

THÈSE

pour obtenir le grade de

**DOCTEUR DE LA COMMUNAUTE
UNIVERSITE GRENOBLE ALPES**

Spécialité : **Automatique-Productique**

Arrêté ministériel : 25 Mai 2016

Présentée par

Mukhtar SANI

Thèse dirigée par **Ahmad HABLY** et
codirigée par **Bogdan ROBU**

préparée au sein du **GIPSA-Lab**
dans **Electronique, Electrotechnique, Automatique,
Traitement du Signal (EEATS)**

**Real-time Control Of Mobile Robots Using Model
Predictive Control And Game-theoretic Approaches**

Thèse soutenue publiquement le 4th **March 2022**,
devant le jury composé de:

M. Didier GEORGES

Professeur, Grenoble-INP, Université Grenoble Alpes, Président

Mme. Ouidad LABBANI-IGBIDA

Professeur, Université de Limoges, Rapporteur

M. Pedro CASTILLO

Directeur de Recherche CNRS, UTC, Rapporteur

M. Ahmed CHEMORI

Recherche CNRS, LIRMM, Université de Montpellier, Examineur

M. Florin STOICAN

Professeur, University Politehnica of Bucharest, Examineur

M. Ahmad HABLY

Maitre de conférence, Grenoble-INP, UGA, Directeur de thèse

M. Bogdan ROBU

Professeur Associe, Université Grenoble Alpes, Co-directeur de thèse
(invité)

M. Maher El Rafei

Professeur Associe, Lebanese University (invité)



Dedicated to my mother,
Zainab Nuhu Shanono.

Acknowledgements

First of all, I would like to express my sincere appreciation to my supervisors Ahmad and Bogdan firstly for accepting my candidature to pursue my doctoral research under their tutelage and secondly for their support and encouragement in directing my thesis. Also, I want to extend my appreciation to the jury members of my thesis for reading and assessing the manuscript and for their interesting comments.

Secondly, I must sincerely thank the management of Gipsa-lab for their administrative support and services throughout my stay at the prestigious laboratory. The kindness, humility, care and help given to me will forever remain in my heart. Furthermore, I want to appreciate the Gipsa-Doc, COPERNIC team and the MODUS team for providing platforms for integrating all members of the lab socially, academically as well as administratively.

Thirdly, I would like to acknowledge the Kano state government of Nigeria and the French embassy for the scholarship. I have to mention the wonderful job of Campus France for managing the bilateral agreements.

Finally, I would also like to acknowledge the support of my wife, Zainab Shehu Aliyu, my family and friends for their consistent encouragement and prayers to me. I consider these as one of the most important components of my success. I would not conclude without praising and thanking the almighty God for the gift of life, health, strength and the opportunity to pursue my Masters and PhD in one of the best country and university in the world, Alhamdulillah.

Thesis Abstracts

Autonomous control and navigation of mobile robots received a lot of attention due to the ability of robots to carry out sophisticated tasks in a complex environment with a high level of precision and efficiency. Classical control problems related to mobile robots involved go-to-goal, object tracking, and path following consist a target with pre-defined behavior. As such, the control design does not take into account the future behavior of the target. In surveillance, interception, pursuit-evasion problems, the future behavior of the target must be taken into consideration. These problems where the player (control system) engages an adversary are best tackled using game theory which provides the best strategy for winning. However, game-theoretic algorithms required a lot of information on the opponent to take into account the optimal strategy of the opponent, which is the worst-case scenario from the perspective of the player. This information requirement often restricts the application of game theory on mobile robots. Also, majority of the works found in the literature proposed offline solutions which are applicable to holonomic systems only. This PhD thesis proposed three different solutions to non-cooperative game problems based on the opponent's information available to each player. The proposed solutions are online in nature with the ability to incorporate obstacles avoidance. Also, the controllers designed are applied on nonholonomic mobile robots first in a simulation then validated experimentally in a similar environment. In the first part of the work, the point-stabilization problem in a complex environment was handled using Nonlinear Model Predictive Control (NMPC) with Static and dynamic obstacles avoidance which revolves around the target position. Secondly, the problem was modified to involve a moving target that has a conflicting objective to form a pursuit-evasion game. The problem was solved using Nonlinear Model Predictive Control where two stabilizing approaches are compared. The NMPC method works such that only the current states of the opponent are known to each player. Game-theoretic algorithms are then proposed to solve the same problem. The first method requires all the information of the opponent while the other requires only the current position of the opponent. The methods are compared in terms of capture time, computation time, ability to incorporate obstacle avoidance, and robustness to noise and disturbance. A new problem that lies at the intersection between the point stabilization and pursuit-evasion problem was formulated and solved using game-theoretic model predictive control. The problem is called the Differential Game of Target Defense (DGTD) which involves interception of a moving object before reaching a static target. Finally, all the proposed controllers are validated experimentally using two mobile robots and the Motion Capture platform of the laboratory.

Keywords: Nonlinear Model Predictive Control, Game-theory, Autonomous systems, Nonholonomic mobile robots, Obstacles avoidance, Real-time experimental validation.

