\mathbb{P} = pI_n, p > 0$. Let

$$r_\mu = \max_{\mu \in M} \|\mu\|^2, \tag{4.14}$$

where M is a compact set with $0 \in M$.

The stability of the estimator is proved using the results from [140–143]. Consider the cost function defined as:

$$J = \|X_\tau - \hat{X}_\tau\|^2_{\mathbb{P}_\tau} + \sum_{k=\tau}^{m} \|h(X_k) - z_k\|^2_{\Gamma^{-1}}, \tag{4.15}$$

then we can state the following theorem [142, 143].

Theorem 3. *If the Assumptions 11,12 are satisfied and the Remarks 2,3 hold, then there exists an upper bound defined by*

$$\|X_\tau - \hat{X}_\tau\|^2 \leq \zeta_\tau, \tag{4.16}$$

where ζ_m *is found using the equation*

$$\zeta_{m+1} = \left(\frac{c_1 k_f p}{p + c_2 \delta}\right) \zeta_m + \left(\frac{c_3}{p + c_2 \delta}\right) r_\mu, \tag{4.17}$$

c_1, c_2*, and* c_3 *are positive constants. Let*

$$a(p, \delta) = \frac{c_1 k_f p}{(p + c_2 \delta)}, \tag{4.18}$$

and if p *is selected such that* $a(p, \delta) < 1$*, then the dynamics of (4.17) is asymptotically stable.*

Proof. A summary of the proof given by [142] is detailed here for completeness. The proof is based on defining upper and lower bounds on the optimal cost J^*_m, which is the cost corresponding to an optimal estimate $\hat{X}_\tau^*$. First, the upper bound on J_m^* should be defined. Let us define X_m^o as the true value of the state X at time m and assume that $\Gamma = I$. We have that

$$J_m^* \leq \|X_\tau^o - X_\tau^*\|_{\mathbb{P}_\tau}^2 + \sum_{k=\tau}^{m} \|F^{N_E}(X_k^o) - z_k\|^2, \tag{4.19}$$

which can be modified as

$$J_m^* \leq \|X_\tau^o - X_\tau^*\|_{\mathbb{P}_\tau}^2 + C, \tag{4.20}$$

where C is a positive constant (please see Lemma. 1 from [142]). Next, the upper bound on J_m^* is defined. We can write

$$\begin{aligned}
\|F^{N_E}(X_\tau^o) - F^{N_E}(\hat{X}_\tau)\|^2 &= \|[z_\tau - F^{N_E}(\hat{X}_\tau)] - [z_\tau - F^{N_E}(X_\tau^o)]\|^2, \\
\|F^{N_E}(X_\tau^o) - F^{N_E}(\hat{X}_\tau)\|^2 &\leq 2\|z_\tau - F^{N_E}(\hat{X}_\tau)\|^2 + 2\|z_\tau - F^{N_E}(X_\tau^o)]\|^2, \\
\|z_\tau - F^{N_E}(\hat{X}_\tau)\|^2 &\geq \frac{1}{2}\|F^{N_E}(X_\tau^o) - F^{N_E}(\hat{X}_\tau)\|^2 - \|z_\tau - F^{N_E}(X_\tau^o)]\|^2.
\end{aligned}$$

From (4.20), we can write

$$\|z_\tau - F^{N_E}(X_\tau^o)]\|^2 < C. \tag{4.21}$$

Hence, we obtain

$$\|z_\tau - F^{N_E}(\hat{X}_\tau)\|^2 \geq \frac{1}{2}\|F^{N_E}(X_\tau^o) - F^{N_E}(\hat{X}_\tau)\|^2 - C.$$

By using a similar procedure, we can write

$$\|X_\tau^* - \hat{X}_\tau\|^2 \geq \frac{1}{2}\|X_\tau^o - \hat{X}_\tau\|^2 - \|X_\tau^o - X_\tau^*\|^2.$$

Now the upper bound can be defined as

$$J_m^* \geq \frac{1}{2}\|X_\tau^o - \hat{X}_\tau^*\|_{\mathbb{P}_\tau}^2 + \frac{1}{2}\|F^{N_E}(X_\tau^o) - F^{N_E}(\hat{X}_\tau^*)\|^2 - \|X_\tau^o - X_\tau^*\|_{\mathbb{P}_\tau}^2 - C. \tag{4.22}$$

Now, by combining the bounds (4.20) and (4.22), and rewriting we get

$$\frac{1}{2}\|X_\tau^o - \hat{X}_\tau^*\|_{\mathbb{P}_\tau}^2 + \frac{1}{2}\|F^{N_E}(X_\tau^o) - F^{N_E}(\hat{X}_\tau^*)\|^2 \leq 2\|X_\tau^o - X_\tau^*\|_{\mathbb{P}_\tau}^2 + 2C.$$

According to the Remarks 2 and 3, the above equation can be written as

$$\|F^{N_E}(X_\tau^o) - F^{N_E}(\hat{X}_\tau^*)\|^2 = \phi\left(\|X_\tau^o - \hat{X}_\tau^*\|^2\right),$$

and

$$\delta\|X_\tau^o - \hat{X}_\tau^*\|^2 \leq \phi\left(\|X_\tau^o - \hat{X}_\tau^*\|^2\right).$$

Now, it is possible to define the bound on the estimation error as

$$\|X_\tau^o - \hat{X}_\tau^*\|^2 \leq \frac{4p}{p+\delta}\|X_\tau^o - X_\tau^*\|^2 + \frac{4}{p+\delta}C.$$

Using the Lipschitz continuity of $f(\cdot)$, it can be written that

$$\|X_\tau^o - X_\tau^*\|^2 = 2k_f\|X_{\tau-1}^o - \hat{X}_{\tau-1}^*\|^2 + 2r_\mu.$$

Hence,

$$\|X_\tau^o - \hat{X}_\tau^*\|^2 \leq \zeta_\tau.$$

It can also be deduced that if $\zeta_m < \zeta_{m-1}$ and $a(p,\delta) < 1$, then ζ_m tends to $\frac{\beta}{1-a(p,\delta)}$ as $m \longrightarrow +\infty$ (please see Theorem 1 from [141]), where $\beta = \left(\frac{c_3}{p+c_2\delta}\right) r_\mu$. $\square$

Now let's check if the Assumptions 11,12 and the Remarks 2,3 hold for the system (4.4). From equations (4.5)-(4.7), it can be seen that the Assumption 11 is satisfied. Assumption 12 also holds since (4.3) and (4.8) are C^2 functions. The observability conditions given in Remarks 2,3 will be satisfied if the vehicles have connections to at least two landmarks [139]. We will be discussing how to ensure this condition to hold in Sec. 4.5. $\mathbb{P}$ is selected as equal to the covariance matrix P, which is positive semi-definite. Hence, according to Theorem 3, $\|X_\tau - \hat{X}_\tau\|^2$ is bounded by a positive value ζ_τ.

The following section presents the derivation of the analytical expression for calculating the state error covariance/estimation covariance matrix (P) using the path information. The covariance P is a measure of uncertainty in the estimated states, given that the models are accurate. This result will be later used for predicting covariances for the NMPC cost function.

4.4 Covariance (P) calculation

Consider an example configuration of two vehicles as shown in Fig. 4.3a(i), where the vehicles are represented by 1 and 2 and two landmarks by a and b. In order to understand how the paths/connections /measurements from a landmark to a vehicle influence the uncertainty of the vehicle states, we consider each landmark separately and analyze it. Consider the segment $a-1-2$ of the graph in Fig. 4.3a(i), as shown in Fig. 4.3a(ii). The observability matrix for the system can be written as

$$O = \begin{bmatrix} oa1 & 0 \\ o12 & -o12 \end{bmatrix}, \tag{4.23}$$

where $oa1, o12$, and $-o12$ are the derivatives of the measurements with respect to the vehicle states. For example, $oa1$ is the derivative of the measurement between the landmark a and the vehicle-1 with respect to the vehicle-1. Since there is no measurement between the landmark a and the vehicle-2, the corresponding entry (O_{12}) is zero. Note that the size of O depends on the number of edges, and it may not be a square matrix. Assuming the measurement covariance matrix $\Gamma = I$, and zero-mean white Gaussian noise, the observability Grammian is defined as $O^T\Gamma^{-1}O$, and the covariance matrix P is written as

$$\begin{aligned} P &\leq (O^T\Gamma^{-1}O)^{-1}, && (4.24)\\ &\leq \begin{bmatrix} \frac{1}{oa1^2} & \frac{1}{oa1^2} \\ \frac{1}{oa1^2} & \frac{1}{o12^2}+\frac{1}{oa1^2} \end{bmatrix}. && (4.25) \end{aligned}$$

The first element of the P matrix corresponds to the vehicle-1 connecting to the landmark a, hence $oa1$ (let us discard the square and fraction for easy understanding). The element $\frac{1}{o12^2}+\frac{1}{oa1^2}$ of the P matrix corresponds to the vehicle-2. Observing that it is connected to the landmark a through vehicle-1, we can see both $oa1$ and $o12$ is present in the entry. Next, we consider the section $1-2-b$ of the graph, as shown in Fig. 4.3a(iii). The observability and covariance matrices for the system are written as

$$O = \begin{bmatrix} 0 & ob2 \\ o12 & -o12 \end{bmatrix}, P \leq \begin{bmatrix} \frac{1}{o12^2}+\frac{1}{ob2^2} & \frac{1}{ob2^2} \\ \frac{1}{ob2^2} & \frac{1}{ob2^2} \end{bmatrix}. \tag{4.26}$$

The first element of P indicates that vehicle-1 is connected to the landmark b through vehicle-2. The last entry shows that vehicle-2 is directly connected to the landmark b, and hence only $ob2$ is present.

Next, we extend similar analysis for a three vehicle configuration as shown in Fig. 4.3b(i). Consider the section $3-a-1-2$ of the graph, shown in Fig. 4.3b(ii). The observability matrix and covariance

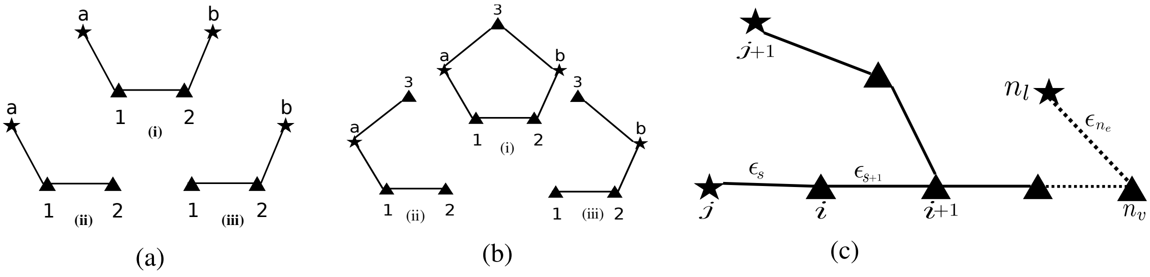


Figure 4.3: (a) Different configurations of two vehicles and two landmarks. (b) Different configuration of the system with 3 vehicles and 2 landmarks (c) Notations for a general multi-vehicle-landmark RPMG.

matrix P for this configuration are

$$O = \begin{bmatrix} oa1 & 0 & 0 \\ o12 & -o12 & 0 \\ 0 & 0 & oa3 \end{bmatrix}, \tag{4.27}$$

$$P \leq \begin{bmatrix} \frac{1}{oa1^2} & \frac{1}{oa1^2} & 0 \\ \frac{1}{oa1^2} & \frac{1}{o12^2} + \frac{1}{oa1^2} & 0 \\ 0 & 0 & \frac{1}{oa3^2} \end{bmatrix}. \tag{4.28}$$

The first element of the P matrix contains only $oa1$ since the vehicle-1 is directly connected to the landmark a. The vehicle-2 is connected to the landmark a through vehicle-1. Hence both $oa1$ and $o12$ can be seen in the corresponding entry of the P matrix. Since vehicle-3 is directly connected to the landmark, the last element of P contains only $oa3$ as expected. Now, let us take the section $1-2-b-3$ of the graph, shown in Fig. 4.3b(iii). The observability and covariance matrices are given by

$$O = \begin{bmatrix} 0 & ob2 & 0 \\ o12 & -o12 & 0 \\ 0 & 0 & ob3 \end{bmatrix}, \tag{4.29}$$

$$P \leq \begin{bmatrix} \frac{1}{o12^2} + \frac{1}{ob2^2} & \frac{1}{ob2^2} & 0 \\ \frac{1}{ob2^2} & \frac{1}{ob2^2} & 0 \\ 0 & 0 & \frac{1}{ob3^2} \end{bmatrix}. \tag{4.30}$$

Vehicle-2 and vehicle-3 are directly connected to the landmark b. Therefore, the corresponding entries in the P matrix contain only $ob2$ and $ob3$. The vehicle-1 is connected to the landmark b through vehicle-2, and this information is clearly reflected in the first entry of P.

Using the above example configuration analysis, we can further extend the analysis to a general result with n_v vehicle and n_l landmarks as shown in Fig. 4.3c. The landmarks are represented using stars, and the vehicles are represented using triangles. All the vehicles are denoted by $i, i+1, \ldots, n_v$, landmarks as $j, j+1, \ldots, n_l$, and edges connecting the vehicles and landmarks as $\epsilon_s, \epsilon_{s+1}, \ldots, \epsilon_{n_e}$, where $i = j = s = 1$. The observability vector associated with an edge/measurement is represented using ϵ with the edge number as a subscript for simplicity. For example, the observability vector between landmark-j and vehicle-i, oji is represented by ϵ_s. Now, the following theorem for a general RPMG can be stated.

Theorem 4. *The covariance associated with the vehicle-i, $i = 1, \ldots, n_v$, due to the landmark-j, $j = 1, \ldots, n_l$, is given by*

$$p_{ij} \leq \sum_{s \in \mathcal{S}} \frac{1}{\epsilon_s}, \tag{4.31}$$

where $s = 1, \ldots, n_e$, $\mathcal{S}$ is the set of edges that forms a path from vehicle-i to the landmark-j, n_v is the number of vehicles, n_l is the number of landmarks, and n_e is the number of edges in the RPMG.

Proof. The covariances associated with each vehicle for a two-vehicle-two-landmark configuration are given by equations (4.25) and (4.26), followed by three vehicles in (4.28) and (4.30). The relation given in Theorem 4 is clearly reflected in the elements of the corresponding covariance matrices. The generalization to n_v vehicles and n_l landmarks is straightforward from the previous analysis.

However, we prove the theorem through contradiction. Consider Fig. 4.3b(ii). According to Theorem 4, the term/edge $\frac{1}{oa1^2}$ should be present in p_{2a}. Suppose we write the P matrix without that term. The new P matrix and the original P matrix found by observability analysis given by (4.28) are written side-by-side, showing only the element corresponding to p_{2a}.

$$\begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & \frac{1}{o12^2} & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}, \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & \frac{1}{o12^2} + \frac{1}{oa1^2} & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}. \tag{4.32}$$

The new $O^T O$ matrix for the system found by inverting the first P matrix will be

$$O^T O = \begin{bmatrix} \frac{oa1^4}{-o12^2+oa1^2} & \frac{o12^2 oa1^2}{o12^2-oa1^2} & 0 \\ \frac{o12^2 oa1^2}{o12^2-oa1^2} & \frac{o12^2 oa1^2}{-o12^2+oa1^2} & 0 \\ 0 & 0 & \frac{1}{oa3^2} \end{bmatrix}, \tag{4.33}$$

which contradicts with the O^TO matrix derived from equation (4.28), which is

$$\begin{bmatrix} o12^2 + oa1^2 & -o12^2 & 0 \\ -o12^2 & o12^2 & 0 \\ 0 & 0 & oa3^2 \end{bmatrix}. \tag{4.34}$$

Similarly, if an additional term $\frac{1}{oa3^2}$ is present in p_{2a}, the corresponding O^TO matrix for the system will be

$$O^TO = \begin{bmatrix} oa1^2 + \frac{o12^2 oa3^2}{o12^2+oa3^2} & \frac{o12^2 oa3^2}{o12^2+oa3^2} & 0 \\ -\frac{o12^2 oa3^2}{o12^2+oa3^2} & \frac{o12^2 oa3^2}{o12^2+oa3^2} & 0 \\ 0 & 0 & oa3^2 \end{bmatrix}, \tag{4.35}$$

which also contradicts with the O^TO matrix given in (4.34). Hence, the relation given in Theorem 4 is always true. ☐

The following corollaries can be written from Theorem 4.

Corollary 1. *If there is more than one path from a landmark to a vehicle, and these paths are numbered from* 1 *to* g_{ij}*, where* g_{ij} *is the total number of paths from the landmark-*j *to the vehicle-*i*, then the total covariance of the vehicle is given by*

$$p_{ij} \leq \sum_{\gamma=1}^{g_{ij}} p_{ij}^{\gamma}, \tag{4.36}$$

where $i = 1, \ldots, n_v$*,* $j = 1, \ldots, n_l$*, and* p_{ij}^{γ} *is the covariance of the vehicle-*i *due to the landmark-*j *considering only the path-*γ*.*

Corollary 2. *If there is more than one landmark connected to a vehicle, then the covariance of the vehicle is given by*

$$p_i \leq \sum_{j \in \mathcal{J}} p_{ij}, \tag{4.37}$$

where $i = 1, \ldots, n_v$*,* $j = 1, \ldots, n_l$*, and* $\mathcal{J}$ *is the set of landmarks connected to the vehicle-*i*.*

Proof. The method to calculate covariances for Corollary 1 and 2 is given in Theorem 4. The summation is based on the properties of the information matrix given as follows [144, 145]

If $X = (X_1, X_2, \ldots, X_n)$ and $X_1, X_2, \ldots, X_n$ are independent random variables, then $I_X(\alpha) =$

$I_{X_1}(\alpha) + I_{X_2}(\alpha) + \ldots I_{X_n}(\alpha)$, where $I_x(\alpha)$ is the information matrix defined as

$$I_x(\alpha) = E_\alpha \left[\left(\frac{\partial}{\partial \alpha} \log f(X|\alpha) \right)^2 \right] = \mathrm{Var}_\alpha \left(\frac{\partial}{\partial \alpha} \log f(X|\alpha) \right).$$

Since

$$f(x|\alpha) = \prod_{i=1}^{n} f_i(x_i|\alpha),$$

where $f_i(\cdot|\alpha)$ is the pdf of X_i,

$$\begin{aligned} \mathrm{Var}\left[\frac{\partial}{\partial \alpha} \log f(X|\alpha) \right] &= \sum_{i=1}^{n} \mathrm{Var}\left[\frac{\partial}{\partial \alpha} \log f_i(X_i|\alpha) \right], \\ I_X(\alpha) &= \sum_{i=1}^{n} I_{X_i}(\alpha). \end{aligned}$$

Since covariance is the inverse of information, we can find the total covariance associated with each vehicle by adding the components from all the paths and landmarks. □

In the next sub-sections, we show how this information can be used to analyze the evolution of covariance in multi-vehicle-landmark systems with range and bearing measurements.

4.4.1 Range measurements

Even though we consider bearing-only measurements in our problem formulation since the range measurements are hard to obtain from a practical perspective, we start the analysis with the range measurements since it is less complex and easier to understand. We then extend the obtained results to the bearing measurement case.

Consider the configuration given in Fig. 4.3(a). The vehicle kinematics are defined as

$$\begin{aligned} \dot{x}_i &= v \cos \psi_i, \\ \dot{y}_i &= v \sin \psi_i, \end{aligned} \tag{4.38}$$

where $i = 1, 2$ with range measurements

$$\begin{aligned} h_{1a} &= \sqrt{(x_1 - x_a)^2 + (y_1 - y_a)^2}, && (4.39) \\ h_{12} &= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, && (4.40) \\ h_{2b} &= \sqrt{(x_2 - x_b)^2 + (y_2 - y_b)^2}, && (4.41) \end{aligned}$$

and we derive the observability matrix using the Lie derivatives [138]. To simplify the representation we denote $x_{1a} = (x_1 - x_a)$, $x_{12} = (x_1 - x_2)$, $x_{2b} = (x_2 - x_b)$, $y_{1a} = (y_1 - y_a)$, $y_{12} = (y_1 - y_2)$, $y_{2b} = (y_2 - y_b)$. Define

$$f_L = \begin{bmatrix} \cos\psi_1 \\ \sin\psi_1 \\ \cos\psi_2 \\ \sin\psi_2 \end{bmatrix}, \qquad (4.42)$$

and the vehicle kinematics can be represented as

$$\dot{X} = v f_L. \qquad (4.43)$$

The gradient of zero^th order Lie derivatives are given as

$$\begin{aligned} H_{1a} &= \begin{bmatrix} \frac{x_{1a}}{R_{1a}} & \frac{y_{1a}}{R_{1a}} & 0 & 0 \end{bmatrix}, H_{2b} = \begin{bmatrix} 0 & 0 & \frac{x_{2b}}{R_{2b}} & \frac{y_{2b}}{R_{2b}} \end{bmatrix}, \\ H_{12} &= \begin{bmatrix} \frac{x_{12}}{R_{12}} & \frac{y_{12}}{R_{12}} & \frac{-(x_{12})}{R_{12}} & \frac{-(y_{12})}{R_{12}} \end{bmatrix}, \end{aligned}$$

where $R_{(\cdot)}$ is the distance between nodes. The gradient of first order Lie derivatives is given as

$$\frac{\partial}{\partial X}\left(\frac{\partial h_{1a}}{\partial X} \cdot f\right) = \begin{bmatrix} \frac{(y_{1a})^2 C\psi_1 - (x_{1a})(y_{1a}) S\psi_1}{R_{1a}^3} \\ \frac{(x_{1a})^2 S\psi_1 - (x_{1a})(y_{1a}) C\psi_1}{R_{1a}^3} \\ 0 \\ 0 \end{bmatrix}',$$

$$\frac{\partial}{\partial X}\left(\frac{\partial h_{2b}}{\partial X} \cdot f\right) = \begin{bmatrix} 0 \\ 0 \\ \frac{(y_{2b})^2 C\psi_2 + (-x_{2b})(y_{2b}) S\psi_2}{R_{2b}^3} \\ \frac{(x_{2b})^2 S\psi_2 + (-y_{2b})(x_{2b}) C\psi_2}{R_{2b}^3} \end{bmatrix}',$$

$$\frac{\partial}{\partial X}\left(\frac{\partial h_{12}}{\partial X}\cdot f\right)=\begin{bmatrix}\frac{-2(y_{12})S(\frac{\Delta\psi^-}{2})((x_{12})C(\frac{\Delta\psi^+}{2})+(y_{12})S(\frac{\Delta\psi^+}{2}))}{R_{12}^3}\\ \frac{2(x_{12})S(\frac{\Delta\psi^-}{2})((x_{12})C(\frac{\Delta\psi^+}{2})+(y_{12})S(\frac{\Delta\psi^+}{2}))}{R_{12}^3}\\ \frac{2(y_{12})S(\frac{\Delta\psi^-}{2})((x_{12})C(\frac{\Delta\psi^+}{2})+(y_{12})S(\frac{\Delta\psi^+}{2}))}{R_{12}^3}\\ \frac{-2(x_{12})S(\frac{\Delta\psi^-}{2})((x_{12})C(\frac{\Delta\psi^+}{2})+(y_{12})S(\frac{\Delta\psi^+}{2}))}{R_{12}^3}\end{bmatrix}',$$

where sin and cos are abbreviated as S and C, $\Delta\psi^- = \psi_1 - \psi_2$, and $\Delta\psi^+ = \psi_1 + \psi_2$.