Résumé de Thèse(French)

Le contrôle et la navigation autonomes des robots mobiles ont reçu beaucoup d'attention en raison de la capacité des robots à effectuer des tâches sophistiquées dans un environnement complexe avec un haut niveau de précision et d'efficacité. Les problèmes de commande classiques liés aux robots mobiles concernent la poursuite d'un objectif, le suivi d'un objet et le suivi d'une trajectoire consistant en une cible au comportement prédéfini. En tant que telle, la conception du contrôle ne tient pas compte du comportement futur de la cible. Dans les problèmes de surveillance, d'interception, de poursuite-évasion, le comportement futur de la cible doit être pris en considération. Ces problèmes où le joueur (système de contrôle) engage un adversaire sont mieux abordés en utilisant la théorie des jeux qui fournit la meilleure stratégie pour gagner. Cependant, les algorithmes de la théorie des jeux nécessitent beaucoup d'informations sur l'adversaire pour prendre en compte la stratégie optimale de l'adversaire, qui est le pire scénario du point de vue du joueur. Cette exigence d'information limite souvent l'application de la théorie des jeux aux robots mobiles. De plus, la majorité des travaux trouvés dans la littérature ont proposé des solutions hors ligne qui ne sont applicables qu'aux systèmes holonomiques. Cette thèse de doctorat a proposé trois solutions différentes aux problèmes de jeux non-coopératifs basées sur l'information de l'adversaire disponible pour chaque joueur. Les solutions proposées sont en ligne par nature et ont la capacité d'incorporer l'évitement des obstacles. De plus, les contrôleurs conçus sont appliqués sur des robots mobiles nonholonomes d'abord dans une simulation puis validés expérimentalement dans un environnement similaire. Dans la première partie du travail, le problème de la stabilisation du point dans un environnement complexe a été traité à l'aide d'une commande prédictive de modèle non linéaire (NMPC) avec évitement statique et dynamique des obstacles qui tourne autour de la position cible. Ensuite, le problème a été modifié pour impliquer une cible mobile qui a un objectif conflictuel pour former un jeu de poursuite-évasion. Le problème a été résolu à l'aide d'une commande prédictive par modèle non linéaire où deux approches de stabilisation sont comparées. La méthode NMPC fonctionne de telle sorte que seuls les états actuels de l'adversaire sont connus de chaque joueur. Des algorithmes de la théorie des jeux sont ensuite proposés pour résoudre le même problème. La première méthode requiert toutes les informations de l'adversaire tandis que l'autre ne requiert que la position actuelle de l'adversaire. Les méthodes sont comparées en termes de temps de capture, de temps de calcul, de capacité à intégrer l'évitement des obstacles et de robustesse au bruit et aux perturbations. Un nouveau problème qui se situe à l'intersection entre le problème de stabilisation ponctuelle et le problème de poursuite-évasion a été formulé et résolu à l'aide de la commande prédictive de modèle en théorie des jeux. Le problème est appelé le jeu différentiel de défense de cible (DGTD) qui implique l'interception d'un objet mobile avant d'atteindre une cible statique. Enfin, tous les contrôleurs proposés sont validés expérimentalement en utilisant deux robots mobiles et la plateforme de capture de mouvement du laboratoire.

Mots clés: Contrôle prédictif de modèle non linéaire, théorie des jeux, systèmes autonomes, robots mobiles nonholonomes, évitement d'obstacles, expériences physiques.

Contents

Table of Acronyms	xxi
Mathematical Notation	xxiii
1 General Introduction	1
1.1 Motivation	1
1.2 Autonomous systems considered in the thesis	1
1.3 Control problems considered in the thesis	3
1.3.1 Dynamic obstacles avoidance by a single stabilizing mobile robot	4
1.3.2 Pursuit-evasion game problem	4
1.3.3 Differential game of target defense	4
1.4 Proposed solution methods	5
1.5 Thesis outlines	6
2 Dynamic Obstacles Avoidance Using Nonlinear Model Predictive Control	9
2.1 Introduction	9
2.2 Preliminaries	11
2.2.1 Control problems and solution methods for nonholonomic robots	11
2.2.2 Kinematic model of nonholonomic mobile robots	12
2.2.3 Controllability and stability of nonholonomic robots	14
2.2.4 Discretization methods	15
2.2.4.1 Euler method	15
2.2.4.2 Runge-Kutta 4 th order method	16
2.2.5 Obstacle's detection and avoidance	17
2.3 Model predictive control	18

2.3.1	Mathematical formulation of MPC	19
2.4	Control design and implementation	20
2.4.1	Point stabilization with static obstacles avoidance using NMPC	21
2.4.2	Point stabilization with dynamic obstacles avoidance using NMPC	21
2.5	Numerical simulations	22
2.5.1	Discretization method performance comparison	22
2.5.2	Static obstacles avoidance	24
2.5.3	Dynamic obstacles avoidance	26
2.6	Conclusion	32
3	Pursuit-Evasion Games Using Nonlinear Model Predictive Control	33
3.1	Overview of pursuit-evasion games	33
3.1.1	Two player pursuit-evasion games	34
3.1.2	Multi-player pursuit-evasion	35
3.2	Problem statement	36
3.3	Nonlinear model predictive control	37
3.3.1	Stability of NMPC	39
3.3.1.1	Stabilizing terminal cost	39
3.3.1.2	Stability of NMPC without stabilizing constraints or cost	40
3.4	Numerical simulations	41
3.4.1	PEG with stabilizing terminal cost function	42
3.4.2	PEG with monotonically increasing weight profile	43
3.4.3	Comparison of the performance of the two stabilizing methods	43
3.5	Conclusion	46
4	Pursuit-Evasion Games Using Game-theoretic Approaches	47
4.1	Overview of game-theory	47

4.1.1	Elements of game theory	48
4.2	Game theory and control	49
4.2.1	Cooperative games	49
4.2.2	Non-cooperative games	50
4.2.2.1	Homicidal chauffeur problem	50
4.2.2.2	The two car differential games	51
4.2.2.3	Pursuit-evasion game in a constrained environment	51
4.2.2.4	Pursuit-evasion with incomplete information	52
4.2.2.5	Aerial pursuit-evasion problems	52
4.2.2.6	Other pursuit evasion games with two players	53
4.3	Game-theoretic approaches for solving pursuit-evasion problems	53
4.3.1	Game-theoretic algorithm with complete information (GTA-1)	54
4.3.1.1	Pursuer's game-theoretic algorithm (GTA-1)	54
4.3.1.2	Evader's game-theoretic algorithm (GTA-1)	55
4.3.2	Improved game-theoretic algorithm with complete information (GTA-2)	56
4.3.2.1	Pursuer's game-theoretic algorithm (GTA-2)	56
4.3.2.2	Evader's game-theoretic algorithm (GTA-2)	57
4.3.3	Game-theoretic algorithm with incomplete information (GTA-3)	57
4.3.3.1	Prediction of opponent's heading angle	58
4.3.3.2	Pursuer's game-theoretic algorithm (GTA-3)	58
4.3.3.3	Evader's game-theoretic algorithm (GTA-3)	59
4.4	Numerical simulations	60
4.4.1	Simulation results of PEG using GTA-1	60
4.4.2	Simulation results of PEG using GTA-2	63
4.4.3	Simulation results of PEG using GTA-3	64
4.5	Conclusion	66

5	Differential Game of Static Target Defense	69
5.1	Introduction	69
5.2	The differential game of static target defense	71
5.3	Game-theoretic model predictive control	72
5.4	Simulation results	73
5.5	Conclusion	77
6	Real-time Experimental Validations	79
6.1	Introduction	79
6.2	Experimental set-up	79
6.3	Control of single robot with obstacles avoidance	80
6.4	Pursuit-evasion games	84
6.4.1	Pursuit-evasion games using nonlinear model predictive controls	84
6.4.2	Game-theoretic method with incomplete information	87
6.5	Differential game of target defense	92
6.5.1	Experimental results of DGSTD in obstacles free environment	92
6.5.2	Experimental results of DGSTD in an environment with obstacle	93
6.6	Conclusion	95
7	Summary and Outlooks	97
7.1	Research summary based on objective I	98
7.2	Research summary based on objective II	98
7.3	Research summary based objective III	99
7.4	Research summary based on objective IV	99
7.5	Research summary based on objective V	100
7.6	Summary of contributions	100
7.6.1	Lists of publications	101

7.7	Future research directions	101
-----	--------------------------------------	-----

List of Figures

1.1	Holonomic Mobile Robots [Lynch and Park 2017]	2
1.2	Nonholonomic Mobile Robots [Lynch and Park 2017]	3
2.1	Schematic diagram of differential drive robot	13
2.2	Runge-Kutta Discretization	16
2.3	Control system architecture	21
2.4	Euler Discretization	23
2.5	Runge-Kutta 4 th Order Discretization	23
2.6	Runge-Kutta 4 th Order Discretization Method, N=5	24
2.7	Runge-Kutta 4 th Order Discretization Method, N=10	25
2.8	Runge-Kutta 4 th Order Discretization Method, N=20	25
2.9	Computation time	26
2.10	Simulation: Dynamic Obstacles Avoidance	27
2.11	Dynamic Obstacles Avoidance, Circular movement, N =10	29
2.12	Dynamic Obstacles Avoidance, Circular movement, N =20	30
2.13	Dynamic Obstacles Avoidance, Circular movement, N =30	31
2.14	Computation time	32
3.1	Penalizing the orientation angle in PEG	41
3.2	PEG with stabilizing terminal cost	42
3.3	PEG with stabilizing weight	43
3.4	Computation time comparison	44
3.5	Effect of noise of PEG using NMPC	45
3.6	PEG with obstacles avoidance using NMPC	46

4.1	Comparing agility effect in pursuit-evasion game	60
4.2	PEG with GTA-1 and NMPC in an obstacles free environment, the pursuer marked 'P' is green while the evader marked 'E' is red	61
4.3	PEG with GTA-1 and NMPC in the presence of obstacles, the pursuer marked 'P' is green while the evader marked 'E' is red	62
4.4	Comparing the performance of GTA-2 and NMPC in the absence of obstacles, the pursuer marked 'P' is green while the evader marked 'E' is red	63
4.5	Comparing the performance of GTA-2 and NMPC in the presence of obstacles, the pursuer marked 'P' is green while the evader marked 'E' is red and the blue circles are the obstacles	64
4.6	Comapring the performance of GTA-3 and NMPC	65
4.7	PEG with noisy measurements (noise power = 0.001)	66
4.8	PEG with noisy measurements (noise power = 0.01)	66
5.1	Attacker-defender-taget game using GT-NMPC, $\alpha = 0.10$	74
5.2	Attacker-defender-taget game using GT-NMPC, $\alpha = 0.25$	74
5.3	Attacker-defender-taget game using GT-NMPC, $\alpha = 0.5$	75
5.4	Attacker-defender-taget game using GT-NMPC, $\alpha = 0.75$	75
5.5	Attacker-defender-taget game using GT-NMPC, $\alpha = 0.90$	75
5.6	Differential game of static target defense using NMPC	76
5.7	Differential game of static target defense with obstacles avoidance using GTMPC, $\alpha = 0.75$	76
5.8	Differential game of static target defense with obstacles avoidance using NMPC	77
6.1	Robots, communication and motion capture systems.	80
6.2	Point stabilization using NMPC in an obstacles free-environment	81
6.3	Static obstacles avoidance using NMPC	82
6.4	Dynamic obstacles avoidance using NMPC, $N = 10$	82
6.5	Dynamic obstacles avoidance using NMPC, $N = 20$	83
6.6	Dynamic obstacles avoidance using NMPC, $N = 30$	83

6.7	PEG using NMPC in an obstacles free environment	85
6.8	PEG using NMPC in the presence of obstacles	86
6.9	PEG with pursuer using LI-NMPC, evader using NMPC in a free-bounded space	88
6.10	PEG with pursuer using LI-NMPC, evader using NMPC in an environment containing obstacles	89
6.11	PEG with both players using LI-NMPC in a free-bounded space	90
6.12	PEG with both players using LI-NMPC in an environment containing obstacles	91
6.13	Differential game of static target defense using GT-NMPC	93
6.14	Differential game of static target defense using NMPC	94
6.15	Differential game of static target defense using GTMPC	95
6.16	Differential game of static target defense using NMPC	96