The observability matrix is formed by vertically stacking the Lie derivatives of each measurement up to the first order as

$$O = \begin{bmatrix}\nabla L^0\\ \nabla L^1\end{bmatrix}, \tag{4.44}$$

where L^0 and L^1 are the zero^th^ and first order Lie derivatives respectively. The covariance matrix P is found by inverting $O^T O$ with the assumption of $\Gamma = I$.

The standard deviation in x direction for the first vehicle can be found by taking the square root of the first element of the P matrix,

$$\sigma_{x_1} = \sqrt{P(1,1)}. \tag{4.45}$$

Similarly, the standard deviation in the y direction and the combined position uncertainty can be found as

$$\sigma_{y_1} = \sqrt{P(2,2)}, \tag{4.46}$$

$$\sigma_{p_1} = \sqrt{\sigma_{x_1}^2 + \sigma_{y_1}^2}. \tag{4.47}$$

Since the derived P matrix is very large with several terms, we consider some simplifying assumptions to formulate an approximate relation. All the distance terms were substituted with a single average value. The ratios of relative distances between the nodes were converted to trigonometric equivalents and substituted with an average of line-of-sight (LOS) angles between them. The resulting relation is given as

$$\sigma_{p_1} = \sqrt{\frac{2}{3} + R_g^2 \csc^2(\psi_1 - \theta_g)}, \tag{4.48}$$

where R_g, θ_g are the average values of the distances and the LOS angles. csc is the short form for cosec. It is evident from the relation that the covariance of the vehicle depends on the distances from the landmarks and the LOS angles to them. As the distance increases, the covariance also increases.

Similarly, σ_p of other vehicles can also be found.

4.4.2 Bearing measurements

Let's extend the vehicle model to contain three states. The new model is given as

$$\begin{aligned} \dot{x}_i &= v \cos \psi_i, \\ \dot{y}_1 &= v \sin \psi_i, \\ \dot{\psi}_i &= \omega_i, \end{aligned}$$

where vehicle $i = 1, 2$. With the same configuration as in Fig. 4.3(a), define bearing measurement equations as

$$\begin{aligned} h_{a1} &= \arctan \left(\frac{y_a - y_1}{x_a - x_1} \right) - \psi_1, && (4.49) \\ h_{b2} &= \arctan \left(\frac{y_b - y_2}{x_b - x_2} \right) - \psi_2, && (4.50) \\ h_{12} &= \arctan \left(\frac{y_1 - y_2}{x_1 - x_2} \right) - \psi_2, && (4.51) \\ h_{21} &= \arctan \left(\frac{y_2 - y_1}{x_2 - x_1} \right) - \psi_1, && (4.52) \end{aligned}$$

the gradient of zeroth order Lie derivatives are given as

$$\begin{aligned} H_{a1} &= \begin{bmatrix} \frac{-(y_1 - y_a)}{R_{a1}^2} & \frac{x_1 - x_a}{R_{a1}^2} & -1 & 0 & 0 & 0 \end{bmatrix}, && (4.53) \\ H_{b2} &= \begin{bmatrix} 0 & 0 & 0 & \frac{-(y_2 - y_b)}{R_{b2}^2} & \frac{x_2 - x_b}{R_{b2}^2} & -1 \end{bmatrix}, && (4.54) \\ H_{12} &= \begin{bmatrix} \frac{y_2 - y_1}{R_{12}^2} & \frac{x_1 - x_2}{R_{12}^2} & 0 & \frac{y_1 - y_2}{R_{12}^2} & \frac{x_2 - x_1}{R_{12}^2} & -1 \end{bmatrix}, && (4.55) \\ H_{21} &= \begin{bmatrix} \frac{y_2 - y_1}{R_{21}^2} & \frac{x_1 - x_2}{R_{21}^2} & -1 & \frac{y_1 - y_2}{R_{21}^2} & \frac{x_2 - x_1}{R_{21}^2} & 0 \end{bmatrix}, && (4.56) \end{aligned}$$

using the geometry, the equations are changed to make it in terms of the line-of-sight (LOS) angles (θ) between the nodes as follows

$$H_{a1} = \begin{bmatrix} \frac{\sin\theta_{a1}}{R_{a1}} & \frac{-\cos\theta_{a1}}{R_{a1}} & -1 & 0 & 0 & 0 \end{bmatrix}, \tag{4.57}$$

$$H_{b2} = \begin{bmatrix} 0 & 0 & 0 & \frac{\sin\theta_{b2}}{R_{b2}} & \frac{-\cos\theta_{b2}}{R_{b2}} & -1 \end{bmatrix}, \tag{4.58}$$

$$H_{12} = \begin{bmatrix} \frac{-\sin\theta_{12}}{R_{12}} & \frac{\cos\theta_{12}}{R_{12}} & 0 & \frac{\sin\theta_{12}}{R_{12}} & \frac{-\cos\theta_{12}}{R_{12}} & -1 \end{bmatrix}, \tag{4.59}$$

$$H_{21} = \begin{bmatrix} \frac{\sin\theta_{21}}{R_{21}} & \frac{-\cos\theta_{21}}{R_{21}} & -1 & \frac{-\sin\theta_{21}}{R_{21}} & \frac{\cos\theta_{21}}{R_{21}} & 0 \end{bmatrix}. \tag{4.60}$$

Define

$$\begin{aligned} f_v &= \begin{bmatrix} \cos\psi_1 & \sin\psi_1 & 0 & \cos\psi_2 & \sin\psi_2 & 0 \end{bmatrix}', \\ f_{\omega_1} &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}', \\ f_{\omega_2} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}', \end{aligned}$$

and the dynamics can be represented as

$$\dot{X} = \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = f_v v + f_{\omega_1}\omega_1 + f_{\omega_2}\omega_2, \tag{4.61}$$

and a similar procedure to the range measurement case is followed to find the first order Lie derivatives, observability grammian, and covariance matrix P. The covariance in the position of the first vehicle is given by

$$\sigma_{p_1}^2 = \frac{9}{2}R_g^2\left(1 + \frac{R_g^2}{2 + R_g^2 + 2\cos(2(\psi_1 - \theta_g))}\right) + (R_g^2 + R_g^4)\csc^2(\psi_1 - \theta_g). \tag{4.62}$$

Similarly, σ_p of other vehicles can also be found. This approximate closed form covariance is used in the NMPC as Step 2 in Fig. 4.2(a).

The next section presents the complete NMPC formulation combining the MHE scheme given in Sec. 4.3 and the uncertainty results derived from the analysis given in Sec. 4.4.

4.5 NMPC formulation

NMPC is a state-of-the-art technique for real-time optimal control. At each time step, the constrained optimization problem is solved based on the plant model for a finite time horizon, and the procedure is repeated with states updated through feedback in the next iteration [146]. Fig. 4.2 shows the block diagram of the NMPC scheme used in this paper. The optimal control sequence is computed for the prediction horizon τ_h, from which only the first action is applied to the system at each time step.

A point mass kinematic model is considered for the vehicles. We assume that the altitude and velocities of the AAVs remain constant during transit. The general kinematic model is given as

$$\dot{X} = \begin{bmatrix} v\cos\psi_1 \\ v\sin\psi_1 \\ \omega_1 \\ \vdots \\ v\cos\psi_{n_v} \\ v\sin\psi_{n_v} \\ \omega_{n_v} \end{bmatrix}, \tag{4.63}$$

where v is the linear velocity, ψ the heading angle, ω the angular velocity, and n_v is the number of vehicles.

The objective function of the NMPC is defined as

$$\begin{aligned} &\min_{\boldsymbol{\omega_1}\cdots\boldsymbol{\omega_{n_v}}\in\mathcal{PC}(t,t+\tau_h)} J, \\ &J = \int_t^{t+\tau_h} \sum_{i=1}^{n_v} \left[C_{1_i} + W_i C_{2_i}\right], \end{aligned} \tag{4.64}$$

subject to:

$$\begin{aligned} &\dot{X} = f(X, \boldsymbol{\omega}), \\ &X \in \left[X^-, X^+\right], \\ &\boldsymbol{\omega} \in \left[\boldsymbol{\omega}^-, \boldsymbol{\omega}^+\right], \end{aligned}$$

where

$$C_{1_i} = (x_i - x_{D_i})^2 + (y_i - y_{D_i})^2, \tag{4.65}$$

is the cost associated with minimizing the distance between the vehicle and the destination. (x_i, y_i) is the position of the i^{th} vehicle and (x_{D_i}, y_{D_i}) are their respective destination points. X^-, X^+ are the lower and upper bounds on the states, $\boldsymbol{\omega}^-$ and $\boldsymbol{\omega}^+$ are the lower and upper bounds of $\boldsymbol{\omega}$, and $\mathcal{PC}(t, t+\tau_h)$ denotes the space of piece-wise continuous function defined over the time interval $[t, t+\tau_h]$. C_{2_i} is the cost to ensure the estimation covariance is within a bound. It is defined as:

$$C_{2_i} = \begin{cases} 0, & \text{if } \lambda_i \geq \eta. \\ (\eta - \lambda_i)^2, & \text{otherwise.} \end{cases} \tag{4.66}$$

where η is a tuning parameter related to the number of connections required. Increasing η will result in vehicles moving closer to the landmarks and increasing the connections. For satisfactory localization, observability conditions should be satisfied, which require connections with at least two landmarks [139]. Hence, the value of η should be selected as $\eta \geq 2$. The parameter λ_i is the second smallest eigenvalue of the Laplacian matrix, which is formed as:

$$L_i(X) = \Delta_i(X) - A_i(X), \tag{4.67}$$

where $A_i(X)$ is the adjacency matrix defined similar to [97] as

$$A_{i_{mn}} = \begin{cases} e^{\frac{-\kappa(||R_{mn}|| - \rho)}{R_s - \rho}}, & ||R_{mn}|| \leq R_s. \\ 0, & ||R_{mn}|| > R_s. \end{cases} \tag{4.68}$$

and $\Delta_i(X)$ is a diagonal matrix with elements

$$\Delta_{i_{mm}} = \sum_{n=1}^{N} A_{i_{mn}} \tag{4.69}$$

where $m, n = 1$ to N, and N is the number of nodes of the graph connecting the vehicles and landmarks. κ is a constant that determines the convergence rate of the exponential function, and ρ is used to set a minimum distance between the landmarks and the vehicles to avoid collisions. $||R_{mn}||$ is the distance between the m^{th} and n^{th} nodes, and R_s is the sensor range of the vehicles. This formulation of the adjacency matrices, rather than updating it with binary values, helps drive the vehicles closer to the landmarks than just maintaining the connections by keeping them in the sensor range. Also, reducing the distance between the landmarks and the vehicles helps decrease the estimation covariance, as explained in sec. 4.4. The maximum value λ_i can take is equal to the number of connections of each vehicle, and this insight is used in formulating (4.66).

The weight W_i associated with C_{2_i} is defined in the following way

$$W_i = \begin{cases} W, & \text{if } 3\sigma_{p_i} \geq \sigma_c. \\ 0, & \text{otherwise.} \end{cases} \tag{4.70}$$

where σ_{p_i} is the standard deviation in the estimated position of the i^{th} vehicle and the constant σ_c is the specified critical value. This adaptive weight formulation is used to obtain a trade-off between the two objectives. σ_{p_i}s are calculated using the expression

$$\sigma_{p_i}^2 = \frac{9}{2}R_g^2\left(1 + \frac{R_g^2}{2 + R_g^2 + 2\cos(2(\psi_i - \theta_g))}\right) + (R_g^2 + R_g^4)\csc^2(\psi_i - \theta_g), \tag{4.71}$$

which is explained in detail in section 4.4.2. The terms C_1 and C_2 of the objective function (4.64) is normalized as follows:

$$C_1(t) = \left(\frac{C_1(t) - \min(C_1(t))}{\max(C_1(t)) - \min(C_1(t))}\right), \tag{4.72}$$

$$C_2(t) = \left(\frac{C_2(t) - \min(C_2(t))}{\max(C_2(t)) - \min(C_2(t))}\right). \tag{4.73}$$

The NMPC objective function given equation (4.64) uses the expressions given in equations (4.65) (4.66) (4.70) (4.72), and equation (4.73). The NMPC objective function is solved along with state and control constraints.

4.6 Results and Discussion

Extensive numerical simulations were carried out to validate the proposed scheme using CasADi(3.5.5)-Python(3.8) [137]. Optimizations for the NMPC and the MHE were carried out using the interior-point optimizer IPOPT similar to the previous chapters. We use the total path length and average estimation error as metrics for analyzing the performance of the proposed approach. We perform the following analysis (i) the effect of horizon length in the NMPC on the path length and estimation error, (ii) the effect of cooperation, (iii) comparison with the approach proposed in [90] and (iv) effect of increasing the number of vehicles in the region to 10. Before presenting the analysis, we will describe the simulation setting.

4.6.1 Simulation setup

We consider an environment of 200 m $\times$ 200 m, where 20 landmarks are randomly placed. Each vehicle starts at a given location and has a desired goal location. The vehicles have a constant velocity of 5 m/s. The angular velocities of the agents are constrained by $[-\pi/2, \pi/2]$ rad/s due to the practical considerations on the turn rate of the agents. The value of η is selected as 2, and the values of κ and ρ are selected as 5 and 0.5, respectively. The weight W is selected as 10000, sensor range of the vehicles, R_s = 50 m, and σ_c = 3 m. The measurement noise covariance matrix Γ is selected as a $n_\Gamma \times n_\Gamma$ matrix with 0.01 in its diagonals, where n_Γ is the number of received measurements. Each time step is 0.1s for all the simulations. The simulations were carried out on an Ubuntu 18.04, Intel i9 workstation with 64GB RAM.

4.6.2 Effect of NMPC horizon length

In NMPC, the horizon length plays a key role between path optimality and computational time. The larger the horizon, the better the path obtained at the cost of increased computational time. This effect can be seen in Fig. 4.4a, where the computation time for the prediction horizon of $\tau_h = 1$ s is 0.056 s; however, the paths are not optimal. With the increase in τ_h to 15 s, there is a significant improvement in the path of vehicle 1 at an increased computational time of 0.42 s per iteration. With further increase in horizon length to $\tau_h = 40$ s, the average time to compute an iteration is 3.43 s, but the obtained path length for the vehicles is near-optimal. The path length for $\tau_h = 25$ s is close to that obtained with $\tau_h = 40$ but takes only 1.2 s.

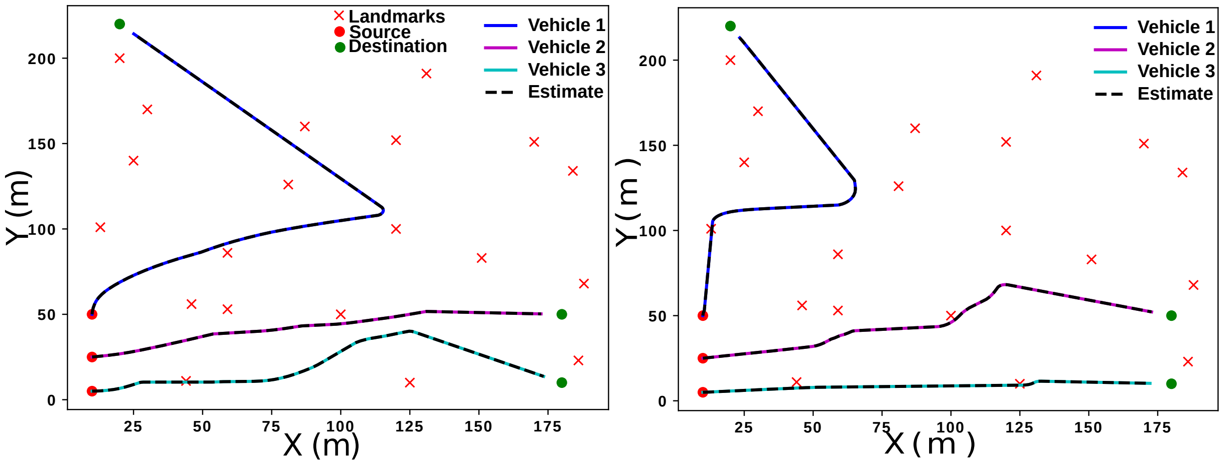
Further, we conducted Monte-Carlo simulations to see the effect of τ_h on a larger number of simulations for a given number of landmarks (20). For each value of τ_h, 25 simulations were performed by randomly placing the landmarks. We will use the maximum time taken ($T_{\max}$) by a vehicle to reach the given destination as the mission time for that simulation. From Fig. 4.5(a), we can see that the computation time for $\tau_h = 1$ s is very less, but the average $T_{\max}$ is high as shown in 4.5(b). Note that, on average, $T_{\max}$ for $\tau_h = 25$ s and $\tau_h = 40$ s are almost similar, however, $\tau_h = 25$ takes far less time. Hence we consider $\tau_h = 25$ s for the rest of the simulations.

4.6.3 Effect of number of landmarks (n_l) and sensing range R_s

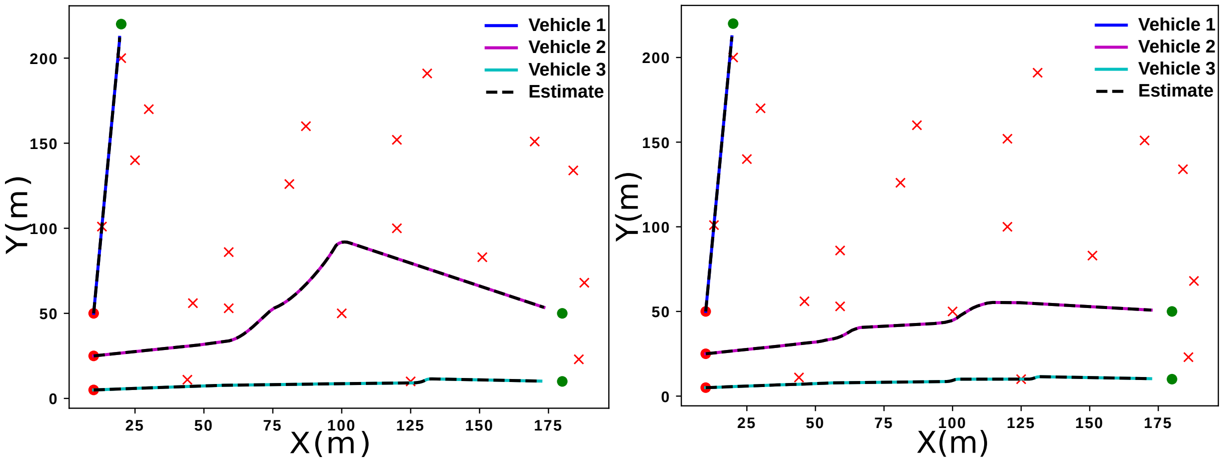
The number of landmarks and the sensing range of the vehicles play a role in the ability of the proposed approach to determine the path. Considering $\tau_h = 25$, we conducted simulations with various numbers of landmarks (5,10,15, and 20) and varying the sensing range from 10 m to 50 m. For each combination of landmark and sensing range, 25 simulations were performed by randomly placing the landmarks. In total, 500 simulations were carried out. Figure 4.5(c) shows the percentage of successful mission bar plots. Each simulation was given 100 seconds, and if the simulation was completed within this time, the mission was assumed to be accomplished. Each stack of bars corresponds to a given landmark for varying R_s. When the number of landmarks is less and the sensing range is less, then finding paths is difficult. This aspect can be seen in Fig. 4.5(c) for $n_l = 5$. As R_s increases, the mission success improves. This is natural as the vehicle can detect landmarks due to increased sensing range. Similarly, as we increase the number of landmarks for a given R_s, we can see improvement in the mission success rate. For $R_s = 10$ (blue bars), we can see that the success rate improves from 8% to 76% as the number of landmarks increases from 5 to 20. With a higher sensing range, we can always achieve complete success, as shown for $R_s = 50$ (green bars) in the figure for any number of landmarks. This study shows that we need to select n_l and R_s appropriately for an improved mission success rate.

4.6.4 Effect of cooperation

One of the main contributions of this work is to show that with cooperation, the vehicles can jointly determine minimal distance paths to their goal locations while meeting localization accuracy. To show the effect of cooperation, we consider a specific scenario as shown in Fig. 4.6, where the landmarks are located at the top of the scenario. Two simulations were carried out with five vehicles, in which one scenario involved cooperative vehicles and the other without cooperation. We consider τ_h = 25 s, and $R_s = 30$ m. The trajectories taken by the vehicles in both cases are shown in Fig. 4.6. It can be seen from Fig. 4.6(a) that the vehicles with cooperation determined a shorter path since they used adjacent vehicles for localization, whereas the non-cooperative vehicles took a longer path to go near the landmarks for localization, as shown in Fig. 4.6(b). The estimation errors are a bit high for the cooperative case due to the mutual localization of the vehicles.



(a) $\tau_h = 1\,\mathrm{s}$ (b) $\tau_h = 15\,\mathrm{s}$



(c) $\tau_h = 25\,\mathrm{s}$ (d) $\tau_h = 40\,\mathrm{s}$

Figure 4.4: The average computational time taken per iteration for different τ_h. (a) 0.05 s (b) 0.42 s (c) 1.21 s (d) 3.43 s.