List of Tables

4.1	Prisoner's Dilemma	48
4.2	Comparative analysis	67

Table of Acronyms

MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control
LMPC	Linear Model Predictive Control
MIMO	Multi Inputs Multi Outputs
SISO	Single Input Single Output
LTI	Linear Time Invariant
LPV	Linear Parameter Varying
DOF	degree of freedom
GTA	Game-theoretic Algorithm
RHC	Receding Horizon Control
DGSTD	Diferential Game of Static Target Defense
GTMPC	Game-theoretic Model Predictive Control
LIMPC	Limited Information Model Predictive Control
RK4	Runge-Kutta 4^{th} order
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
LIDAR	Light Detection and Ranging
OCP	Optimal Control Problem
APF	Artificial Potential Field
PEG	Pursuit-evasion Game
PEDG	Pursuit-evasion Differential Game
UDP	User Datagram Protocol
IRC	Internet Relay Chat

Mathematical Notation

\mathbb{R}	Real numbers
\mathbf{x}	A vector of state variables
\mathbf{u}	A vector of control variables
\mathbf{x}_p	A vector of pursuer state variables
\mathbf{x}_e	A vector of evader state variables
\mathbf{x}_A	A vector of attacker state variables
\mathbf{x}_D	A vector of defender state variables
$\mathbf{f}(\cdot)$	A function
Q	Weighting matrix of the state variable
R	Weighting matrix of the control variables
N	Prediction Horizon
J	Cont function
t	Time variable
k	An open-loop time index
T_s	Sampling time
\mathbf{x}_0	Initial condition
\mathbf{x}_r	Reference state
\mathcal{G}	Game tuple
\mathcal{U}	Strategy sets
$\ \mathbf{x}\ $	Euclidean norm of a vector
$\ \mathbf{x}\ _Q^2$	A short notation for the matrix $\mathbf{x}^T Q \mathbf{x}$
$\text{diag}(a,b,\dots,n)$	Diagonal matrix with entries a,b,\dots,n

General Introduction

This chapter presents the overall picture of this thesis by highlighting the motivations, control challenges addressed, solution methodologies employed, and the contribution of the thesis.

1.1 Motivation

As our world becomes more robotic as a result of human curiosity and the ability of machines to perform sophisticated tasks in industries, military, and civilian applications, autonomous control of mobile robots is an area whose significance cannot be overstated. Autonomous systems can be programmed to perform a wide range of tasks in a complex environment where humans are at risk. They include exploration, interception of dangerous objects, surveillance, and many others. The inherent efficiency of machines and robots allows for a high level of accuracy and dependability for mission execution. Autonomy of a robot or any moving object cannot be completed without the ability to avoid obstacles and some adversaries. As a result, it is essential to investigate control techniques that can autonomously navigate mobile robots to carry out complex missions. These control techniques should be capable of driving the system away from any obstacles it may encounter.

Furthermore, the recent commencement of Roborace, an international competition where autonomous vehicles are programmed to engage in autonomous racing. All the participating vehicles are electrically powered with the same chassis and powertrain, but each team must develop its real-time computing algorithms and artificial intelligence technologies. Due to the competitive nature of this game, as each team strives to win the race against their co-participants, the computing algorithms must take into account the strategies and actions of the opponent. This interesting competition motivated my research interest to delve into game theory with a strong desire to integrate it into the control of nonholonomic mobile robots in a complex environment.

1.2 Autonomous systems considered in the thesis

Autonomous robots are grouped into numerous categories, including wheeled and legged; aerial, ground, and underwater; holonomic (omnidirectional) and nonholonomic robots; homogeneous multi-robot systems and heterogeneous; and large scale, macro, and micro robots;



Figure 1.1: Holonomic Mobile Robots [Lynch and Park 2017]

see [Jualin 2015; Lynch and Park 2017] for more details. While a lot of works on holonomic mobile robots have been presented in the literature, the scope of this thesis includes nonholonomic wheeled mobile robots. To better understand the difference between the two, here is a brief explanation.

Holonomic mobile robots are robots that are capable of moving in any direction without changing their orientation. Figure 1.1 shows two holonomic mobile robots, one with three omniwheels and the other with four mecanum wheels (often called Swedish wheels). An omniwheel is a typical wheel augmented with rolls on its outer circumference. These rollers spin freely about axes in the plane of the wheel and tangential to the wheel’s outer circumference and they allow sideways sliding while the wheel drives forward or backward without slipping in that direction. Mecanum wheels are similar, except that the spin axes of the circumferential rollers are not in the plane of the wheel. The sideways sliding allowed by omniwheels and mecanum wheels ensures that there are no velocity constraints on the robot’s chassis.

Nonholonomic mobile robots, on the other hand, have limited degrees of freedom, as they have to change their orientation in order to move in some directions because the dimension of its admissible velocity space is smaller than the dimension of the configuration space. They can be described as a class of mobile robots that possess differential constraints (also called nonholonomic or velocity constraints). Nonholonomic wheeled mobile robots employ conventional wheels that do not allow sideways sliding (drift) i.e the robot can only move in the direction normal to the axis of the driving wheels. It is pertinent to note that some underwater and aerial robots are nonholonomic.

There are four different types of nonholonomic wheeled mobile robots as depicted in figure 1.2. The unicycle type in figure 1.2a has two controlled wheels at the left and right side as well as one omnidirectional freewheel at the front side of the robot. The bicycle-type in figure 1.2b has one controlled wheel at the back and one steerable wheel at the front. In figure 1.2c, a tricycle type of nonholonomic wheeled mobile robot was depicted. The difference between the unicycle and the tricycle is that the tricycle has one steerable wheel at the front instead of an omnidirectional wheel as in the case of the unicycle. Lastly, the car-like type presented in figure 1.2d possessed two controlled wheels at the back and two steerable wheels at the front.

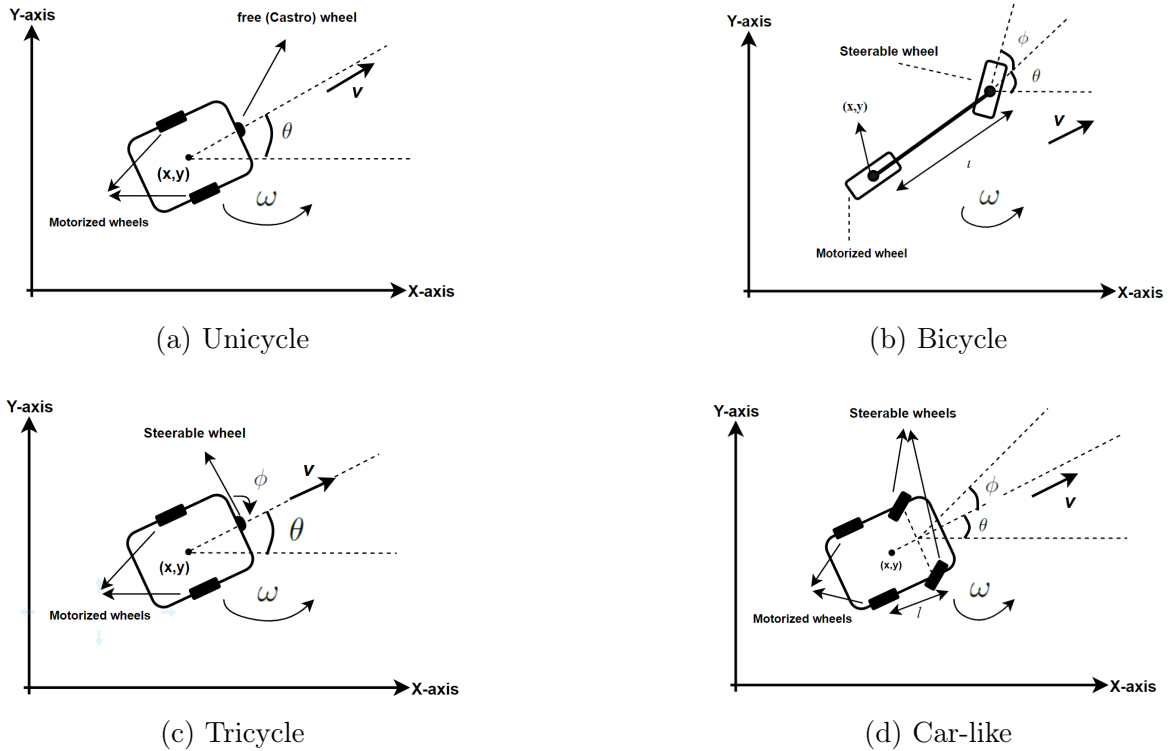


Figure 1.2: Nonholonomic Mobile Robots [Lynch and Park 2017]

The interest in employing nonholonomic mobile robots in this thesis arises due to the theoretical and practical motivations. From the theoretical point of view, the nonholonomic robots have limited number of control inputs, in other words, they are under-actuated. This feature imposes a controllability issue that is directly linked to the point-stabilization task which cannot be accomplished with a pure feedback control [Mehrez 2017]. Brockett’s condition implies that the linearised nonholonomic model is not stabilizable [Brockett 1983b]. With fewer control actuators, they are found in a wide range of applications such as surveillance, planetary exploration, transportation, human-machine interface for people with disabilities, military target tracking/attack, and so on.

Although the control algorithms developed in this thesis are applied on the wheeled non-holonomic mobile robots, the algorithms can be applied to aerial systems with slight modifications.

1.3 Control problems considered in the thesis

The basic tasks in the robotic application include but are not limited to mapping, localization, planning, and control [Morin and Samson 2008]. In most cases, mapping and localization take place at the same time to localize themselves once the mapping of the given environment

is known, thus a widely known technique is called simultaneous mapping and localization (SLAM). After localization, the next task is the trajectory/path planning or control where an algorithms are developed to control the robots. In this thesis, three control problems are addressed sequentially. They are dynamic obstacles avoidance by a single stabilizing mobile robot, pursuit-evasion game, and the differential game of static target defense. These problems are formulated and solved in both numerical simulations and real-time experiments.

1.3.1 Dynamic obstacles avoidance by a single stabilizing mobile robot

The first problem tackled in this thesis is the dynamic obstacles avoidance by a single stabilizing nonholonomic mobile robot. The goal is to develop a control algorithm that could steer a mobile robot autonomously from a random starting position to a final position and orientation while avoiding both static and dynamic obstacles whose speeds are unknown. Control problem related to reference point stabilization was integrated with dynamic obstacle avoidance. When the obstacle occurs along the path of the moving obstacle, the proposed control algorithm autonomously navigate the robot away for the obstacle by calculating the minimal distance from which the avoidance maneuver starts. However, unlike in the state-of-the-art techniques, the speed of the dynamic obstacle is unknown to the controller.

1.3.2 Pursuit-evasion game problem

The second control problem addressed in this thesis is the so-called pursuit-evasion game problem. In this problem, there are two players with conflicting objectives such that the pursuer aimed to capture the evader while the evader tries to dodge. To capture the evader, the pursuer's controller has the task of minimizing the relative distance between the two-player. Defending the physical property and the strategy employed, the capture time is the main parameter for evaluating the performance of the pursuer's controller. On the other hand, the evader's controller maximizes the relative distance between the players. In the worst situation of guaranteed capture, the evader may tend to maximize the capture time instead.

The success or the winning possibility of each player in the context of the pursuit-evasion game depends on the level of information each player has on the opponent. This level of information is critical to the choice of strategy by each player which eventually leads to winning or losing the game. We have investigated different variants of pursuit-evasion problems presented by Isaacs in his book called *The differential games* [Isaacs 1956] based on complete and incomplete information on the opponents.