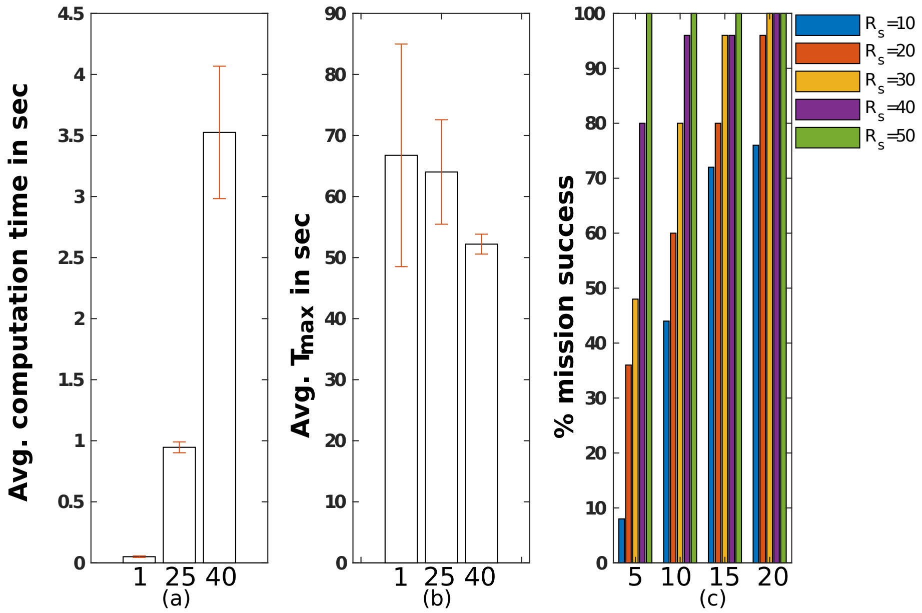


Figure 4.5: Monte-Carlo simulation for (a) Average computational time per iteration for $\tau_h = 1, 25, 40$ (b) Average time taken by all the agents to reach their destinations for $\tau_h = 1, 25, 40$ (c) Percentage mission completed for the varying number of landmarks $n_l = 5, 10, 15, 20$ and sensing ranges $R_s = 10, 20, 30, 40, 50$.

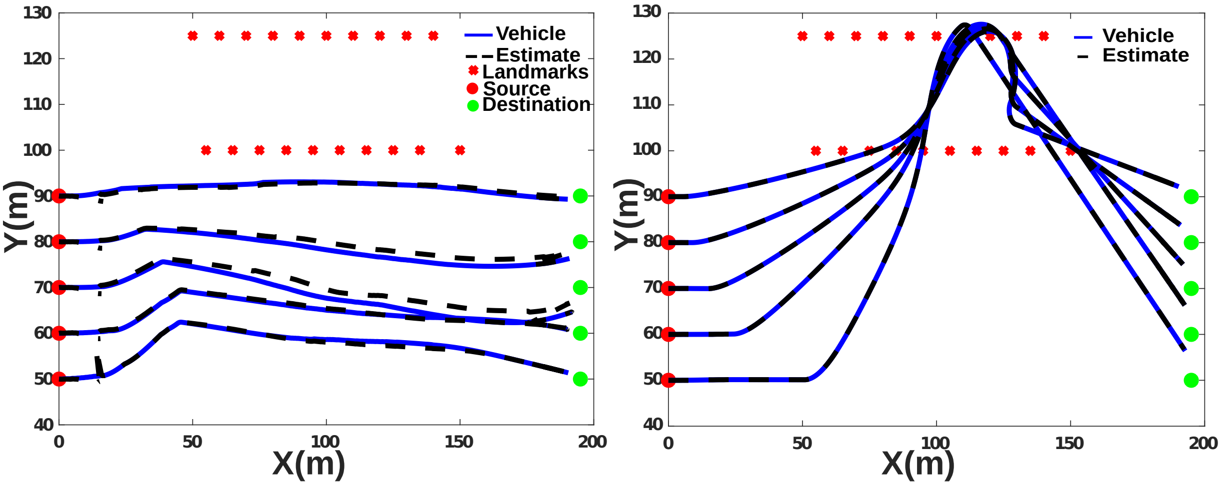


Figure 4.6: Effect of cooperation. (a) The trajectory of vehicles with cooperation. (b) The trajectory of vehicles without cooperation.

4.6.5 MHE vs EKF

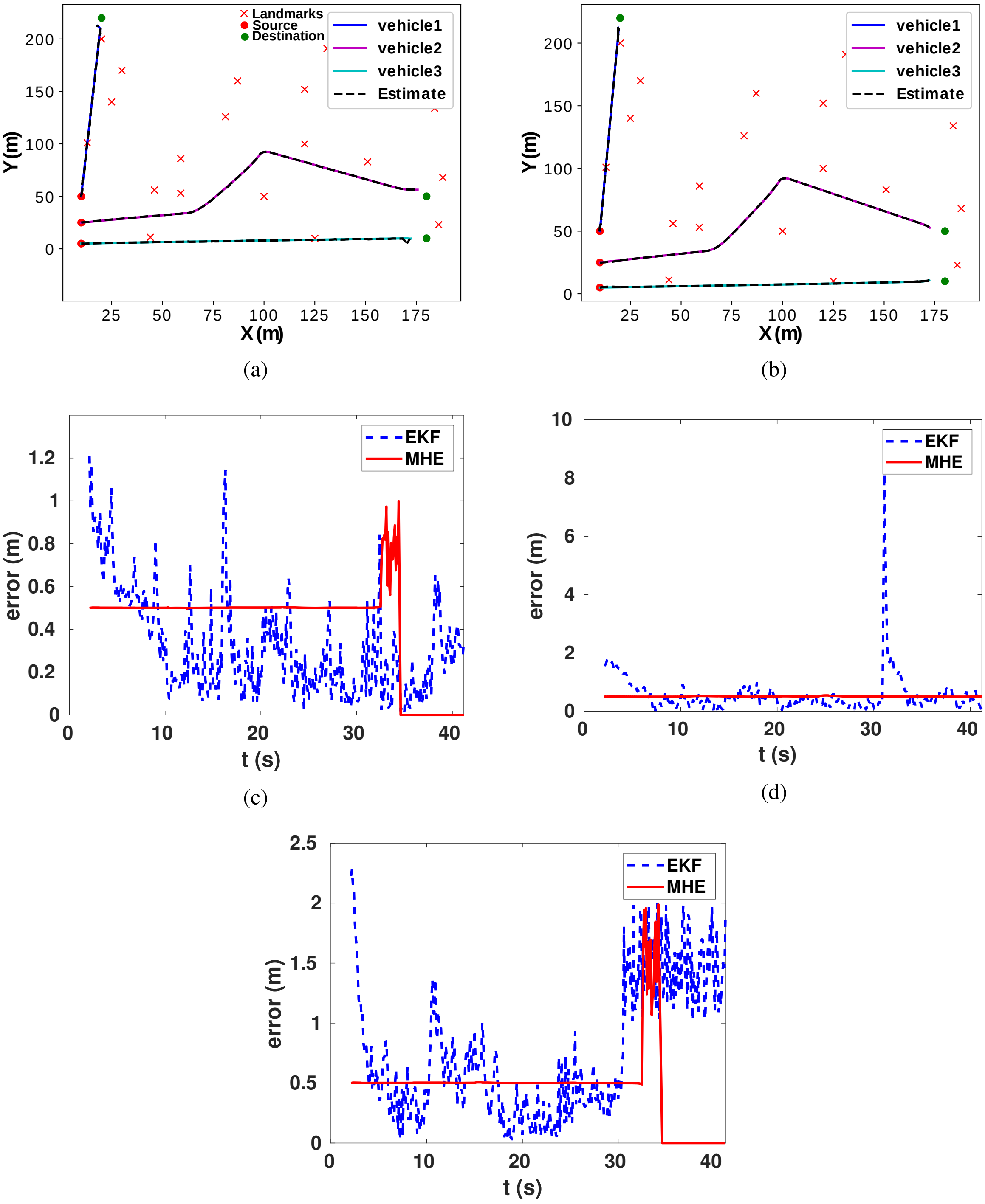
The performance of the MHE estimator was compared against a standard EKF formulation to validate the superiority of the proposed scheme. In Fig. 4.2, the step 1 sub-block of the MHE is replaced by the EKF. The horizon length for the MHE is selected as $N_e = 20$, and all other initial parameters are taken the same for the MHE and the EKF. The estimated trajectories of the vehicles, true trajectories, and the errors in position for both the MHE and the EKF are given in Fig. 4.7. It can be seen that the estimation errors are less for the MHE, and it is also more stable compared to the EKF. The vehicles' mean square error (MSE) for the MHE is 0.46 m, and that with the EKF is 0.79 m. The computation time required for the MHE is 1.21 s, while the EKF takes 1.10 s. Although the MHE takes 9% more computational time than the EKF, its accuracy is 72% better than the EKF.

4.6.6 10 vehicles

To test the performance of the scheme for larger systems, a scenario involving ten vehicles in a 500 m × 500 m plane with all the vehicles moving at a velocity of 10 m/s was considered. Multiple simulations were carried out with random configurations of the landmarks, initial positions, and goal points. Two example results are shown in Fig. 4.8. It can be seen that the estimator performance was satisfactory since the actual states and the estimated states are well aligned for all ten vehicles. The average computation time per iteration was 7.25 s due to the increase in the size of the state matrix for 10 vehicles.

4.7 Conclusions

A nonlinear model predictive control scheme combined with moving horizon estimation was proposed to aid the cooperative localization of a group of AAVs in transit. The controller used an approximate analytical expression for calculating the expected covariance of the vehicles through the prediction horizon, which was derived using the insights obtained from analyzing the observability and the path information from the landmark-vehicle graph. The controller determined near-optimal paths for the vehicles while satisfying various state and localization constraints. We analyzed through simulations the role of prediction horizon on the optimality of the vehicle paths and the required computation time. The proposed moving horizon estimator also outperformed the EKF with lower estimation error values at a small additional computation time. A comparison was performed between cooperative and non-cooperative vehicles to show the significance of cooperation in determining paths under localization constraints.



(a)

(b)

(c)

(d)

(e)

Figure 4.7: Comparison of MHE with EKF (a) NMPC solution with EKF based estimation (b) NMPC solution with MHE (c) Error in the position of vehicle 1 (d) Error in the position of vehicle 2 (e) Error in the position of vehicle 3

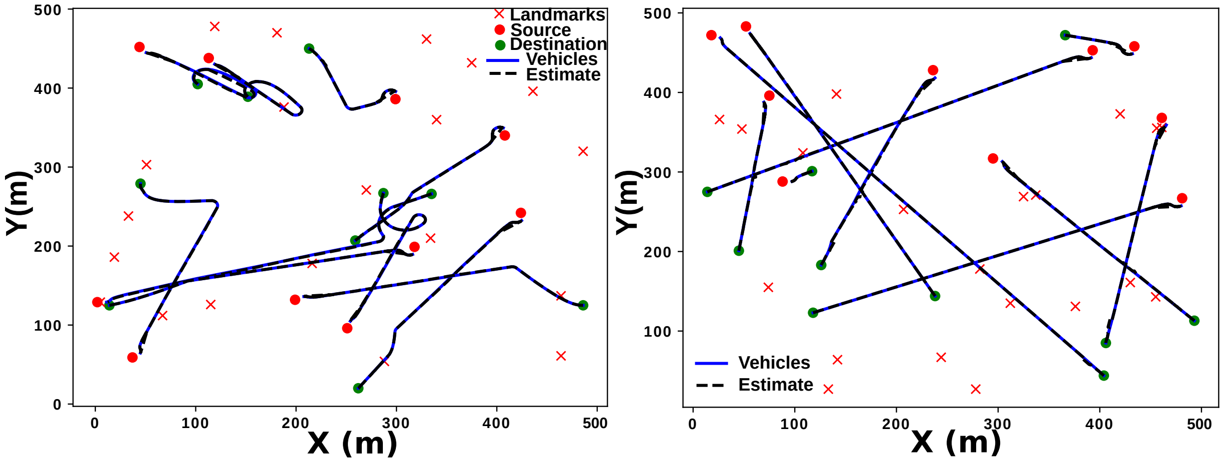


Figure 4.8: Trajectories of 10 vehicles using NMPC-MHE scheme.

Chapter 5

Learning-based NMPC Framework for Car Racing Cinematography Using AAVs

5.1 Introduction

Here, we are interested in the repetitive task of filming autonomous robot race cars with the use of gimbaled fixed-wing and quadrotor autonomous aerial vehicles (AAVs), as illustrated in Fig. 5.1. Competitions involving robots are becoming increasingly familiar with hobbyists, students, and researchers for demonstrating cutting-edge technology or entertainment. The races are conducted in indoor and outdoor settings with varying degrees of difficulty. We are interested in the outdoor races where the tracks are situated in large metropolitan cities. This kind of environment makes the aerial videography of the race very difficult. The cityscape cluttered with buildings and other infrastructures create obstructions for the free movement of the AAVs and, in some cases, occludes the camera field of view (FOV). This problem can be modeled as a target-tracking problem with obstacle avoidance, where the objective is to minimize the tracking errors while respecting the imposed physical constraints. Since the races usually have multiple numbers of cars, it is not possible to tightly follow and film all the cars using a single AAV since the field of view of the camera has limitations. Also, the cars might not move side by side; rather, one car might go ahead, and the other might fall behind. In situations like these, multiple AAVs are required. Also, using a quadrotor in the mix helps in better tracking and collision avoidance since quadrotors have the ability to hover in the air.

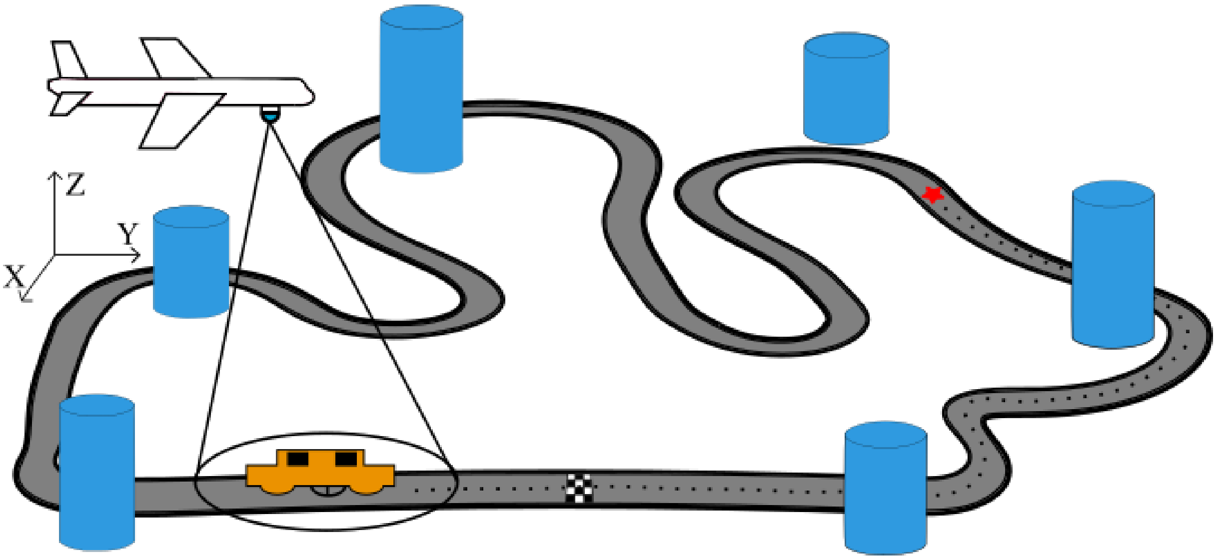


Figure 5.1: Abstract representation of a race track showing a car filmed by an AAV.

5.2 Learning-Based NMPC

We assume that the cars and the AAVs start at the same initial conditions for all iterations (laps of the race), and the cars follow the same trajectory for all iterations. The control, state, and gimbal parameters are constrained to mimic a realistic scenario. We propose the use of NMPC for control since it can easily handle state and input constraints. Tuning the weights for a multi-objective cost function is a tedious task. Hence, the RL technique is used to learn the weights of the components of the cost function. The combined RL-NMPC scheme, which we refer to as learning-based NMPC (L-NMPC) from here onward, results in precise tracking of the cars due to the perfect balance of the weights in the objective function.

A block diagram showing the proposed L-NMPC scheme is given in Fig. 5.2. The state parameters of the AAVs, gimbals, and cars are given to the NMPC block as feedback. Computed control commands from the NMPC block are applied to the AAV-gimbal system. The weights required in the NMPC cost function are calculated by the RL block.

5.2.1 System kinematics

We use a discrete-time model to represent the motion of the gimbaled AAVs. The state vector of a single AAV is represented by $X_A \in \mathbb{R}^5$ and the evolution of the states over the time step j is written as

$$X_{A,j+1} = X_{A,j} + f(X_{A,j}, U_{A,j})T_s, \tag{5.1}$$

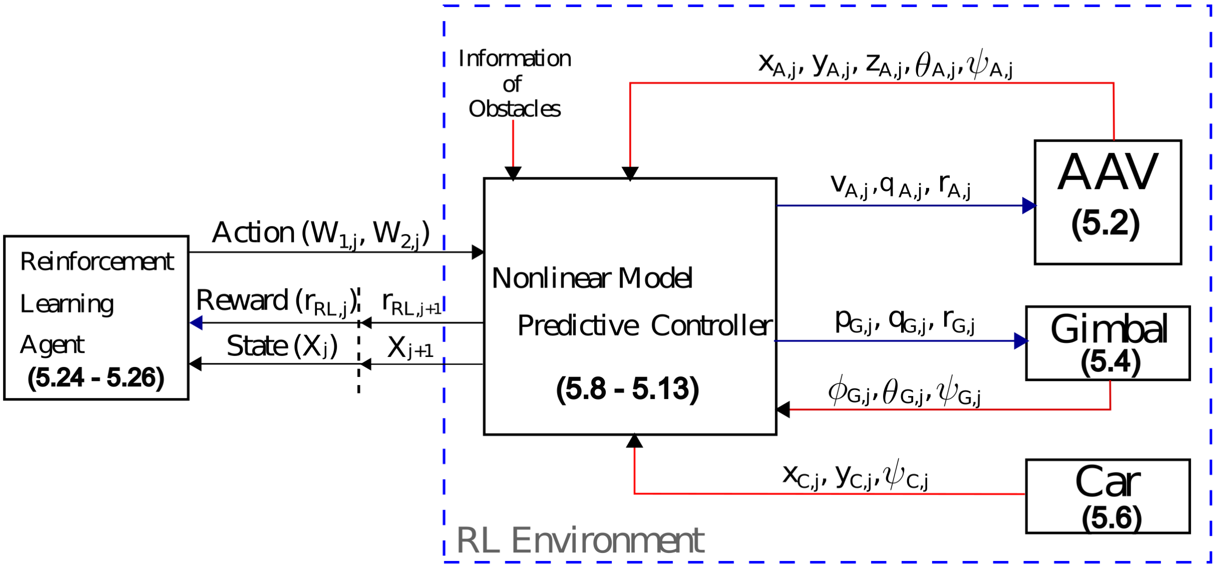


Figure 5.2: Block diagram of the proposed RL-based Learning-NMPC framework.

where T_s is the sampling time, $X_{A,j} = [x_{A_j}, y_{A_j}, z_{A_j}, \theta_{A_j}, \psi_{A_j}]^T$ and $U_{A,j} = [v_{A_j}, q_{A_j}, r_{A_j}]^T$ are the state and control vector of the AAV. Omitting the time subscript j for readability, the function $f(X_A, U_A)$ is defined as:

$$f(X_A, U_A) = \begin{bmatrix} v_A \cos\psi_A \cos\theta_A \\ v_A \sin\psi_A \cos\theta_A \\ v_A \sin\theta_A \\ q_A \\ r_A \end{bmatrix}, \tag{5.2}$$

where (x_A, y_A, z_A) is the location coordinates of the AAV in Cartesian frame, θ_A and ψ_A are pitch and heading (yaw) angles, respectively. v_A represents the linear velocity of the AAV, and q_A and r_A represent the pitch and yaw rates, respectively.

A 3-DoF gimbal is attached to the AAV with the assumption that the center of the camera co-inside with the center of gravity of the AAV. The kinematics of the gimbal is defined as

$$X_{G,j+1} = X_{G,j} + f(X_{G,j}, U_{G,j})T_s, \tag{5.3}$$

where $X_G = [\phi_G, \theta_G, \psi_G]^T$ and $U_G = [p_G, q_G, r_G]^T$. The function $f(X_G, U_G)$ is defined as:

$$f(X_G, U_G) = \begin{bmatrix} p_G \\ q_G \\ r_G \end{bmatrix}, \tag{5.4}$$

where ϕ_G, θ_G, ψ_G are the roll, pitch, and yaw angles of the gimbal, and p_G, q_G, r_G are the corresponding angular rates.

We assume that the race track is flat and the car moves only in a 2D plane. We also assume that the linear velocity v_C, angular velocity r_C, and heading angle ψ_C along with the location coordinates of the cars (x_C, y_C) are available to the AAVs. A standard Kalman filter can be used if the car states are not directly available. The kinematic model of the car is given as

$$X_{C,j+1} = X_{C,j} + f(X_{C,j}, U_{C,j})T_s, \tag{5.5}$$

where $X_C = [x_C, y_C, \psi_C]^T$ and $U_C = [v_C, r_C]^T$. The function $f(X_C, U_C)$ is given by

$$f(X_C, U_C) = \begin{bmatrix} v_C \cos\psi_C \\ v_C \sin\psi_C \\ r_C \end{bmatrix}. \tag{5.6}$$

The states of AAV, gimbal, and car are combined to formulate a joint state vector given by

$$X = \begin{bmatrix} X_A \\ X_G \\ X_C \end{bmatrix}. \tag{5.7}$$

5.2.2 Cost function

NMPC is used to determine the optimal control inputs for the AAVs and the gimbals such that the AAVs can film the cars without physically colliding with the obstacles and other AAVs and always keep the cars within the camera FOVs.

The cost function is defined as:

$$\begin{aligned} &\min_{U_j} J(X_j, U_j), \\ &J(X_j, U_j) = \sum_{k=j}^{j+N} \sum_{i=1}^{n} W_{1,i,k} f_1(X_{i,k}) + W_{2,i,k} f_2(X_{i,k}), \end{aligned} \tag{5.8}$$

subject to:

$$X_{i,j+1} = f(X_{i,j}, U_{i,j}), \tag{5.9}$$

$$X_{i,j} \in [X_i^-, X_i^+], \tag{5.10}$$

$$U_{i,j} \in [U_i^-, U_i^+], \tag{5.11}$$

$$-\big[D_{o,i,k} - (R_{A_i} + R_{o,k})\big] \leq 0, \;\; k = 1, 2 \ldots, n_o, \tag{5.12}$$

$$-\big[D_{A,i,k} - (R_{A_i} + R_{A,k})\big] \leq 0, \;\; i, k = 1, \ldots, n, i \neq k, \tag{5.13}$$

where n is the number of AAVs and cars, $U_j = [v_{A,j}, p_{A,j}, r_{A,j}, p_{G,j}, q_{G,j}, r_{G,j}]^T$ is the combined control vector of the AAV and the gimbal, W_1 and W_2 are the weighting coefficients, X^-, X^+ are the lower and upper bounds on the states, and U^-, U^+ are the lower and upper bounds of the control inputs, respectively. The constraints (5.12),(5.13) are added to incorporate the obstacle and inter-collision avoidance capability for the AAVs. The parameter $D_{A,i,k}$ is the distance between the AAVs and $D_{o,i,k}$ is the Euclidean distance between the center of the projection of the i^{th} AAV on the XY-plane and the center of projection of the k^{th} obstacle on the XY-plane given by

$$D_{o,i,k} = \sqrt{(x_{A_i} - x_{o,k})^2 + (y_{A_i} - y_{o,k})^2}, \;\; k = 1, 2, \ldots, n_o, i = 1, \ldots, n, \tag{5.14}$$

where n_o is the number of obstacles present, R_A is the radius of the AAV, $R_{o,k}$ is the radius of the k^{th} obstacle, and $x_{o,k}, y_{o,k}$ are the coordinates of the center of the k^{th} obstacle on the XY-plane as shown in Fig. 5.3.