1.3.3 Differential game of target defense

The differential game of target defense problem is considered to strike a balance between the point-stabilization and classical pursuit-evasion problems. Three players exist in this problem

namely, attacker, defender, and the target such that the attacker aims to reach the target and avoid the defender concurrently. The defender aims to capture the attacker as well as protect the target. Static target is considered here so as to mimic the game of guarding a target proposed by Issac. The defender must strive alone to prevent the attacker from reaching the target without any cooperative support from the target. This problem is also integrated with obstacles avoidance.

1.4 Proposed solution methods

In this thesis, the first control problem considered is the dynamic obstacles avoidance by a single nonholonomic mobile robot moving from its initial position to the goal position. A novel method was employed based on the nonlinear model predictive control technique to solve the problem. The novelty in the proposed controller was the ability to simultaneously navigate the robot from its initial pose to the final pose and to steer the robot away from dynamic obstacles without the need to measure the speed of the obstacles which require more sensors. The method also does not require any additional computation which could increase the computation time. The stability of the proposed controller was achieved using the classical terminal cost inclusion.

The problem was then reformulated to involve a moving target that is controlled by an independent conflicting controller. The pursuit-evasion problem was solved by two different nonlinear model predictive controllers formulated in a converse way. The pursuer's controller solves minimization cost function while the evader's controller solves maximization cost function. The stability of these controllers was achieved using two methods; the first one includes terminal cost while the second one includes a monotonically increasing weight profile in the cost function, excluding the terminal cost. The controllers worked based on the availability of the deterministic information on the state of each system. Therefore, this approach does not require the need to use state observers.

Next, the game-theoretic approach was then developed formulated to improve the intelligence and to provide strategic advantage to the players. Firstly, a game-theoretic technique was proposed under the assumption of knowledge of complete information on the opponent was proposed. The method that tackled the pursuit-evasion game worked in such a way that each payer firstly predicts the optimal future move of the opponent and then computes its Nash equilibrium strategy. Then, the method was improved to predict more optimal steps of the opponent in advance under the same assumption. Secondly, another game-theoretic method was proposed which considers incomplete information on the opponent. The method was termed limited information model predictive control dealt with the burden of high computation time and the requirement of additional sensors. These proposed game-theoretic methods have the capability to avoid static and dynamic obstacles.

Finally, a novel method termed game-theoretic nonlinear model predictive control was proposed to deal with the game of guarding a static target with a single defending agent when both the attacker and the defender have the same speed and are equidistant from the target.

A parameter is designed to provide a trade-off between pursuing and the defending objectives of the defender. The proposed method was stabilized by adding a terminal cost and can avoid obstacles.

1.5 Thesis outlines

Three different control problems for nonholonomic mobile robots are considered in this thesis. The general introduction was given in chapter 1 while the concluding chapter was given in chapter 7. The contributing chapters of the thesis are 2, 3, 4, 5 and 6. The outline of the thesis are as follows:

- Chapter 1: General introduction was given in this chapter. It comprises the motivation of the thesis, the autonomous systems considered, the control problems and their proposed solution methods as well as the outline of the thesis.
- Chapter 2: In this chapter, discrete-time formulation of NMPC was presented to stabilize a single nonholonomic mobile robot to the desired posture while avoiding static and dynamic obstacles. Several numerical experiments were carried out to verify the proposed controller. The content of this chapter was published in [Sani, Robu, and Hably 2021a]
 - Mukhtar Sani, Bogdan Robu and Ahmad Hably, "Dynamic Obstacles Avoidance Using Nonlinear Model Predictive Control" in the proceedings of the 47th Annual Conference of the IEEE Industrial Electronic Society (IECON'2021), Toronto, Canada (2021).
- Chapter 3: This chapter builds on the previous chapter by employing a moving target to form Pursuit-evasion Games. The problem was solved using NMPC.
- Chapter 4: This chapter builds on the previous chapter by employing a game-theoretic approaches to deal with pursuit-evasion game based on the available information of the opponent. The content of this chapter appears in the following publications [Sani, Robu, and Hably 2020; Sani, Robu, and Hably 2021c; Sani, Robu, and Hably 2021b]
 - Mukhtar Sani, Bogdan Robu and Ahmad Hably, "Pursuit-evasion games for Non-holonomic mobile Robots with Obstacles Avoidance Using Nonlinear Model Predictive Control" in the proceedings of the 28th Mediterranean Conference on Control and Automation (MED'2020), St Raphael, France (2020).
 - Mukhtar Sani, Bogdan Robu and Ahmad Hably, "Pursuit-evasion Games Based on Game-theoretic and Model Predictive Control Algorithms" in the proceedings of the 5th International Conference on Control, Automation and Diagnosis (ICCAD'2021), Grenoble, France (2021).
 - Mukhtar Sani, Bogdan Robu and Ahmad Hably, "Limited Information Model Predictive Control for Pursuit-evasion Games" in the proceedings of the 60th IEEE Conference on Decision and Control (CDC'2021), Texas, USA (2021).

-
- Chapter 5: This chapter presents a game of guarding target as a differential game of target defense. We used game-theoretic model predictive control for the problem in the presence and absence of obstacles. The content of this chapter was submitted to the Asian Journal of Control.
 - Chapter 6: In this chapter, experimental results were presented. The experimental setup comprising of the camera systems for localization, and the communication systems are explained in detail.
 - Chapter 7: This chapter summarizes and analyses the findings of the thesis and then draws the related conclusions. In addition, future perspectives are outlined.

Dynamic Obstacles Avoidance Using Nonlinear Model Predictive Control

This chapter deals with dynamic and static obstacle avoidance by a nonholonomic wheeled mobile robots moving on a plane. The control algorithms are based on real-time nonlinear model predictive control technique where an optimization problem is solved at every time instant. Control problem related to reference point stabilization is integrated with dynamic and static obstacles avoidance. When the obstacle occurs along the path of the moving obstacle, the controller autonomously navigate the robot away for the obstacle by calculating the minimal distance from which the avoidance maneuver starts. However, unlike in the state-of-the-art techniques, the speed of the dynamic obstacle is unknown to the controller. Measurements of the full states of the robot and the obstacles are deterministic, thus exclude the need of an observer. Several simulation results are presented to show the performance of the proposed method.

2.1 Introduction

Due to the rapid technological development in the twenty-first century, a wide range of applications for mobile robots in various sectors have attracted great interest from researchers. Unmanned vehicles comprising of aerial, ground, and underwater vehicles have been utilized for many applications such as mine clearance [Portugal, Marques, and Armada 2014], patrolling mission [Portugal and Rocha 2011], surveillance [Acevedo et al. 2013], search & rescue mission [Bernard et al. 2011] and educational research purposes [Arai, Pagello, and Parker 2002]. However, several unmanned vehicles in use today are not fully autonomous because of the presence of human-in-the-loop. The human's natural intelligence and skills are being utilized to pilot the robot's navigation. The problem of human-in-the-loop systems is the reliability of the communication system between the robots and the human in the base station. Problems such as delay, bandwidth limitation, jamming, and loss of signals are critical to the reliability of communication systems. To remove the human from the loop, the systems should be converted to partial or fully autonomous to solve or minimize the effects of the communication issues and also to automatically avoid obstacles and hazardous conditions. A suitable automatic control algorithm would then be employed to pilot the robot.

Several approaches for autonomous control of mobile robots are presented in the literature.

They include dynamic feedback linearization [Oriolo, De Luca, and Vendittelli 2002], Lyapunov control [Indiveri 1999; Castillo, Dzul, and Lozano 2004], smooth time-varying control [M'Closkey and Murray 1997] and piece-wise continuous feedback control [Wit and Sordalen 1991]. However, these approaches do not incorporate constraints on the mobile robots states which are pertinent for obstacles avoidance.

Therefore, the natural candidate for the control of autonomous systems is the Model Predictive Control due to its flexibility, its ability to handle both soft and hard constraints as well as its ability to compute optimal control inputs for nonlinear systems. Several works such as [Gu and Hu 2005; Kuhne, Lages, and Silva 2005; Worthmann et al. 2016] have dealt with stabilization problems for nonholonomic mobile robots. Others such as [Gu and Hu 2006; Lim et al. 2008a; Faulwasser, Kern, and Findeisen 2009; Raffo et al. 2009; Taktak-Meziou et al. 2014] solve trajectory tracking and path following problems using MPC. Many others such as [Mehrez, Mann, and Gosine 2013; Xie and Fierro 2008; Mehrez, Mann, and Gosine 2015] consider solving point stabilization and tracking problems simultaneously without incorporating obstacles avoidance.

On the other hand, safety has become the eternal theme of autonomous vehicles [Rosolia, De Bruyne, and Alleyne 2017]. Active collision avoidance system has become a research hot-spot in the field of automotive due to its ability to effectively improve traffic safety [Li et al. 2019]. Static obstacles avoidance have been dealt with in [Lim et al. 2008b; Abbas, Milman, and Eklund 2017] for tracking problems, in [Sani, Robu, and Hably 2020] for pursuit-evasion games, in [Garimella et al. 2018] for point-stabilization and in [Rosolia, De Bruyne, and Alleyne 2017; Zhang et al. 2020] for path following problems. Dynamic collision avoidance among multiple mobile robots has been considered in [Mikumo and Ichihara 2017]. The literature in [Castillo-Lopez et al. 2018] dealt with dynamic obstacle avoidance for an Unmanned Aerial Vehicles (UAV) while [Li et al. 2019] presented dynamic trajectory planning and tracking with dynamic obstacles for an Unmanned Ground Vehicle (UGV). However, these works assumed full knowledge of the obstacle speed and size, which were either obtained by measurement or additional computation. While measurement requires embedding an additional sensor, computation increased the computation time. Following these drawbacks, it is therefore interesting to work on another obstacle avoidance method that would not require measuring or predicting the movement of the obstacles.

In this work, an NMPC based dynamic obstacles avoidance algorithm that only needs the instantaneous position of the obstacles for a point-stabilization problem was developed. Different from the work in [Li et al. 2019], the algorithm does not require additional computation for predicting the speed of the obstacles which could lead to higher computation time and wrong prediction in the case of intelligent obstacles. This method incorporates obstacle avoidance as a constraint to be considered while solving the optimal control problem. The second contribution of this chapter is that the proposed approach could be used to simultaneously handle both static and dynamic obstacles avoidance. Finally, the performance of two discretization methods, the 4th order Runge-Kutta and the Euler method are compared.

2.2 Preliminaries

In this section, the control problems for nonholonomic mobile robots as well as the solution methods are discussed, the kinematic model of a nonholonomic mobile robot, the discretization methods and the obstacles avoidance techniques.

2.2.1 Control problems and solution methods for nonholonomic robots

There are three types of feedback control problems for nonholonomic mobile robots namely point stabilization or regulation, trajectory tracking and path following/tracking [Lynch and Park 2017; Mehrez 2017; Faulwasser 2012; Feng 2004]. For the purpose of differentiating these control problems, let's consider the discrete model of a system (2.1) where $\mathbf{x} \in \mathbb{R}^n$ is the state vector of the system, $\mathbf{u} \in \mathbb{R}^m$ is the control inputs of the system, $\mathbf{f} \in \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ is the nonlinear mapping and k is the current sampling time.