The components of the cost function, $f_1(X)$ and $f_2(X)$, are defined as follows (AAV index i is omitted for readability).

$f_1(X)$: for accurately tracking the car, it is desired that the horizontal projection of the AAV stays aligned with the x, y coordinates of the car. Hence, f_1 is defined as the distance between the AAV and the car in the XY plane:

$$f_1(X) = \sqrt{(x_A - x_C)^2 + (y_A - y_C)^2}, \tag{5.15}$$

It might not be possible to make this function zero for all instances of time because of the difference in the speed of the AAV and the car. Since the fixed-wing AAVs do not have hover capability, they will start circling on top of the car to minimize the distance in the XY plane.

$f_2(X)$: this cost component makes sure that the car is within the FOV (Field Of View) of the camera,

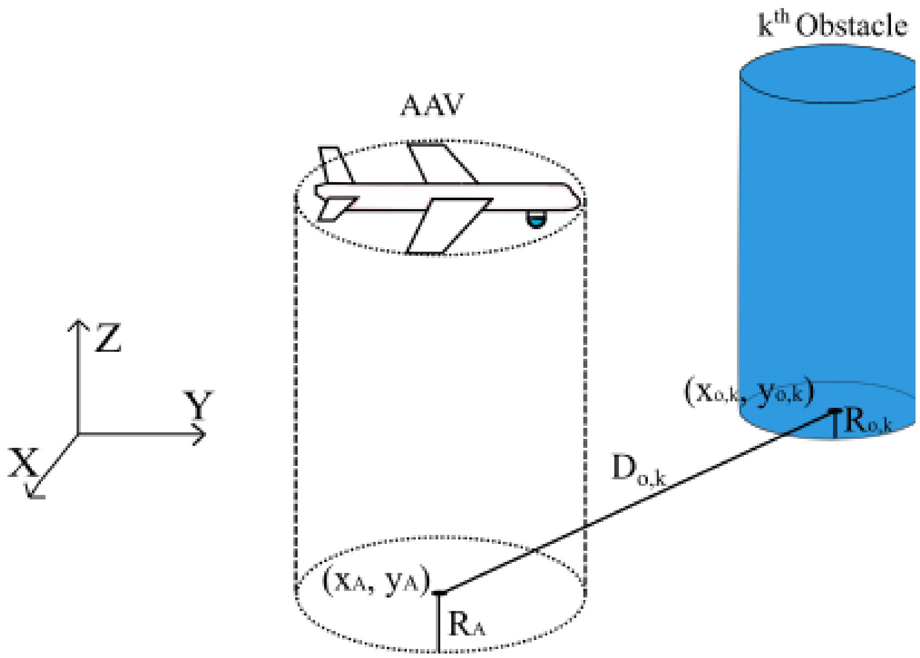


Figure 5.3: Obstacle avoidance of the AAV.

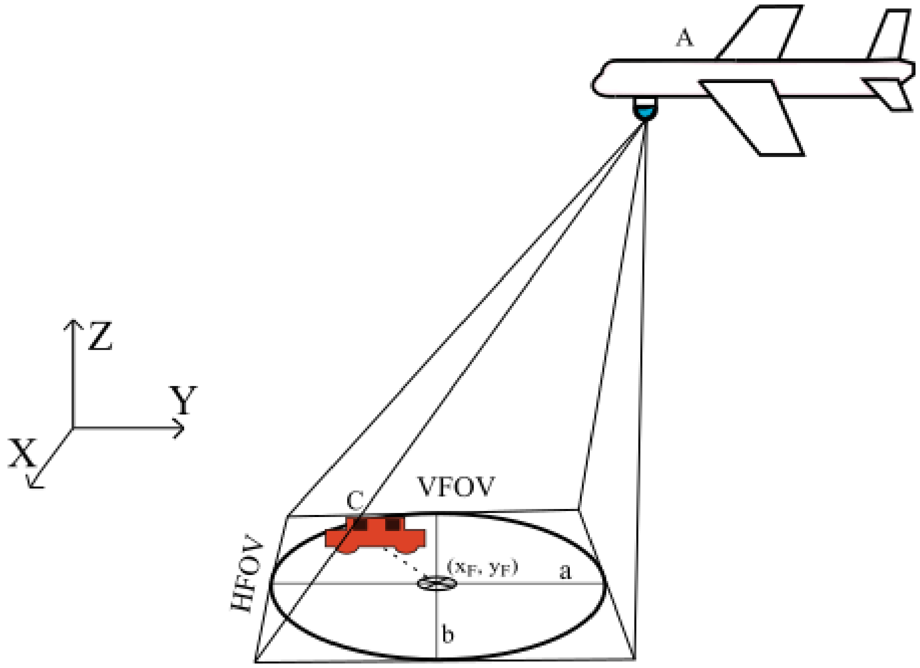


Figure 5.4: Illustration showing the FOV ellipse of the gimbaled camera, the race car and the AAV.

which is modeled as an ellipse.

$$f_2(X) = \Lambda_1(x_C - x_F)^2 + \Lambda_2(x_C - x_F)(y_C - y_F) + \Lambda_3(y_C - y_F)^2 - 1, \tag{5.16}$$

where (x_F, y_F) is the center of the FOV calculated as follows:

$$x_F = a + x_A + z_A \tan(\theta_G - VFOV/2), \tag{5.17}$$

$$y_F = b + y_A + z_A \tan(\phi_G - HFOV/2), \tag{5.18}$$

where a and b are the major and minor axis of the FOV ellipse, defined as

$$a = \frac{1}{2}\Big(z_A \tan\left(\theta_G + VFOV/2\right) - z_A \tan\left(\theta_G - VFOV/2\right)\Big), \tag{5.19}$$

$$b = \frac{1}{2}\Big(z_A \tan\left(\phi_G + HFOV/2\right) - z_A \tan\left(\phi_G - HFOV/2\right)\Big). \tag{5.20}$$

The parameters $VFOV$ and $HFOV$ are the vertical and horizontal field of view of the camera, which is available in the camera specifications. The coefficients Λ_1, Λ_2, and Λ_3 are defined as:

$$\Lambda_1 = \frac{\cos^2\psi_G}{a^2} + \frac{\sin^2\psi_G}{b^2}, \tag{5.21}$$

$$\Lambda_2 = 2\cos\psi_G \sin\psi_G \left(\frac{1}{a^2} - \frac{1}{b^2}\right), \tag{5.22}$$

$$\Lambda_3 = \frac{\sin^2\psi_G}{a^2} + \frac{\cos^2\psi_G}{b^2}. \tag{5.23}$$

The cost component (5.16) gives a negative value when the car is within the FOV and a positive value if the car is outside of the FOV. The aforementioned parameters are illustrated in Fig. 5.4.

5.3 Reinforcement Learning

RL is a nature-inspired, experience-based learning method with three main components: state, action, and reward. The RL agent makes a selection among the possible actions, observes the state and reward obtained for the corresponding action, and makes future actions to maximize the reward [147]. The selection of actions has two modes, exploration, and exploitation. In exploration, the agent searches for actions that will give a higher reward, and in exploitation, the same action is used to increase the sum of rewards. Striking a balance between these two modes is the key to increasing the performance of the agent.

In this work, we used the constant alpha Q-learning algorithm for learning the weights for each iteration (lap) of the race. The Q-update rule for the constant alpha is given by [147] as:

$$Q^{new}(s_j, a_{RL,j}) = Q(s_j, a_{RL,j}) + \alpha_{RL}\Bigg(r_{RL,j} + \gamma_{RL} \max_{a_{RL}} Q(s_{j+1}, a_{RL}) - Q(s_j, a_{RL,j})\Bigg), \quad (5.24)$$

where Q^{new} is the updated Q-value of the state-action pair $(s_j, a_{RL,j})$ while α_{RL} and γ_{RL} are the learning rate and the discount factor, respectively. $Q(s_j, a_{RL,j})$ is the old Q-value, and $\max_{a_{RL}} Q(s_{j+1}, a_{RL})$ represents the estimate of the optimal future value.

We consider each lap of the race as an RL episode. The state space s is the same as the NMPC state vector X. The action space a_{RL} is defined as the NMPC weight set $(W_{1,1}, W_{1,2}, \ldots, W_{n,1}, W_{n,2})$ where $W_1, W_2 \in [1, 100]$. The reward is defined as the reciprocal of the tracking errors given by

$$r_{RL} = \frac{1}{\text{error}} = \frac{1}{\sum_{i=1}^{n} \sqrt{(x_{F_i} - x_{C_i})^2 + (y_{F_i} - y_{C_i})^2}}, \quad (5.25)$$

The reward function gives a higher reward when the distance between the cars and the center of the FOVs is less.

Action selection is governed by the ε-greedy method where epsilon decay is defined as exponential decay formula given as:

$$\varepsilon = \varepsilon_{min} + (\varepsilon_{max} - \varepsilon_{min})e^{-\lambda episode}, \quad (5.26)$$

where λ is the epsilon decay rate while ε_{min} and ε_{max} are the maximum and minimum values of ε.

5.4 Results and Discussion

The performance of L-NMPC for filming the race-cars was evaluated through numerous simulations with race tracks of different configurations, and the selected results are analyzed in this section. The NMPC algorithm was implemented using the CasADi(3.5.5)-python(3.8) [148] framework, and a custom

AAV constraints			Gimbal constraints		
variable	min	max	variable	min	max
$z_A(m)$	75	150	$\phi_G(rad)$	$-\pi/6$	$\pi/6$
$\theta_A(rad)$	-0.2618	0.2618	$\theta_G(rad)$	$-\pi/6$	$\pi/6$
$v_A(m/s)$	14	30	$\psi_G(rad)$	$-\pi/2$	$\pi/2$
$p_A(rad/s)$	$-\pi/30$	$\pi/30$	$p_G(rad/s)$	$-\pi/30$	$\pi/30$
$r_A(rad/s)$	$-\pi/21$	$\pi/21$	$q_G(rad/s)$	$-\pi/30$	$\pi/30$
			$r_G(rad/s)$	$-\pi/30$	$\pi/30$

Table 5.1: Constraints on the AAV-gimbal system.

reinforcement learning environment was created using the OpenAI Gym [149] library. First, we present simulation results for a single AAV-race car pair and then move on to a multi-vehicle simulation.

5.4.1 Simulation setup

The sampling time for the NMPC is selected as T_s = 0.2 s, and the length of the prediction horizon N = 15 for all simulations. The cars move with a constant speed of v_C = 12 m/s, and the track is surrounded by ten cylindrical obstacles with a radius of 50 m and a height of 120 m. In order to balance the exploration and exploitation, ε-greedy method with ε_{min} = 0.03, ε_{max} = 1.0, and ε-decay rate λ = 0.008 were used. Learning rate was selected as α = 0.95 and discount factor γ = 0.85. The RL parameters were selected in a way that the agent can learn with a low number of episodes, where each episode has 2000 NMPC steps. The simulations consisted of 500 episodes of training for the single vehicle simulations and 100 episodes for multi-vehicle simulations, which were carried out on an Intel i7 workstation with 16 GB of RAM.

5.4.2 Single AAV-race car pair

The initial positions of the car and the AAV are selected as (x_C, y_C) = (100,150) and (x_A, y_A, z_A) = (99,150,80), respectively. The bounds on the states and controls are given in Table 5.1. At the beginning of the race, the RL agent explores some random actions (weight set), and they are passed to the NMPC cost function. The NMPC finds the optimal control inputs for the AAV-gimbal system with the received weight set and the error values. Here, the NMPC framework is considered as the RL environment, which gives the reward to the RL agent. The agent updates the Q-table with the weights for the corresponding NMPC step, and the iterative process is continued to get the best actions.

Consider Fig. 5.5. It shows the optimal trajectory of the AAV obtained using the proposed L-NMPC

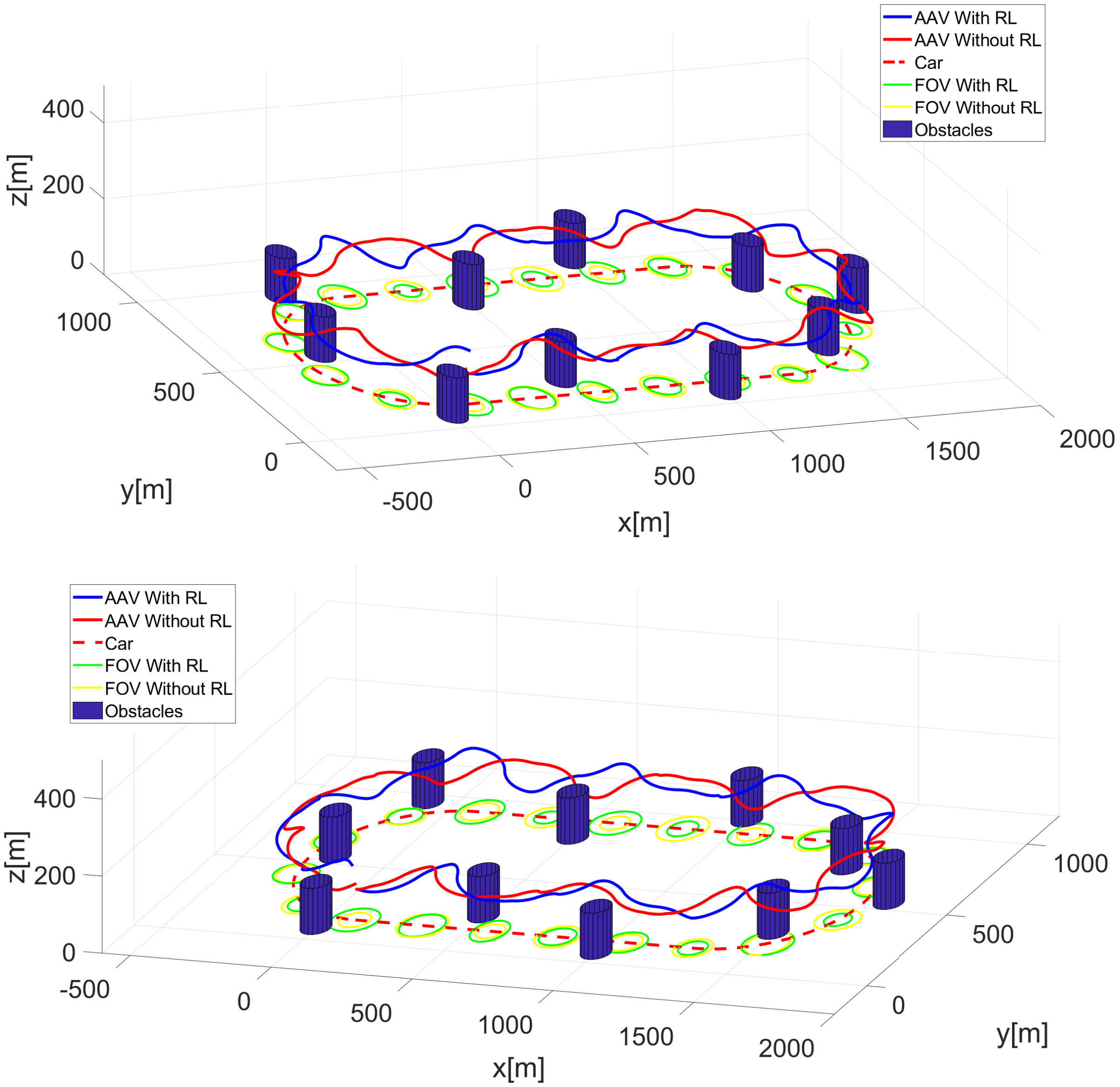
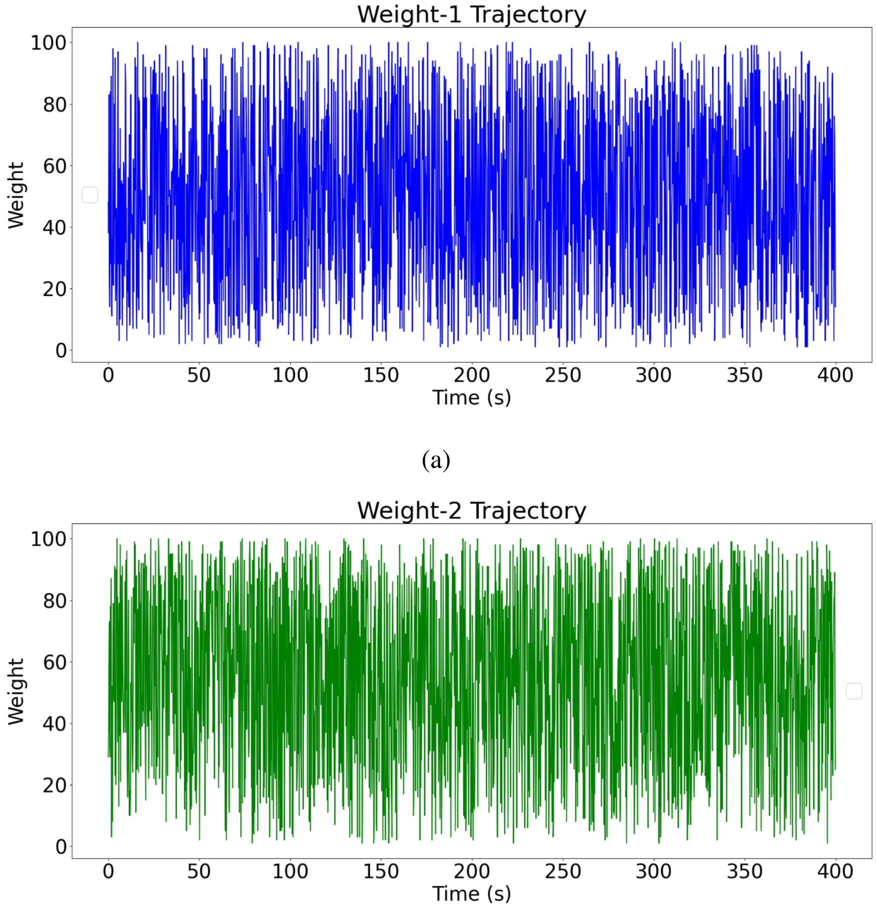


Figure 5.5: Different views of the AAV tracking the car. The trajectories of the RL-trained agent (blue) and the untrained agent (red) are shown. The FOVs of the agents at intermittent time steps are also given.



(a)

(b)

Figure 5.6: Weights for the NMPC cost function obtained through the learning process: (a) W_1 which regulates the term $f_1(X)$ defined in (5.15), (b) W_2 which regulates the term $f_2(X)$ defined in (5.16).

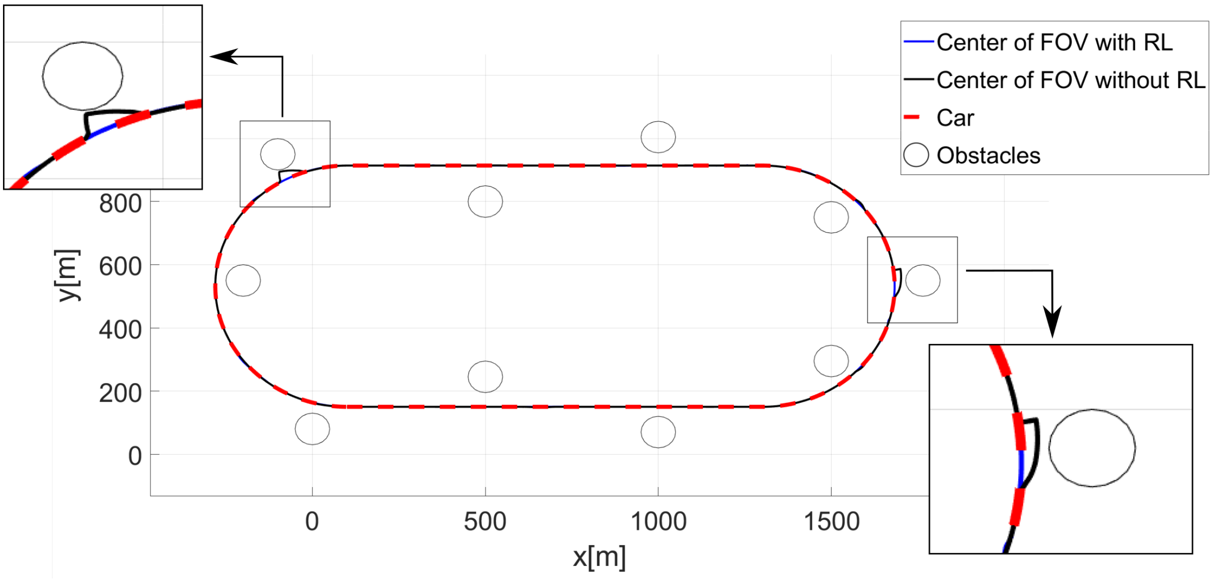


Figure 5.7: Top view of the track showing the trajectories of the center of FOV of RL-trained and untrained agents and the car trajectory.

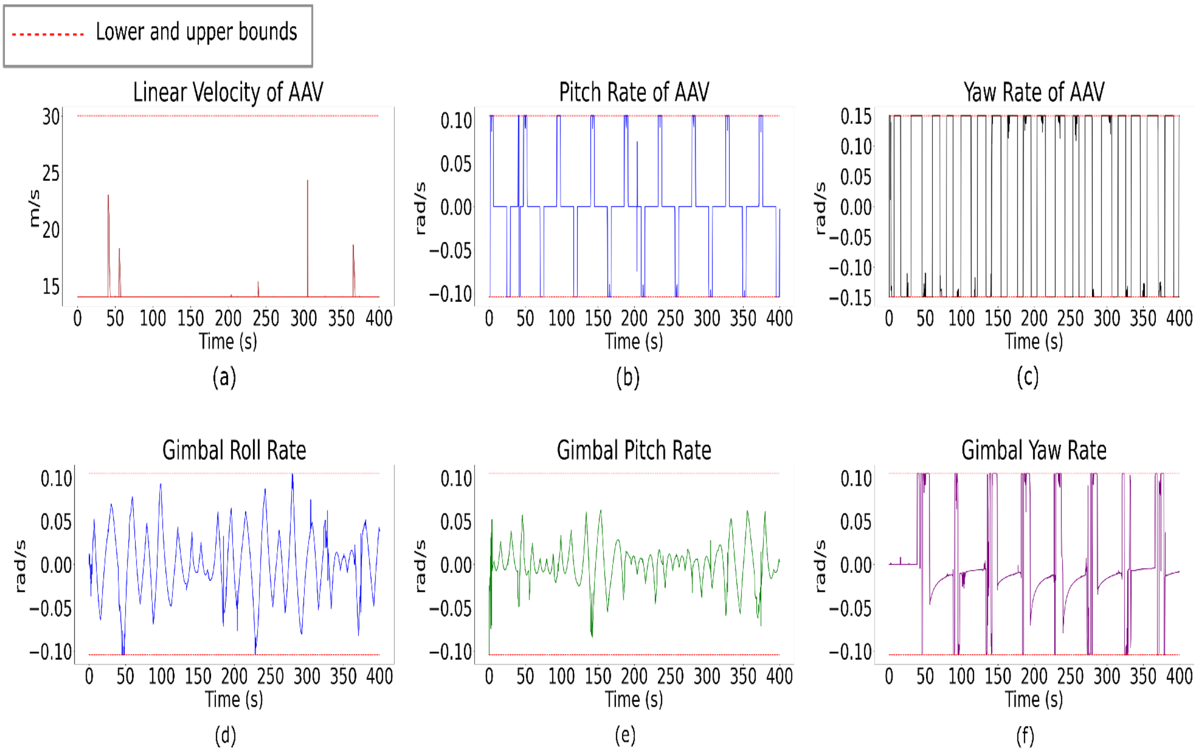


Figure 5.8: Control actions of L-NMPC: (a) Linear velocity of AAV, (b) Pitch rate of AAV, (c) Yaw rate of AAV, (d) Roll rate of the gimbal, (e) Pitch rate of the gimbal, (f) Yaw rate of the gimbal.

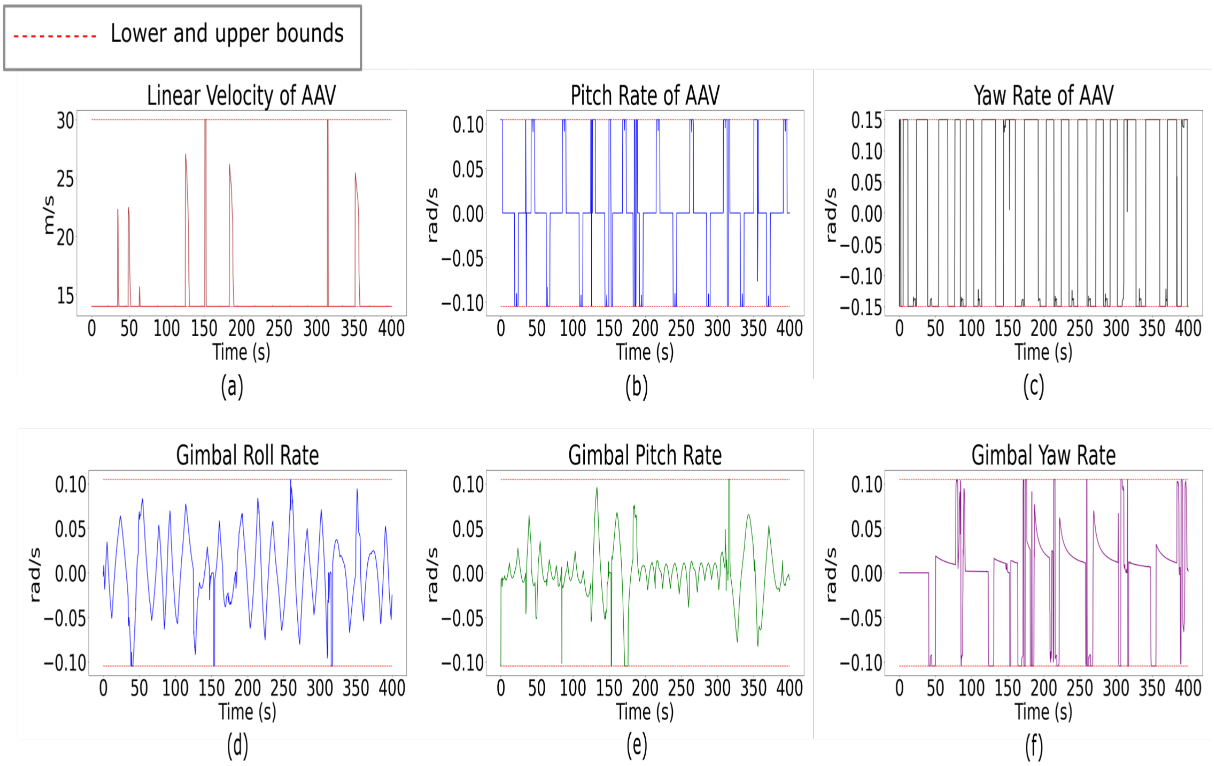


Figure 5.9: Control actions of untrained NMPC: (a) Linear velocity of AAV, (b) Pitch rate of AAV, (c) Yaw rate of AAV, (d) Roll rate of the gimbal, (e) Pitch rate of the gimbal, (f) Yaw rate of the gimbal.

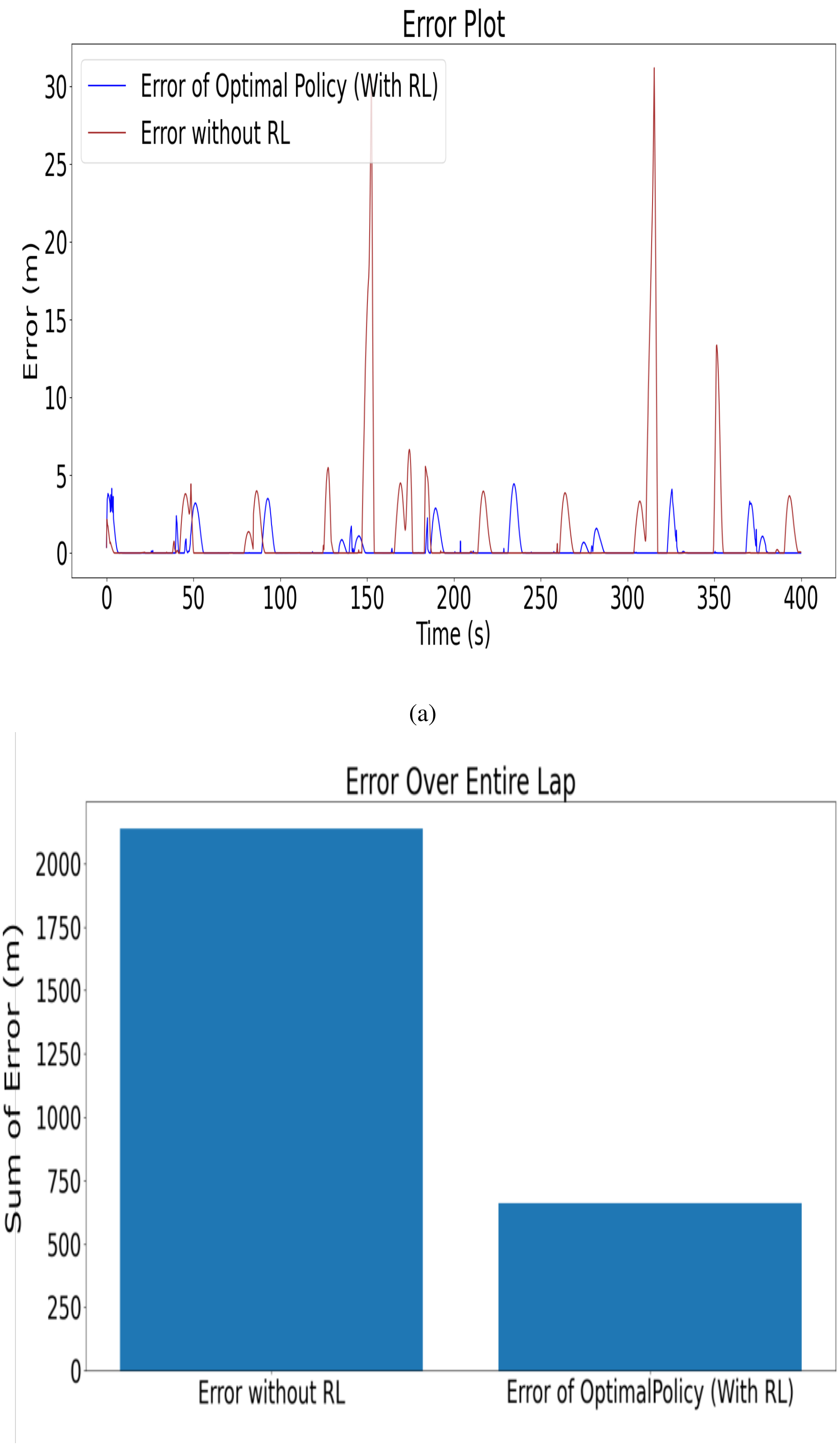
scheme. It is also compared with an agent using a standard NMPC scheme with constant pre-defined weights ($W1 = 1, W2 = 1$). It can be seen that the RL-trained AAV was able to track the car while avoiding collision with the obstacles. The performance of the trained AAV is better than the untrained AAV, as shown in Fig. 5.7. The error between the trajectory of the car and the center of FOV of the AAV is lower for the RL agent. The untrained agent intermittently lost track of the car while the proposed scheme delivered precise tracking. The center of FOV of the untrained agent is moving closer to the obstacles in some places since the AAV deviated far away on the outside to avoid the obstacles. The weights obtained from the optimal RL episode are shown in Fig. 5.6. It is to be noted that the rapid switching of weights does not put any strain on the physical system since they are not actuating signals.

The control inputs for the AAVs and the gimbals computed by the L-NMPC and the standard NMPC are shown in Figures 5.8 and 5.9 respectively. The number of spikes in the linear velocity is lower for the RL-trained AAV compared to the untrained one, as shown in Fig. 5.8(a) and Fig. 5.9(a), respectively. This helps the trained AAV to film the car with more precision while roaming smoothly in the urban environment. A similar trend is visible in the other control inputs of the AAV and the gimbal as well. In all the cases, the rates are well within the specified constraints. The pitch rate of the AAV is not changing frequently since the change in altitude is required only for flying over obstacles. The yaw rate is frequently changing since the AAV cannot make sudden changes in direction due to the turn rate constraints. It can be seen that the value frequently saturates at the specified limit.

The error between the car trajectory and the center of FOV of the trained and untrained AAVs are shown in Fig. 5.10(a). Intermittent spikes of large error can be seen for the untrained AAV, meaning that the tracking failed on multiple occasions. The RL-trained AAV showed excellent performance in keeping the tracking error to a minimum. The sum of errors for an entire lap of the race is given in Fig. 5.10(b). It can be seen that the L-NMPC method provided a 70% reduction in the overall lap error compared to the standard.

5.4.3 Multiple AAVs and Cars

Now we consider simulations involving multiple AAVs and cars. We consider three cars and AAVs, among which two are fixed wings and one is a quadrotor. The quadrotor has hovering capability. Hence, the lower bound on its linear velocity is set to zero in the simulations. We do not consider altitude change for AAVs to make the problem less complex and to reduce the computation requirement. Hence, the AAVs fly at a constant altitude of 200 m. All the other values remain the same as given in Table 5.1. The



(a)

(b)

Figure 5.10: Error between the center of FOV and the car trajectory for the trained and untrained agents. (a) Instantaneous error. (b) Total error.

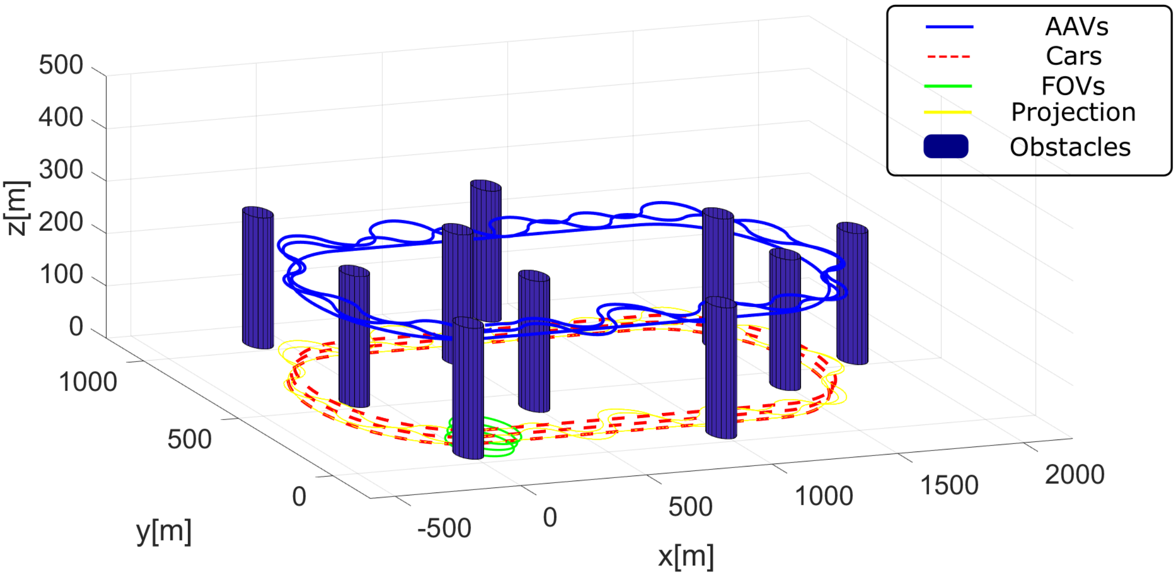
initial positions of the cars and the AAVs are selected as $(x_C, y_C)_1$ = (100,150), $(x_C, y_C)_2$ = (150,200), $(x_C, y_C)_3$ = (200,250), $(x_A, y_A, z_A)_1$ = (99,150,200), $(x_A, y_A, z_A)_2$ = (145,200,200), and $(x_A, y_A, z_A)_3$ = (195,250,200), respectively, where the quadrotor is represented by the index '1,' and the fixed-wings are represented by '2,3'.

Fig. 5.11 shows the optimal trajectory of the AAVs obtained using the proposed L-NMPC scheme. It is also compared with agents using a standard NMPC scheme with constant pre-defined weights. It can be seen that the RL-trained AAVs were able to track the cars while avoiding collision with the obstacles and with other AAVs. The relative distance between the AAVs is given in Fig. 5.12. The control inputs for the AAVs computed by the L-NMPC and the standard NMPC for all the AAVs are shown in Figures 5.13-5.18. The number of spikes in the linear velocity is lower for the RL-trained fixed-wing AAVs compared to the untrained ones. A similar trend is visible in gimbal controls as well. The pitch rates for the AAVs are almost zero since the altitude is constant. The characteristics of the quadrotor seem to be a bit similar for the trained and untrained agents. The reason for this trend is unknown and requires further investigation.

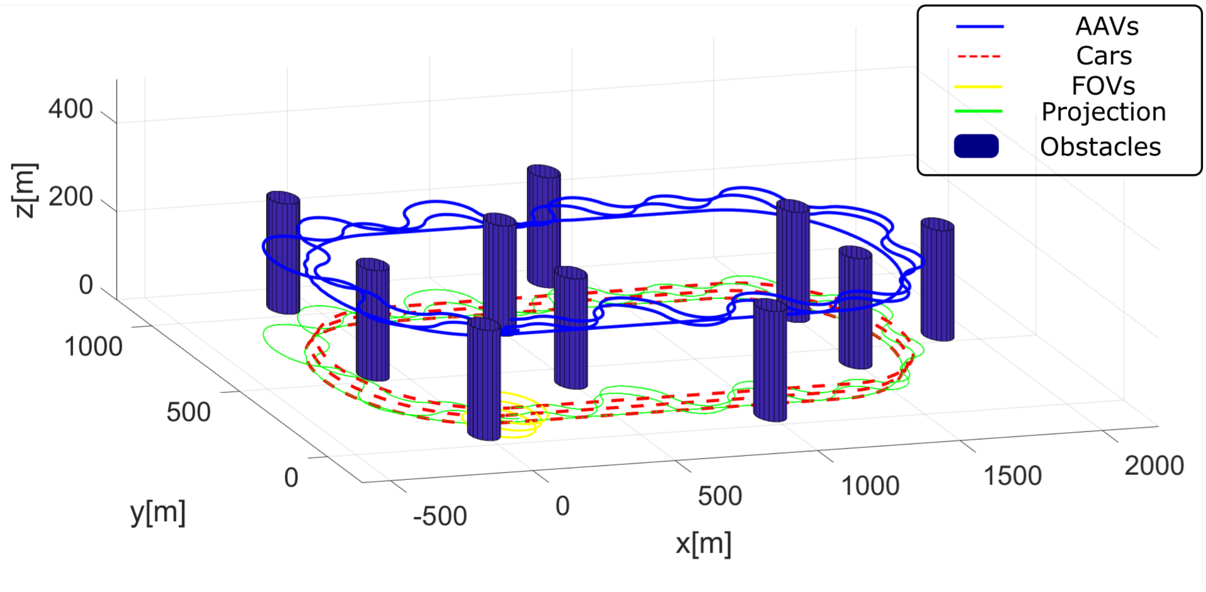
The error between the car trajectories and the center of FOVs of the trained and untrained AAVs are shown in Figures. 5.19-5.21. The total of all three AAV errors is given in Fig. 5.22. As expected, intermittent spikes of large error can be seen for the untrained AAVs, and the sum of errors is lower for all three trained AAVs. It can be seen from Fig. 5.22 that the RL-trained AAVs performed far better than the untrained ones, and the errors were reduced drastically. It can be seen that the L-NMPC method provided a 50% reduction in the overall lap error compared to the standard.

5.5 Conclusions

In this chapter, we presented a Learning-NMPC strategy for filming robot cars using gimbaled AAVs, where the AAVs increase their tracking performance iteratively using the Q-learning RL algorithm. The time-varying weights for each time step are learned to balance the NMPC cost components, resulting in an optimal balance between the multiple NMPC objectives. The proposed control scheme was found to be effective in minimizing the overall tracking errors in the race. The L-NMPC strategy was also able to handle the state and control constraints of the AAV-gimbal system and provided obstacle avoidance capability. Simulation results showed that the L-NMPC framework could increase the filming accuracy by approximately 60% compared to a standard NMPC scheme.



(a)



(b)

Figure 5.11: The trajectories of the AAVs and the cars. a) RL-trained agents b) untrained agents.

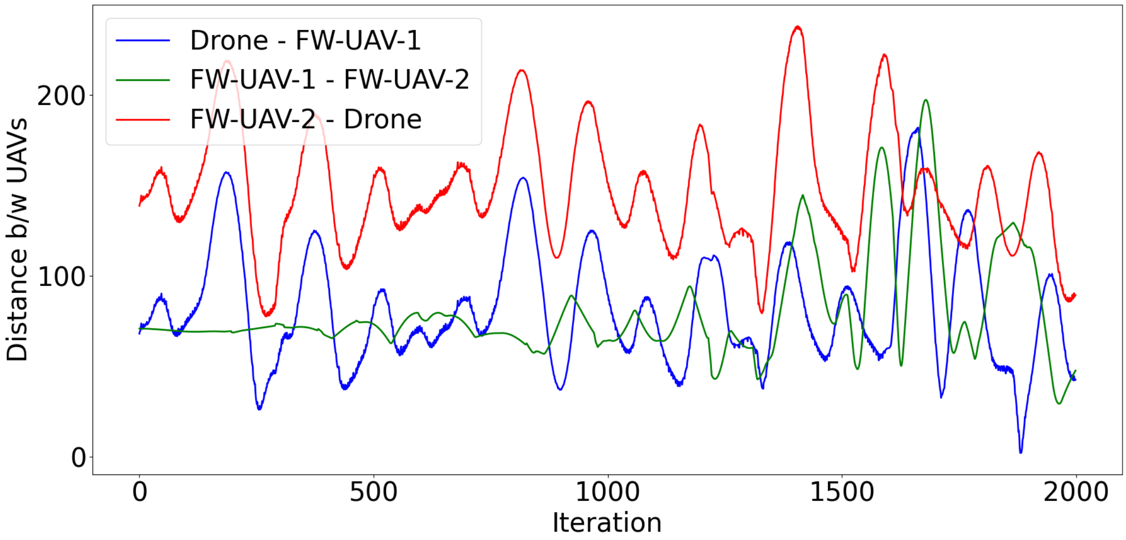


Figure 5.12: Relative distance between the AAVs.

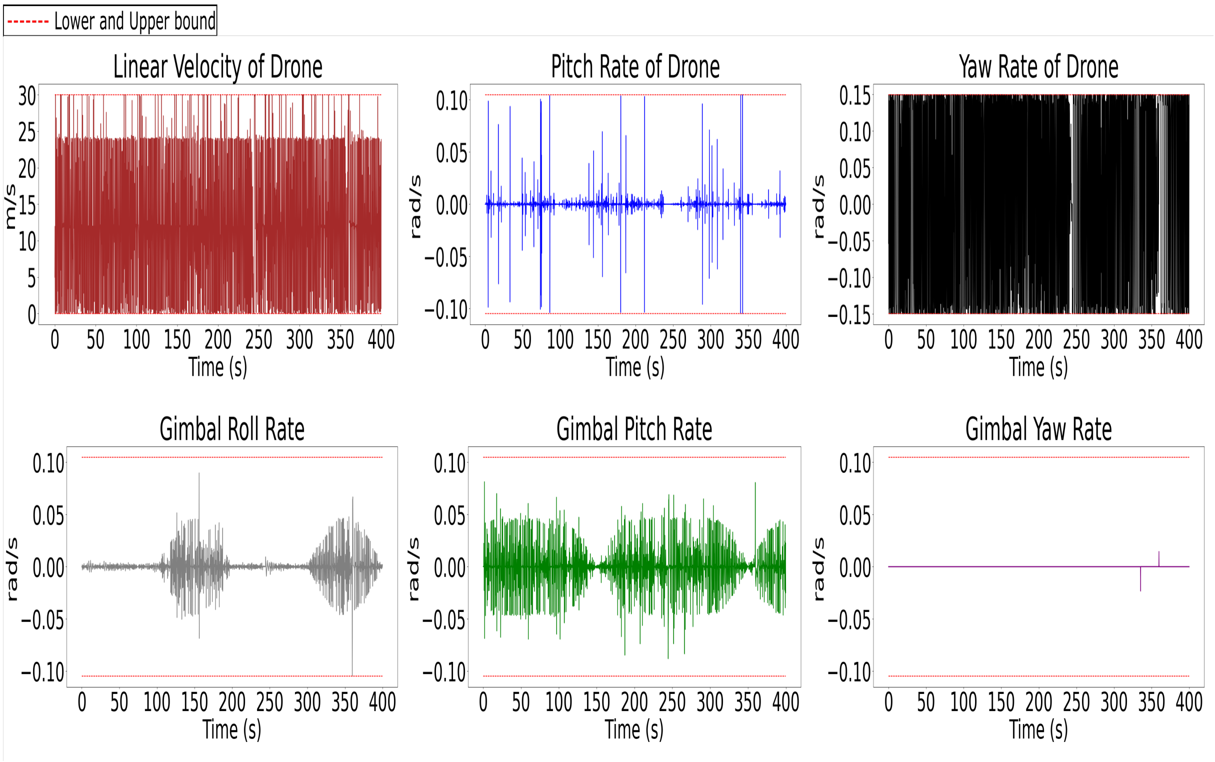


Figure 5.13: Control actions of the quadrotor with L-NMPC: (a) Linear velocity of AAV, (b) Pitch rate of AAV, (c) Yaw rate of AAV, (d) Roll rate of the gimbal, (e) Pitch rate of the gimbal, (f) Yaw rate of the gimbal.

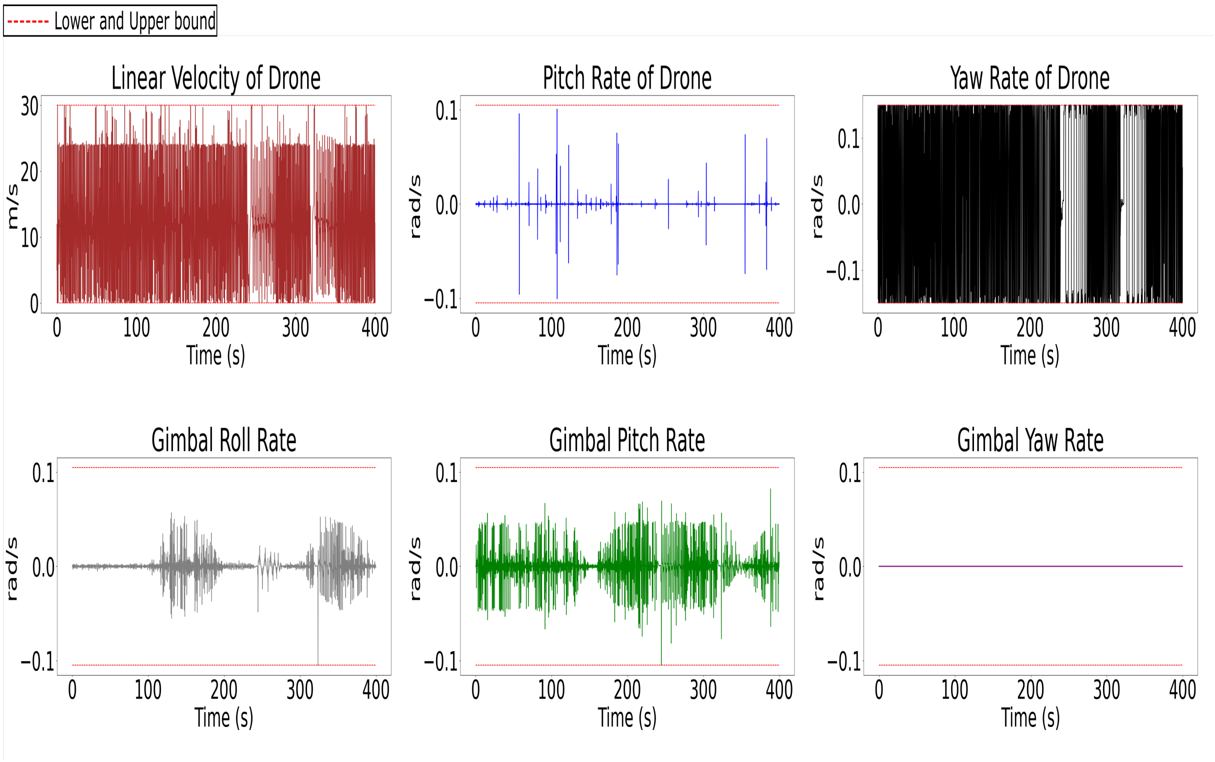


Figure 5.14: Control actions of untrained quadrotor: (a) Linear velocity of AAV, (b) Pitch rate of AAV, (c) Yaw rate of AAV, (d) Roll rate of the gimbal, (e) Pitch rate of the gimbal, (f) Yaw rate of the gimbal.

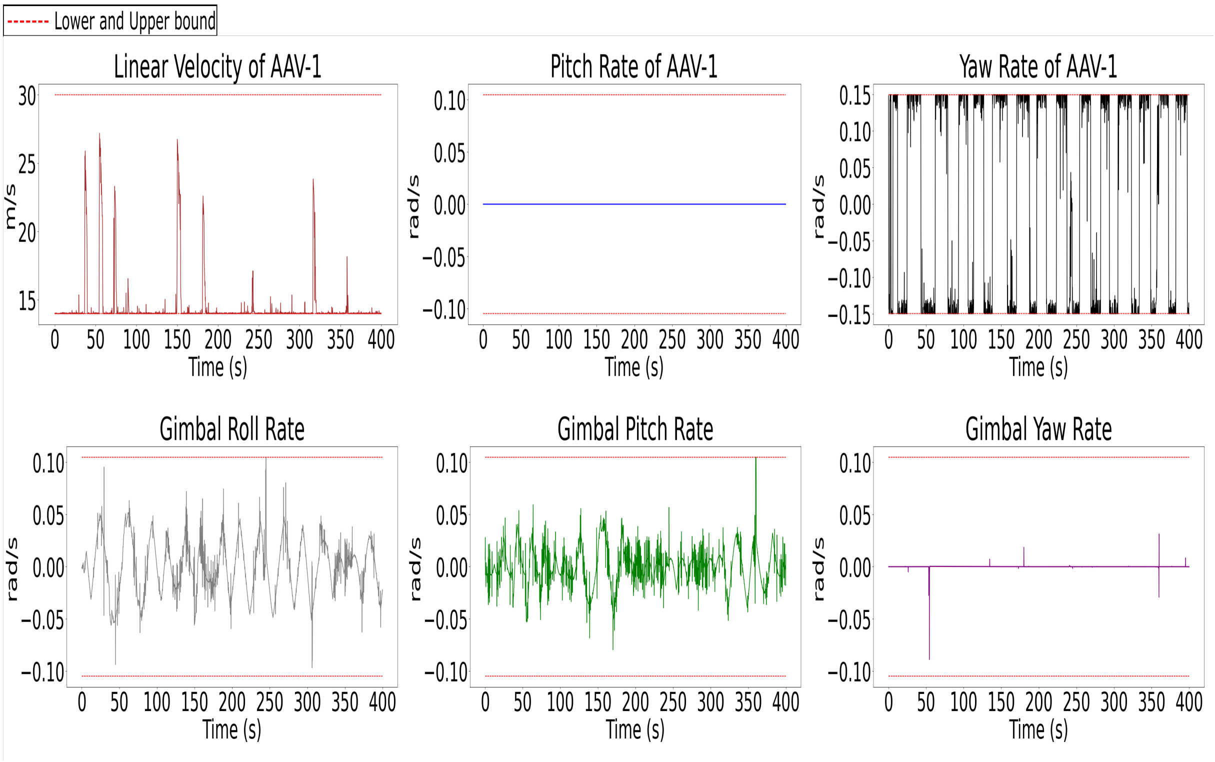


Figure 5.15: Control actions of fixed-wing-1 with L-NMPC: (a) Linear velocity of AAV, (b) Pitch rate of AAV, (c) Yaw rate of AAV, (d) Roll rate of the gimbal, (e) Pitch rate of the gimbal, (f) Yaw rate of the gimbal.

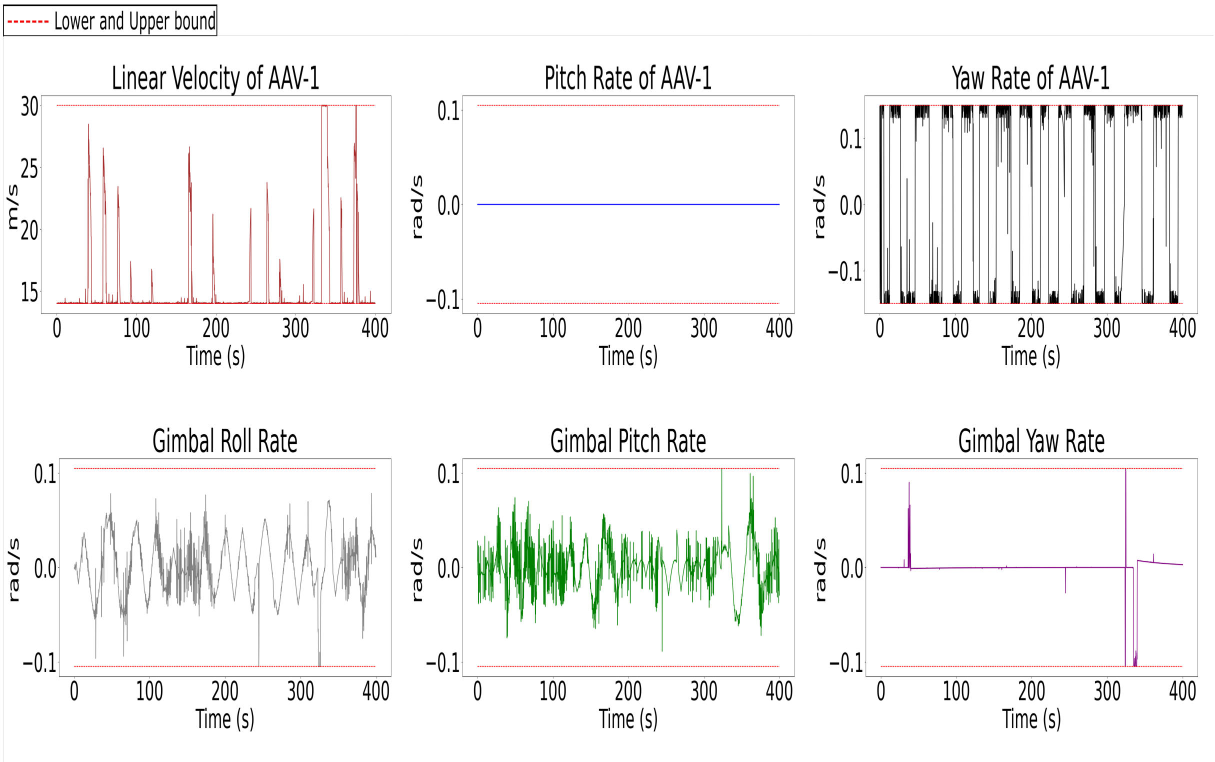


Figure 5.16: Control actions of untrained fixed-wing-1: (a) Linear velocity of AAV, (b) Pitch rate of AAV, (c) Yaw rate of AAV, (d) Roll rate of the gimbal, (e) Pitch rate of the gimbal, (f) Yaw rate of the gimbal.

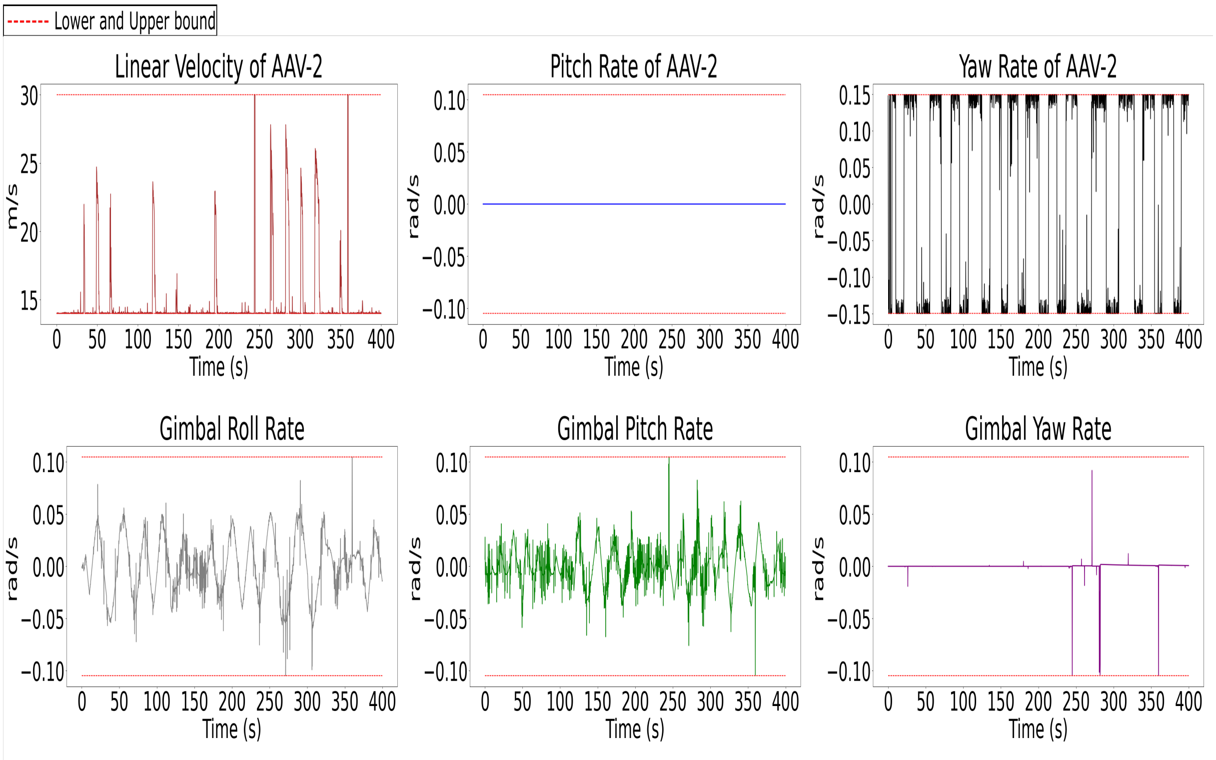


Figure 5.17: Control actions of fixed-wing-2 with L-NMPC: (a) Linear velocity of AAV, (b) Pitch rate of AAV, (c) Yaw rate of AAV, (d) Roll rate of the gimbal, (e) Pitch rate of the gimbal, (f) Yaw rate of the gimbal.

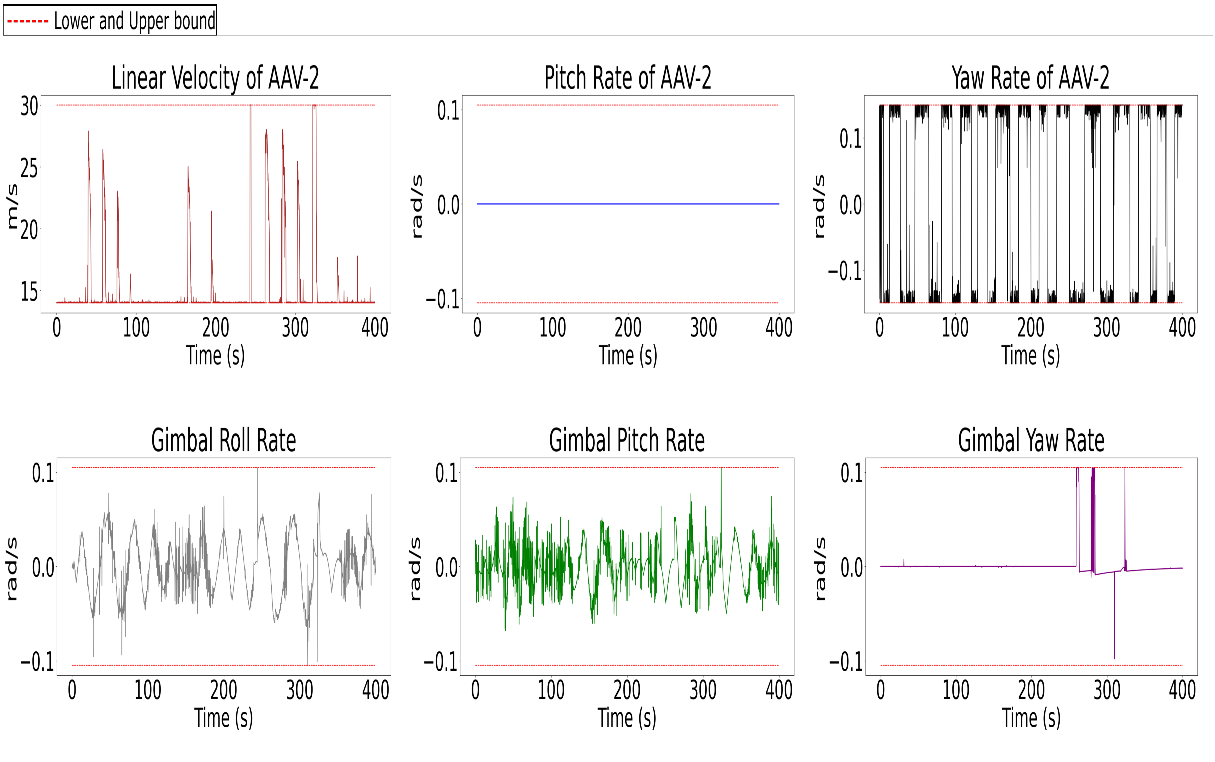
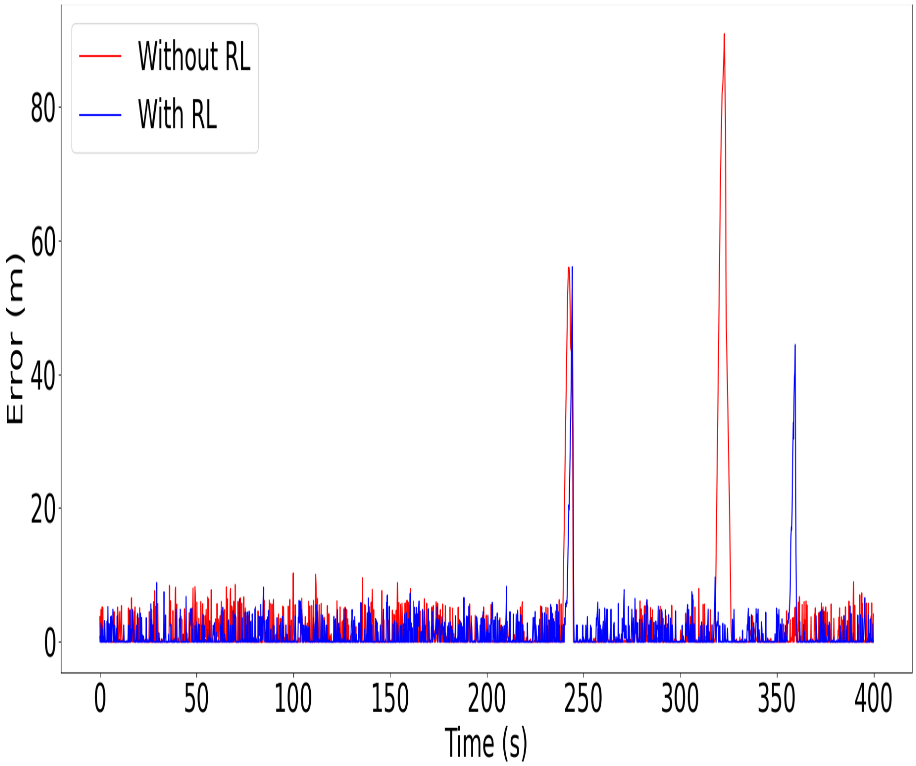
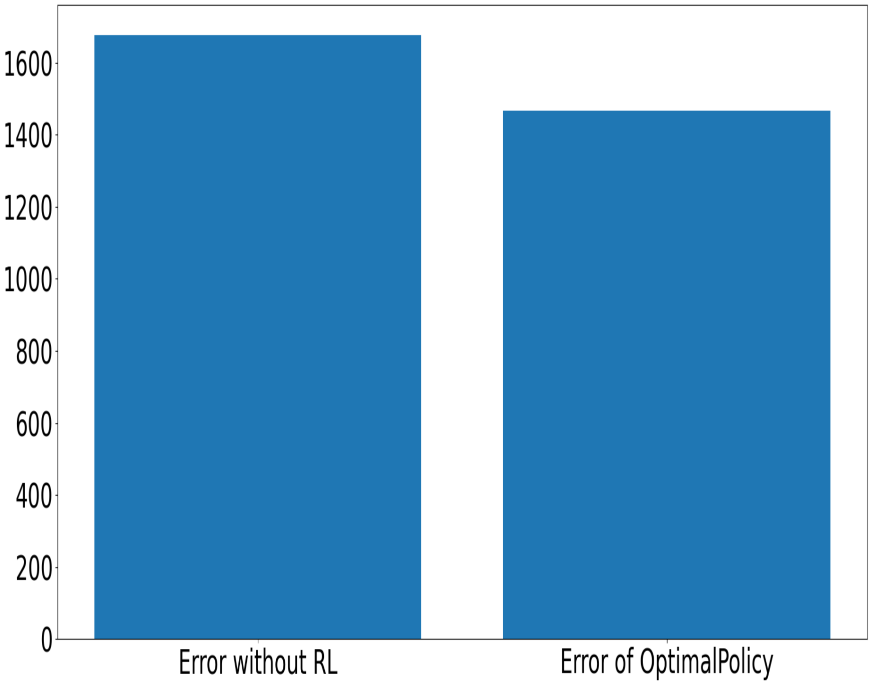


Figure 5.18: Control actions of untrained fixed-wing-2: (a) Linear velocity of AAV, (b) Pitch rate of AAV, (c) Yaw rate of AAV, (d) Roll rate of the gimbal, (e) Pitch rate of gimbal, (f) Yaw rate of the gimbal.

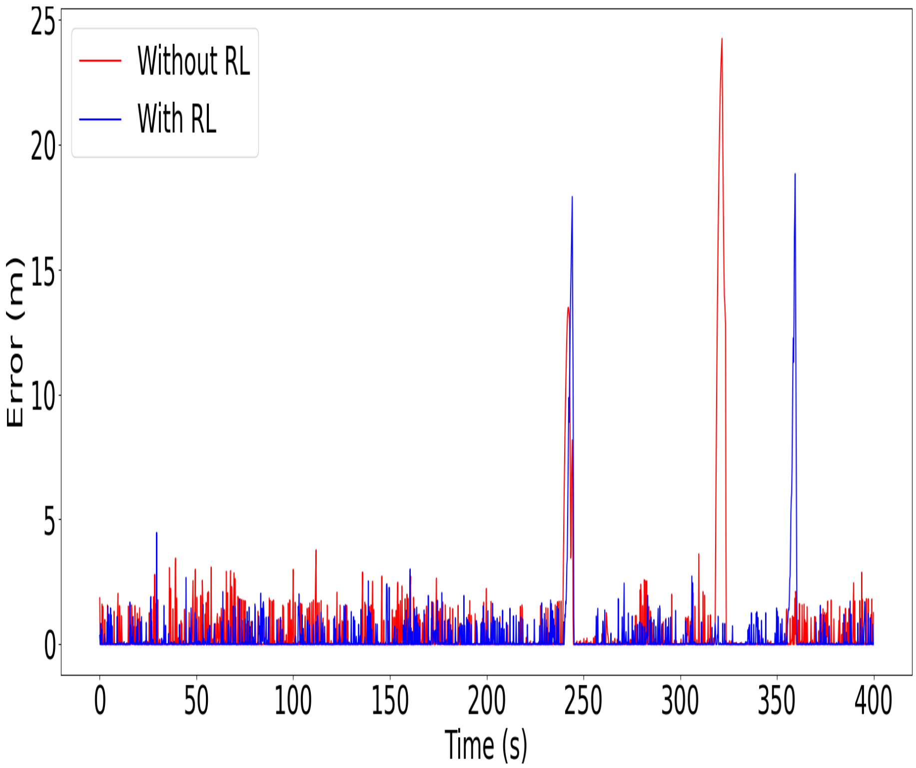


(a)

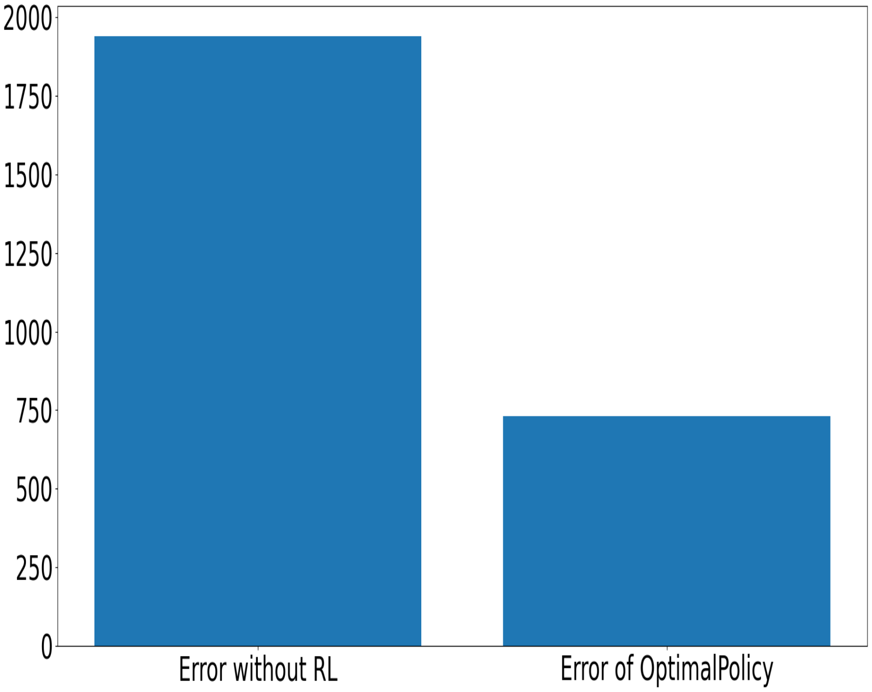


(b)

Figure 5.19: Error between the center of FOV and the car trajectory for the trained and untrained quadrotors. (a) Instantaneous error. (b) Total error.

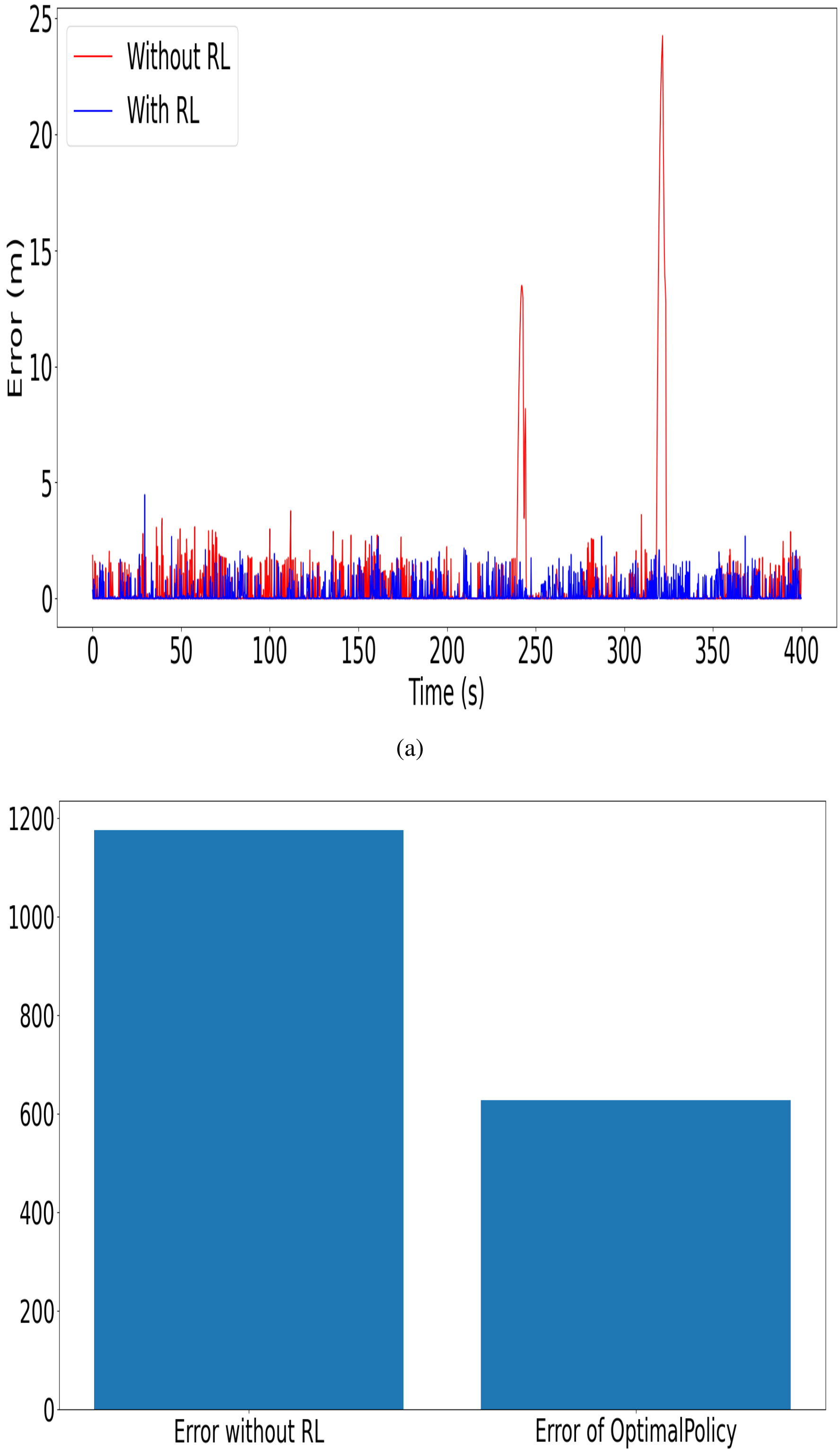


(a)



(b)

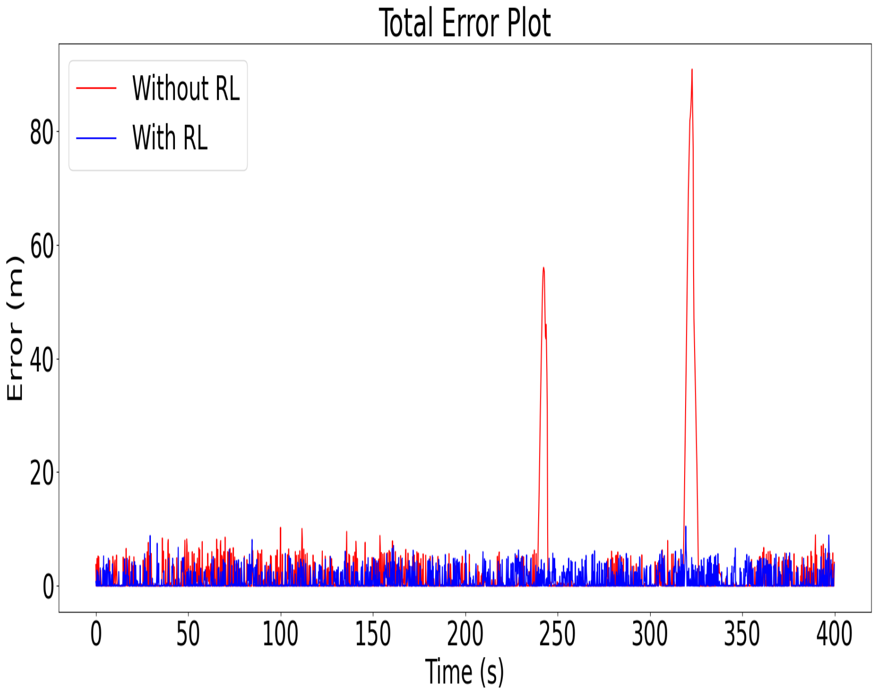
Figure 5.20: Error between the center of FOV and the car trajectory for the trained and untrained fixed-wing-1. (a) Instantaneous error. (b) Total error.



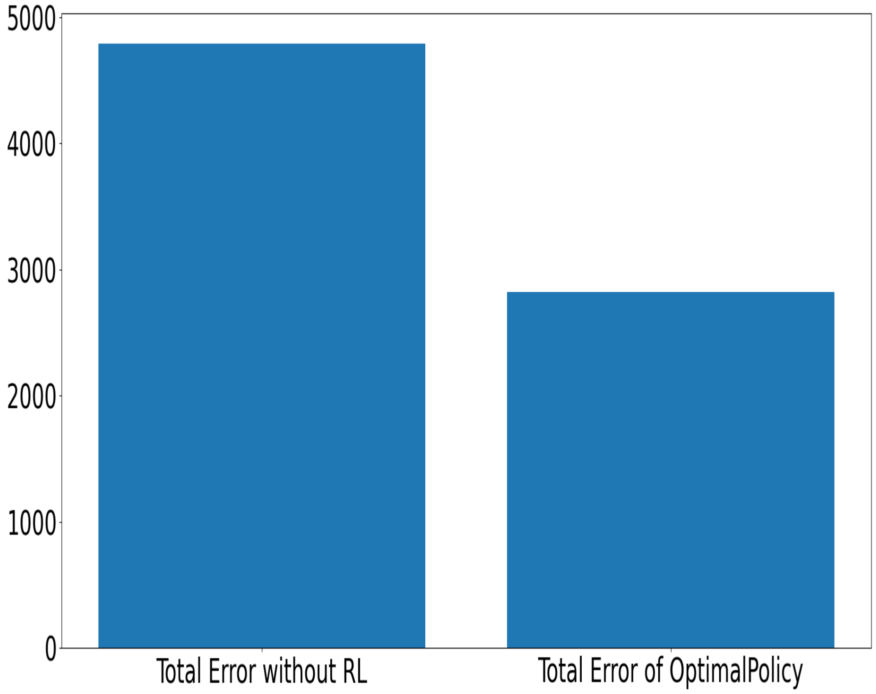
(a)

(b)

Figure 5.21: Error between the center of FOV and the car trajectory for the trained and untrained fixed-wing-2. (a) Instantaneous error. (b) Total error.



(a)



(b)

Figure 5.22: Total error between the center of FOVs and the cars for the trained and untrained AAVs. (a) Instantaneous error. (b) Total error.

Chapter 6

Conclusions and Future work

A nonlinear model predictive control strategy was proposed for the active defense of the target in a TAD problem in Chapter 2. Cooperative control commands were computed using NMPC for a target-defender pair against an individually acting attacker. The attacker states were estimated using an EKF. An analysis conducted using the Apollonius circles revealed the escape regions for the target, which was verified using simulations. The proposed scheme was found to be effective in achieving the objectives while respecting the imposed state and control constraints. The performance of the NMPC strategy was compared against CLOS and A-CLOS formulations, and the results showed that the NMPC outperformed both the CLOS and A-CLOS based strategies.

The pursuit-evasion framework was extended to four agents in Chapter 3. In this chapter, we proposed a cooperative strategy based on NMPC for the active defense of the targets in a two–targets two–attackers (2T2A) problem. The theoretical analysis using Apollonius circles given in Chapter 2 was extended to determine the escape region for the targets. The efficacy of the proposed scheme was validated using numerical simulations. The results verify the performance of the integrated EKF with the NMPC and also support the theoretical analysis.

From the analysis, it was found that the outcome of target escape or capture could be predicted based on the initial positions of the agents. This information will help the strategy planners to correctly position their defensive or offensive assets. Also, the thesis encourages the use of NMPC as a possible substitute for conventional control techniques due to its real-time implimentability, feedback structure, and superior performance.

In Chapter 4, we proposed a nonlinear model predictive control scheme combined with moving horizon estimation (MHE) to aid the cooperative localization of a group of AAVs in transit. The controller used an approximate analytical expression for calculating the expected covariance of the vehicles, which was derived using theoretical analysis conducted using the observability graphs. Near-optimal paths were found for the vehicles while satisfying various state and localization constraints. Simulations were conducted to analyze the role of the prediction horizon on the optimality of the vehicle paths and the required computation time. The proposed moving horizon estimator was compared with the EKF. MHE outperformed the EKF with lower estimation error values at a short additional computation time. A comparison between cooperative and non-cooperative vehicles revealed the significance of cooperation in determining paths under localization constraints. The proposed scheme can make huge impacts in the area of urban air mobility. Precise localization is very important for vehicles flying in cities. The thesis proposes a complete framework for vehicles to suitably plan their paths while maintaining localization accuracy through cooperation and the use of existing features instead of relying on GNSS.

Finally, in Chapter 5, we combined the reinforcement learning technique with the NMPC to create a learning-based controller for the tracking of cars using gimbaled AAVs. The weights of the cost function were learned online using the information obtained from the iterations of the race. The proposed L-NMPC scheme was found to have better tracking performance than a standard NMPC scheme with manually tuned weights. The proposed scheme can be used in general for any persistent monitoring applications of iterative nature. Not only it helps reduce the human workload in tuning the weights but it also improves the performance compared to manual tuning.

The main limitation of this thesis is the use of point-mass models and simple kinematic equations for the agents. According to the application and the agent type, corresponding dynamics could be used to increase the performance of the formulations. The use of actual dynamics will also put extra constraints on the system states and control inputs. The NMPC is capable of handling these difficulties, and we plan to take this into effect for our future work. Other possible extensions for the different problems are explained in the following section.

Future work

Possible extensions for the three-agent problem discussed in Chapter 2 are to broaden the framework to three dimensions taking the terrain map into account. Also, if urban environments are considered, then obstacle avoidance constraints need to be included for the agents. The current NMPC solver suffers from a high value of computation time. If 3D environments with obstacle avoidance constraints are considered, or the number of agents is increased to formulate a multi-pursuer multi-evader problem, the computation requirement will increase drastically, preventing real-time implementation. Hence, we would like to drive our approach toward the use of fast-MPC based solvers for speeding up the computations in the proposed strategy.

There can be several extensions of the work proposed in Chapter 3. The formulation can be extended to 3D, considering the altitude information. The existing 2T2A problem can be developed into a general framework consisting of multiple attackers, targets, and defenders. The agent assignments, which attacker should pursue which target and which defender should intercept which attacker, can be modeled and solved as a task assignment problem. Another extension is to relax some of the assumptions on agent velocities like keeping the attacker velocity $\dot{v}_A$ constant, $\dot{v}_A = 0$, but v_A is not known, or even the case where agent velocities are not constant. We would also like to do a formal analysis of the NMPC stability and the implications of the EKF on the stability.

The approach proposed in Chapter 4 can be extended in several directions. One potential extension is to determine how many landmarks are sufficient for a given vehicle to reach the destination while satisfying localization constraints. A limitation of inter-vehicle localization is that the errors propagate from vehicle to vehicle since the relative position reference is estimated and not a known true value like the landmarks. This problem can only be overcome by ensuring that the reference vehicle is always connected to the number of landmarks required for maintaining near-true accuracy. Finding the exact number of landmarks and their positions required for finding a feasible trajectory with a given source and destination points, the number of vehicles, and the size of the area under consideration is still an open problem for research. Also, since the urban environment contains a lot of buildings and other obstructions, another prospect is to incorporate obstacle avoidance techniques into the path planning algorithm. The current approach uses centralized schemes for estimation and control. The use of decentralized schemes will distribute the computation requirement and increase the speed of computation. Experimental validation is another direction to implement the algorithms in real-world vehicles.

The L-NMPC framework proposed in Chapter 5 can be improved in many directions. Since a race car usually does not move at a constant speed, a variable velocity race car can be considered for the target tracking problem. Also, wind disturbances should be taken into effect when formulating the control law for AAVs. More complex race tracks with hairpin bends and sharp turns will make the problem more challenging and interesting. The problem can also be converted to a real-time assignment problem, where the AAVs switch their corresponding targets to meet certain constraints, such as collision avoidance, or to improve the overall tracking performance. More simulations consisting of different configurations are required to find in-depth the cooperative aspects of the AAVs.

Bibliography

[1] W. Ren and R. W. Beard, “Consensus seeking in multiagent systems under dynamically changing interaction topologies,” *IEEE Transactions on automatic control*, vol. 50, no. 5, pp. 655–661, 2005.

[2] M. Egerstedt and X. Hu, “Formation constrained multi-agent control,” *IEEE transactions on robotics and automation*, vol. 17, no. 6, pp. 947–951, 2001.

[3] J. A. Fax and R. M. Murray, “Information flow and cooperative control of vehicle formations,” *IEEE transactions on automatic control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[5] I. I. Hussein and D. M. Stipanovic, “Effective coverage control for mobile sensor networks with guaranteed collision avoidance,” *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 642–657, 2007.

[6] Y. Jin, Y. Liao, A. A. Minai, and M. M. Polycarpou, “Balancing search and target response in cooperative unmanned aerial vehicle (uav) teams,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 3, pp. 571–587, 2006.

[7] S. Martínez and F. Bullo, “Optimal sensor placement and motion coordination for target tracking,” *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.

[8] J. Hu and G. Feng, “Distributed tracking control of leader–follower multi-agent systems under noisy measurement,” *Automatica*, vol. 46, no. 8, pp. 1382–1387, 2010.

[9] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[10] M. Shalaby, C. C. Cossette, J. R. Forbes, and J. Le Ny, “Relative position estimation in multi-agent systems using attitude-coupled range measurements,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4955–4961, 2021.

[11] G. M. Atınç, D. M. Stipanović, and P. G. Voulgaris, “Supervised coverage control of multi-agent systems,” *Automatica*, vol. 50, no. 11, pp. 2936–2942, 2014.

[12] T. M. Cheng and A. V. Savkin, “Decentralized control of multi-agent systems for swarming with a given geometric pattern,” *Computers & Mathematics with Applications*, vol. 61, no. 4, pp. 731–744, 2011.

[13] J. Shamma, *Cooperative control of distributed multi-agent systems*. John Wiley & Sons, 2008.

[14] F. L. Lewis, H. Zhang, K. Hengster-Movric, and A. Das, *Cooperative control of multi-agent systems: optimal and adaptive design approaches*. Springer Science & Business Media, 2013.

[15] K. H. Movric and F. L. Lewis, "Cooperative optimal control for multi-agent systems on directed graph topologies," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 769–774, 2013.

[16] C. G. Cassandras, X. Lin, and X. Ding, "An optimal control approach to the multi-agent persistent monitoring problem," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 947–961, 2012.

[17] G. Ferrari-Trecate, L. Galbusera, M. P. E. Marciandi, and R. Scattolini, "Model predictive control schemes for consensus in multi-agent systems with single-and double-integrator dynamics," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2560–2572, 2009.

[18] J. Zhan and X. Li, "Consensus of sampled-data multi-agent networking systems via model predictive control," *Automatica*, vol. 49, no. 8, pp. 2502–2507, 2013.

[19] A. Maxim, C. M. Ionescu, C. F. Caruntu, C. Lazar, and R. De Keyser, "Reference tracking using a non-cooperative distributed model predictive control algorithm," *IFAC-PapersOnLine*, vol. 49, no. 7, pp. 1079–1084, 2016.

[20] C. E. Luis, M. Vukosavljev, and A. P. Schoellig, "Online trajectory generation with distributed model predictive control for multi-robot motion planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 604–611, 2020.

[21] A. Nikou and D. V. Dimarogonas, "Decentralized tube-based model predictive control of uncertain nonlinear multiagent systems," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 10, pp. 2799–2818, 2019.

[22] J. Kuhn, C. Reinl, and O. Von Stryk, "Predictive control for multi-robot observation of multiple moving targets based on discrete-continuous linear models," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 257–262, 2011.

[23] T. P. Nascimento, A. P. Moreira, and A. G. S. Conceição, "Multi-robot nonlinear model predictive formation control: Moving target and target absence," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1502–1515, 2013.

[24] S. Emoto, I. Akkaya, and E. A. Lee, "Information seeking and model predictive control of a cooperative multi-robot system," *Artificial Life and Robotics*, vol. 21, pp. 393–398, 2016.

[25] S.-M. Lee and H. Myung, "Cooperative particle swarm optimization-based predictive controller for multi-robot formation," in *Intelligent Autonomous Systems 12: Volume 2 Proceedings of the 12th International Conference IAS-12, held June 26-29, 2012, Jeju Island, Korea*. Springer, 2013, pp. 533–541.

[26] S. Bhattacharya, T. Başar, and M. Falcone, "Surveillance for security as a pursuit-evasion game," in *International Conference on Decision and Game Theory for Security*. Springer, 2014, pp. 370–379.

[27] I. E. Weintraub, M. Pachter, and E. Garcia, "An introduction to pursuit-evasion differential games," in *American Control Conference (ACC)*, 2020, pp. 1049–1066.

[28] V. Sunkara, A. Chakravarthy, and D. Ghose, "Pursuit evasion games using collision cones," in *AIAA Guidance, Navigation, and Control Conference*, 2018, p. 2108.

[29] A. Jagat and A. J. Sinclair, "Nonlinear control for spacecraft pursuit-evasion game using the state-dependent riccati equation method," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 6, pp. 3032–3042, 2017.

[30] J. Breakwell and P. Hagedorn, "Point capture of two evaders in succession," *Journal of Optimization Theory and Applications*, vol. 27, no. 1, pp. 89–97, 1979.

[31] Z. E. Fuchs, P. P. Khargonekar, and J. Evers, "Cooperative defense within a single-pursuer, two-evader pursuit evasion differential game," in *49th Conference on Decision and Control, Atlanta, Georgia, USA*, 2010, pp. 3091–3097.

[32] S. Pan, H. Huang, J. Ding, W. Zhang, C. J. Tomlin *et al.*, "Pursuit, evasion and defense in the plane," in *American Control Conference, Montreal, Canada*, 2012, pp. 4167–4173.

[33] W. Sun, P. Tsiotras, T. Lolla, D. N. Subramani, and P. F. Lermusiaux, "Multiple-pursuer/one-evader pursuit–evasion game in dynamic flowfields," *Journal of guidance, control, and dynamics*, vol. 40, no. 7, pp. 1627–1637, 2017.

[34] M. Ramana and M. Kothari, "Pursuit strategy to capture high-speed evaders using multiple pursuers," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 1, pp. 139–149, 2017.

[35] M. Pachter, A. V. Moll, E. Garcia, D. Casbeer, and D. Milutinović, "Cooperative pursuit by multiple pursuers of a single evader," *Journal of Aerospace Information Systems*, pp. 1–19, 2020.

[36] R. Isaacs, *Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization*. Courier Corporation, 1999.

[37] A. Ratnoo and T. Shima, "Line-of-sight interceptor guidance for defending an aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 2, pp. 522–532, 2011.

[38] T. Yamasaki, S. Balakrishnan, and H. Takano, "Modified command to line-of-sight intercept guidance for aircraft defense," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 3, pp. 898–902, 2013.

[39] O. Prokopov and T. Shima, "Linear quadratic optimal cooperative strategies for active aircraft protection," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 3, pp. 753–764, 2013.

[40] T. Shima, "Optimal cooperative pursuit and evasion strategies against a homing missile," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 2, pp. 414–425, 2011.

[41] E. Garcia, D. W. Casbeer, and M. Pachter, "Cooperative strategies for optimal aircraft defense from an attacking missile," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 8, pp. 1510–1520, 2015.

[42] ——, "Design and analysis of state-feedback optimal strategies for the differential game of active defense," *IEEE Transactions on Automatic Control*, vol. 64, no. 2, pp. 553–568, 2018.

[43] ——, "The complete differential game of active target defense," *Journal of Optimization Theory and Applications*, pp. 1–25, 2021.

[44] A. Perelman, T. Shima, and I. Rusnak, "Cooperative differential games strategies for active aircraft protection from a homing missile," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 3, pp. 761–773, 2011.

[45] A. Ratnoo and T. Shima, "Guidance strategies against defended aerial targets," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 4, pp. 1059–1068, 2012.

[46] G. Eloy, D. Casbeer, and M. Pachter, “Active target defense differential game with a fast defender,” *IET Control Theory and Applications*, vol. 17, no. 11, p. 298, 2017.

[47] S. Rubinsky and S. Gutman, “Three-player pursuit and evasion conflict,” *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 98–110, 2014.

[48] S. R. Kumar and T. Shima, “Cooperative nonlinear guidance strategies for aircraft defense,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 1, pp. 124–138, 2017.

[49] T. Yamasaki and S. Balakrishnan, “Triangle intercept guidance for aerial defense,” in *AIAA Guidance, Navigation, and Control Conference*, 2010, p. 7876.

[50] E. Garcia, D. W. Casbeer, Z. E. Fuchs, and M. Pachter, “Cooperative missile guidance for active defense of air vehicles,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 2, pp. 706–721, 2017.

[51] M. Weiss, T. Shima, D. Castaneda, and I. Rusnak, “Combined and cooperative minimum-effort guidance algorithms in an active aircraft defense scenario,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 5, pp. 1241–1254, 2017.

[52] M. Harel, A. Moshaiov, and D. Alkaher, “Rationalizable strategies for the navigator–target–missile game,” *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 6, pp. 1129–1142, 2020.

[53] V. S. Chipade and D. Panagou, “Multiplayer target-attacker-defender differential game: Pairing allocations and control strategies for guaranteed intercept,” in *AIAA Scitech Forum*, 2019, p. 0658.

[54] A. Manoharan, M. Singh, A. Alessandretti, J. G. Manathara, S. Prusty, N. Mohanty, I. S. Kumar, A. Sahoo, and P. B. Sujit, “Nmpc based approach for cooperative target defence,” in *American Control Conference (ACC)*, 2019, pp. 5292–5297.

[55] S. G. Manyam, D. Casbeer, A. L. Von Moll, E. Garcia, and S. Rasmussen, “Coordinating defender path planning for optimal target-attacker-defender game,” in *AIAA Scitech Forum*, 2019, p. 0388.

[56] D. W. Casbeer, E. Garcia, and M. Pachter, “The target differential game with two defenders,” *Journal of Intelligent & Robotic Systems*, vol. 89, no. 1, pp. 87–106, 2018.

[57] E. Garcia, D. W. Casbeer, A. Von Moll, and M. Pachter, “Multiple pursuer multiple evader differential games,” *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2345–2350, 2020.

[58] Z. W. Tan, R. Fonod, and T. Shima, “Cooperative guidance law for target pair to lure two pursuers into collision,” *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 8, pp. 1687–1699, 2018.

[59] E. Garcia, D. W. Casbeer, M. Pachter, J. W. Curtis, and E. Doucette, “A two-team linear quadratic differential game of defending a target,” in *American Control Conference (ACC)*, 2020, pp. 1665–1670.

[60] H. Liang, J. Wang, J. Liu, and P. Liu, “Guidance strategies for interceptor against active defense spacecraft in two-on-two engagement,” *Aerospace Science and Technology*, vol. 96, p. 105529, 2020.

[61] FAA. (2020) Urban air mobility, concept of operations v1.0. [Online]. Available: https://nari.arc.nasa.gov/sites/default/files/attachments/UAM_ConOps_v1.0.pdf

[62] SESAR. (2016) European drones outlook study. [Online]. Available: https://www.sesarju.eu/sites/default/files/documents/reports/European_Drones_Outlook_Study_2016.pdf

[63] M. Betke and L. Gurvits, "Mobile robot localization using landmarks," *IEEE transactions on robotics and automation*, vol. 13, no. 2, pp. 251–263, 1997.

[64] S. Thrun, "Finding landmarks for mobile robot navigation," in *Proc. of International Conference on Robotics and Automation*, vol. 2, 1998, pp. 958–963.

[65] H. Lategahn and C. Stiller, "Vision-only localization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 1246–1257, 2014.

[66] G.-H. Kim, J.-S. Kim, and K.-S. Hong, "Vision-based simultaneous localization and mapping with two cameras," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 1671–1676.

[67] S. Krishnan, P. Sharma, Z. Guoping, and O. H. Woon, "A uwb based localization system for indoor robot navigation," in *International Conference on Ultra-Wideband*, 2007, pp. 77–82.

[68] S. Zhang, R. Han, W. Huang, S. Wang, and Q. Hao, "Linear bayesian filter based low-cost uwb systems for indoor mobile robot localization," in *IEEE Sensors*, 2018, pp. 1–4.

[69] R. Kurazume and S. Hirose, "Study on Cooperative Positioning System - Optimum Moving Strategies for CPS-III," *IEEE International Conference on Robotics and Automation*, vol. 4, pp. 2896–2903, 1998.

[70] S. Roumeliotis and G. Bekey, "Collective localization: a distributed Kalman filter approach to localization of groups of mobile robots," *IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2958–2965, 2000.

[71] J. Spletzer, A. K. Das, R. Fierro, C. J. Taylor, V. Kumar, and J. P. Ostrowski, "Cooperative localization and control for multi-robot manipulation," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2, pp. 631–636, 2001.

[72] A. I. Mourikis and S. I. Roumeliotis, "Multirobot Cooperative Localization," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 666–681, 2006.

[73] E. D. Nerurkar, S. I. Roumeliotis, and A. Martinelli, "Distributed maximum a posteriori estimation for multi-robot cooperative localization," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1402–1409, 2009.

[74] J. Wan, L. Zhong, and F. Zhang, "Cooperative localization of multi-uavs via dynamic nonparametric belief propagation under gps signal loss condition," *International Journal of Distributed Sensor Networks*, vol. 10, no. 2, p. 562380, 2014.

[75] S. Minaeian, J. Liu, and Y.-J. Son, "Vision-based target detection and localization via a team of cooperative uav and ugvs," *IEEE Transactions on systems, man, and cybernetics: systems*, vol. 46, no. 7, pp. 1005–1016, 2016.

[76] M. Frohle, C. Lindberg, and H. Wymeersch, "Cooperative localization of vehicles without intervehicle measurements," in *IEEE Wireless Communications and Networking Conference (WCNC)*, 2018, pp. 1–6.

[77] C. Pierre, R. Chapuis, R. Aufrère, J. Laneurit, and C. Debain, "Range-only based cooperative localization for mobile robots," in *21st international conference on information fusion (FUSION)*, 2018, pp. 1933–1939.

[78] J. Liu, J. Pu, L. Sun, and Y. Zhang, “Multi-robot cooperative localization with range-only measurement by uwb,” in *Chinese Automation Congress (CAC)*, 2018, pp. 2809–2813.

[79] K. Guo, X. Li, and L. Xie, “Ultra-wideband and odometry-based cooperative relative localization with application to multi-uav formation control,” *IEEE transactions on cybernetics*, vol. 50, no. 6, pp. 2590–2603, 2019.

[80] J. Zhu and S. S. Kia, “Cooperative localization under limited connectivity,” *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1523–1530, 2019.

[81] C. Zheng, L. Li, F. Xu, F. Sun, and M. Ding, “Evolutionary route planner for unmanned air vehicles,” *IEEE Transactions on robotics*, vol. 21, no. 4, pp. 609–620, 2005.

[82] A. A. Agha-Mohammadi, N. K. Ure, J. P. How, and J. Vian, “Health aware stochastic planning for persistent package delivery missions using quadrotors,” *IEEE International Conference on Intelligent Robots and Systems*, pp. 3389–3396, 2014.

[83] N. Mathew, S. L. Smith, and S. L. Waslander, “Planning Paths for Package Delivery in Heterogeneous Multirobot Teams,” *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1298–1308, 2015.

[84] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, “Heuristic approaches in robot path planning: A survey,” *Robotics and Autonomous Systems*, vol. 86, pp. 13–28, 2016.

[85] S. D. Bopardikar, B. Englot, and A. Speranzon, “Multiobjective path planning: Localization constraints and collision probability,” *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 562–577, 2015.

[86] S. Singh and P. B. Sujit, “Landmarks based path planning for uavs in gps-denied areas,” *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 396–400, 2016.

[87] K. Sundar, S. Rathinam, and R. Sharma, “Path planning for unmanned vehicles with localization constraints,” *Optimization Letters*, vol. 13, no. 5, pp. 993–1009, 2019.

[88] B. Irani, J. Wang, and W. Chen, “A localizability constraint-based path planning method for autonomous vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2593–2604, 2018.

[89] Y. Yang, J. Khalife, J. J. Morales, and Z. M. Kassas, “Uav waypoint opportunistic navigation in gnss-denied environments,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 1, pp. 663–678, 2021.

[90] A. Manoharan, R. Sharma, and P. B. Sujit, “Nonlinear model predictive control to aid cooperative localization,” in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019, pp. 26–32.

[91] J. B. Rawlings and B. R. Bakshi, “Particle filtering and moving horizon estimation,” *Computers & chemical engineering*, vol. 30, no. 10-12, pp. 1529–1541, 2006.

[92] S. Wang, L. Chen, D. Gu, and H. Hu, “An optimization based moving horizon estimation with application to localization of autonomous underwater vehicles,” *Robotics and Autonomous Systems*, vol. 62, no. 10, pp. 1581–1596, 2014.

[93] M. W. Mehrez, G. K. Mann, and R. G. Gosine, “An optimization based approach for relative localization and relative tracking control in multi-robot systems,” *Journal of Intelligent & Robotic Systems*, vol. 85, no. 2, pp. 385–408, 2017.

[94] I. K. Erunsal, A. Martinoli, and R. Ventura, “Decentralized nonlinear model predictive control for 3d formation of multirotor micro aerial vehicles with relative sensing and estimation,” in *International Symposium on Multi-Robot and Multi-Agent Systems*, 2019, pp. 176–178.

[95] S. Liu, G. Zhao, Y. He, and C. Gao, “Decentralized moving horizon estimation for networked navigation system with packet dropouts,” in *39th Chinese Control Conference (CCC)*, 2020, pp. 3381–3384.

[96] Y. Kim and M. Mesbahi, “On maximizing the second smallest eigenvalue of a state-dependent graph laplacian,” *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, 2006.

[97] M. C. De Gennaro and A. Jadbabaie, “Decentralized control of connectivity for multi-agent systems,” in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 3628–3633.

[98] H. Choi and Y. Kim, “Uav guidance using a monocular-vision sensor for aerial target tracking,” *Control Engineering Practice*, vol. 22, pp. 10–19, 2014.

[99] H. Oh, S. Kim, H.-S. Shin, B. A. White, A. Tsourdos, and C. A. Rabbath, “Rendezvous and standoff target tracking guidance using differential geometry,” *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1, pp. 389–405, 2013.

[100] P. Theodorakopoulos and S. Lacroix, “A strategy for tracking a ground target with a uav,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 1254–1259.

[101] X. Zhao, F. Pu, Z. Wang, H. Chen, and Z. Xu, “Detection, tracking, and geolocation of moving vehicle from uav using monocular camera,” *IEEE Access*, vol. 7, pp. 101 160–101 170, 2019.

[102] S. Zhu, D. Wang, and C. B. Low, “Ground target tracking using uav with input constraints,” *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1, pp. 417–429, 2013.

[103] R. Bonatti, W. Wang, C. Ho, A. Ahuja, M. Gschwindt, E. Camci, E. Kayacan, S. Choudhury, and S. Scherer, “Autonomous aerial cinematography in unstructured environments with learned artistic decision-making,” *Journal of Field Robotics*, vol. 37, no. 4, pp. 606–641, 2020.

[104] Z. Li, N. Hovakimyan, V. Dobrokhodov, and I. Kaminer, “Vision-based target tracking and motion estimation using a small uav,” in *IEEE Conference on Decision and Control (CDC)*, 2010, pp. 2505–2510.

[105] S. A. Quintero and J. P. Hespanha, “Vision-based target tracking with a small uav: Optimization-based control strategies,” *Control Engineering Practice*, vol. 32, pp. 28–42, 2014.

[106] P. Theodorakopoulos and S. Lacroix, “Uav target tracking using an adversarial iterative prediction,” in *IEEE International Conference on Robotics and Automation*, 2009, pp. 2866–2871.

[107] Y. Watanabe and P. Fabiani, “Optimal guidance design for uav visual target tracking in an urban environment,” *IFAC Proceedings Volumes*, vol. 43, no. 15, pp. 69–74, 2010.

[108] A. Altan and R. Hacıoğlu, “Model predictive control of three-axis gimbal system mounted on uav for real-time target tracking under external disturbances,” *Mechanical Systems and Signal Processing*, vol. 138, p. 106548, 2020.

[109] P. Mali, A. K. Singh, M. Krishnal, and P. B. Sujit, “Model predictive control for target tracking in 3d with a downward facing camera equipped fixed wing aerial vehicle,” in *16th International Conference on Automation Science and Engineering (CASE)*, 2020, pp. 165–172.

[110] P. Tyagi, Y. Kumar, and P. B. Sujit, “Nmpc-based uav 3d target tracking in the presence of obstacles and visibility constraints,” in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 858–867.

[111] J. H. Lee and K. S. Lee, “Iterative learning control applied to batch processes: An overview,” *Control Engineering Practice*, vol. 15, no. 10, pp. 1306–1318, 2007.

[112] Y. Wang, F. Gao, and F. J. Doyle III, “Survey on iterative learning control, repetitive control, and run-to-run control,” *Journal of Process Control*, vol. 19, no. 10, pp. 1589–1600, 2009.

[113] C.-Y. Lin, L. Sun, and M. Tomizuka, “Robust principal component analysis for iterative learning control of precision motion systems with non-repetitive disturbances,” in *American Control Conference (ACC)*, 2015, pp. 2819–2824.

[114] K. S. Lee, I.-S. Chin, H. J. Lee, and J. H. Lee, “Model predictive control technique combined with iterative learning for batch processes,” *AIChE Journal*, vol. 45, no. 10, pp. 2175–2187, 1999.

[115] K. S. Lee and J. H. Lee, “Convergence of constrained model-based predictive control for batch processes,” *IEEE Transactions on Automatic Control*, vol. 45, no. 10, pp. 1928–1932, 2000.

[116] X. Liu and X. Kong, “Nonlinear fuzzy model predictive iterative learning control for drum-type boiler–turbine system,” *Journal of Process Control*, vol. 23, no. 8, pp. 1023–1040, 2013.

[117] E. Bøhn, S. Gros, S. Moe, and T. A. Johansen, “Reinforcement learning of the prediction horizon in model predictive control,” *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 314–320, 2021.

[118] J. Arroyo, C. Manna, F. Spiessens, and L. Helsen, “Reinforced model predictive control (rl-mpc) for building energy management,” *Applied Energy*, vol. 309, p. 118346, 2022.

[119] A. B. Martinsen, A. M. Lekkas, and S. Gros, “Combining system identification with reinforcement learning-based mpc,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 8130–8135, 2020.

[120] ——, “Reinforcement learning-based nmpc for tracking control of asvs: Theory and experiments,” *Control Engineering Practice*, vol. 120, p. 105024, 2022.

[121] X. Xu, H. Chen, C. Lian, and D. Li, “Learning-based predictive control for discrete-time nonlinear systems with stochastic disturbances,” *IEEE transactions on neural networks and learning systems*, vol. 29, no. 12, pp. 6202–6213, 2018.

[122] M. Zanon, S. Gros, and A. Bemporad, “Practical reinforcement learning of stabilizing economic mpc,” in *European Control Conference (ECC)*, 2019, pp. 2258–2263.

[123] M. Mehndiratta, E. Camci, and E. Kayacan, “Automated tuning of nonlinear model predictive controller by reinforcement learning,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3016–3021.

[124] K. M. Cabral, S. R. B. dos Santos, S. N. Givigi, and C. L. Nascimento, “Design of model predictive control via learning automata for a single uav load transportation,” in *Annual IEEE International Systems Conference (SysCon)*, 2017, pp. 1–7.

[125] X. Xu, H. Xie, and J. Shi, “Iterative learning control (ilc) guided reinforcement learning control (rlc) scheme for batch processes,” in *Data Driven Control and Learning Systems Conference (DDCLS)*, 2020, pp. 241–246.

[126] P. Zarchan, *Tactical and strategic missile guidance*. American Institute of Aeronautics and Astronautics, Inc., 2012.

[127] J. A. Rossiter, *Model-based predictive control: a practical approach*. CRC press, 2003.

[128] B. D. Anderson and J. B. Moore, *Optimal filtering*. Courier Corporation, 2012.

[129] V. K. Madyastha, *Adaptive estimation for control of uncertain nonlinear systems with applications to target tracking*. Georgia Institute of Technology, 2005.

[130] S. Pan, H. Su, J. Chu, and H. Wang, “Applying a novel extended kalman filter to missile–target interception with apn guidance law: A benchmark case study,” *CEP*, vol. 18, no. 2, pp. 159–167, 2010.

[131] M. Hovd and R. R. Bitmead, “Interaction between control and state estimation in nonlinear mpc,” *IFAC Proceedings Volumes*, vol. 37, no. 9, pp. 119–124, 2004.

[132] J. Li, Z. Ning, S. He, C.-H. Lee, and S. Zhao, “Three-dimensional bearing-only target following via observability-enhanced helical guidance,” *IEEE Transactions on Robotics*, 2022.

[133] A. Alessandretti, A. P. Aguiar, and C. N. Jones, “Virtualarena: an object-oriented matlab toolkit for control system design and simulation,” in *International Conference on Unmanned Aircraft Systems, Miami, USA*, 2017, pp. 1508–1515.

[134] G. M. Siouris, *Missile guidance and control systems*. Springer Science & Business Media, 2004.

[135] I. Weintraub, E. Garcia, and M. Pachter, “Optimal guidance strategy for the defense of a non-manoeuvrable target in 3-dimensions,” *IET Control Theory & Applications*, vol. 14, no. 11, pp. 1531–1538, 2020.

[136] E. Garcia, I. Weintraub, D. W. Casbeer, and M. Pachter, “Optimal strategies for the game of protecting a plane in 3-d,” in *American Control Conference (ACC)*, 2022, pp. 102–107.

[137] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi – A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[138] R. Sharma, R. W. Beard, C. N. Taylor, and S. Quebe, “Graph-based observability analysis of bearing-only cooperative localization,” *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 522–529, 2011.

[139] R. Sharma, “Observability based control for cooperative localization,” in *International Conference on Unmanned Aircraft Systems*, 2014, pp. 134–139.

[140] C. V. Rao, J. B. Rawlings, and D. Q. Mayne, “Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations,” *IEEE transactions on automatic control*, vol. 48, no. 2, pp. 246–258, 2003.

[141] A. Alessandri, M. Baglietto, T. Parisini, and R. Zoppoli, “A neural state estimator with bounded errors for nonlinear systems,” *IEEE Transactions on Automatic Control*, vol. 44, no. 11, pp. 2028–2042, 1999.

[142] A. Alessandri, M. Baglietto, and G. Battistelli, "Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes," *Automatica*, vol. 44, no. 7, pp. 1753–1765, 2008.

[143] A. Alessandri, M. Baglietto, G. Battistelli, and V. Zavala, "Advances in moving horizon estimation for nonlinear systems," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5681–5688.

[144] D. Pati. (2016) Fisher information. [Online]. Available: https://ani.stat.fsu.edu/~debdeep/Fisher.pdf

[145] P. Zegers, "Fisher information properties," *Entropy*, vol. 17, no. 7, pp. 4918–4939, 2015.

[146] J. A. Rossiter, *Model-based predictive control: a practical approach*. CRC press, 2003.

[147] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[148] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[149] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.