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases} \quad (2.1)$$

Point stabilization problem or regulation is required to design a feedback control law of the form $\mathbf{u} = \mu(\mathbf{x}(k))$ such that for a given desired set-point \mathbf{x}_r which is a constant reference, the solution of (2.1) converges to the desired set-point. i.e the controller drives the error ($\mathbf{x}_r - \mathbf{x}(k)$) to zero as time goes to infinity.

$$\lim_{k \rightarrow \infty} \|\mathbf{x}_r - \mathbf{x}(k)\| = 0 \quad (2.2)$$

For trajectory tracking problem, the desired reference set-point is time-varying $\mathbf{x}_r(k)$, therefore the problem is to design a feedback control law $\mathbf{u} = \mu(\mathbf{x}(k))$ such that the solution of (2.1) converges to the desired time-varying set-points. i.e the controller drives the error ($\mathbf{x}_r(k) - \mathbf{x}(k)$) to zero as time goes to infinity.

$$\lim_{k \rightarrow \infty} \|\mathbf{x}_r(k) - \mathbf{x}(k)\| = 0 \quad (2.3)$$

In the case of path tracking or following problem, it is required to follow a geometric path without regard to the time of the motion. This gives the controller more degrees of freedom than the trajectory tracking problem. To reduce the tracking error, we can choose the speed of the reference configuration along the path [Lynch and Park 2017].

It's worth noting that the point stabilization problem is a special class of trajectory tracking problem. According to Brockett's theorem [Brockett 1983a], the linearized model of nonholonomic mobile robots loses stabilizability, thus there is no time-invariant feedback law that is continuous in the state variables that can stabilize a configuration for a nonholonomic mobile robot. For this trajectory tracking and path tracking problems are simpler than the point stabilization problems since there exist continuous time-invariant feedback laws that can stabilize the desired motions.

There are several solution methods to the point stabilization problem of nonholonomic robots in the literature as reported in [Mehrez 2017] and [Feng 2004]. They include dynamic feedback linearization [Oriolo, De Luca, and Vendittelli 2002], Lyapunov control [Indiveri 1999], smooth time-varying control [M'Closkey and Murray 1997] and piece-wise continuous feedback control [Wit and Sordalen 1991]. However, these approaches do not incorporate constraints on the mobile robots states which are pertinent especially for obstacles avoidance. For this, Model Predictive Control approaches have been developed to handle point stabilization problems for nonholonomic mobile robots in [Gu and Hu 2005; Kuhne, Lages, and Silva 2005; Worthmann et al. 2016; Worthmann et al. 2015].

The trajectory tracking problems have been dealt with using many approaches such as dynamic feedback linearization in [Oriolo, De Luca, and Vendittelli 2002], sliding mode control in [Chwa 2004] and backstepping in [Jiang-Dagger and Nijmeijer 1997]. The drawback of these approaches is that tracking can be achieved only by imposing constraints on the reference velocity. An MPC-based approach which is take into consideration such constraints on the reference velocity was proposed to handle trajectory tracking problems for nonholonomic mobile robots in [Gu and Hu 2006; Lim et al. 2008a; Prodan et al. 2013b; Prodan et al. 2013a].

Some control approaches can simultaneously handle point stabilization and trajectory tracking problem. They include feedback linearization method [Oriolo, De Luca, and Vendittelli 2002] and differential kinematic control [Dixon et al. 2000]. However, it was argued by [Mehrez 2017] that these approaches do not provide a single controller architecture that is capable of handling both point stabilization and trajectory tracking problem without the need to switch between two control modes. This issue was resolved using vector field orientation feedback control [Michalek and Kozłowski 2010], even though it doesn't consider actuator saturation constraint, and saturation feedback controller [Lee et al. 2001]. However, the process of selecting a suitable tuning parameter is difficult. Thus, MPC approaches have been proposed to handle the above drawbacks in [Mehrez, Mann, and Gosine 2013; Xie and Fierro 2008; Mehrez, Mann, and Gosine 2015]. Path-following feedback control problems for nonholonomic robots have been pursued in the literature using various approaches such as transverse feedback linearization in [Nielsen, Fulford, and Maggiore 2010], backstepping in [Aguiar, Hespanha, and Kokotović 2008]. However, it is difficult to impose states and inputs constraints using these approaches. Therefore MPC approaches were proposed in [Faulwasser, Kern, and Findeisen 2009; Raffo et al. 2009; Backman, Oksanen, and Visala 2012; Faulwasser 2012].

2.2.2 Kinematic model of nonholonomic mobile robots

A kinematic model of a mobile robot governs how wheel speeds map to robot velocities, while a dynamic model governs how the torques map to robot acceleration. We will focus on the kinematic model and ignore the dynamic model. The kinematic model of the nonholonomic wheeled mobile robot of unicycle type in Figure 2.1 can be represented in (2.4). The details

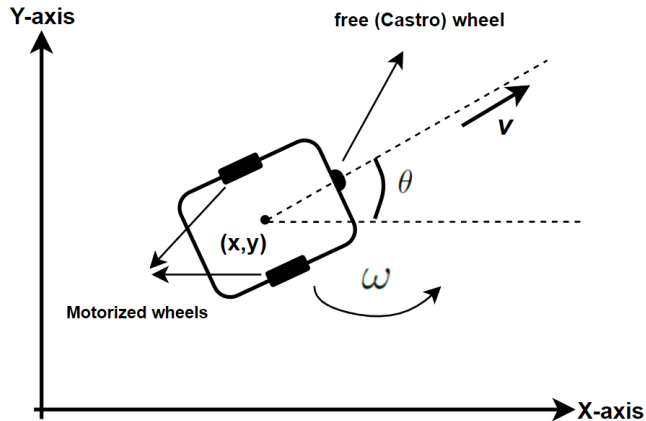


Figure 2.1: Schematic diagram of differential drive robot

can be obtained from [Lynch and Park 2017; Jualin 2015].

$$\begin{cases} \dot{x}(t) &= v(t) \cos \theta(t) \\ \dot{y}(t) &= v(t) \sin \theta(t) \\ \dot{\theta}(t) &= \omega(t) \end{cases} \quad (2.4)$$

where the state variable $\mathbf{x} = [x, y, \theta]^T (m, m, rad)^T$ represent the position of the robot in chassis frame of reference and the orientation of the robot. The control variable $\mathbf{u} = [v, \omega]^T (m/s, rad/s)$, where v and ω stands for the linear speed and the angular speed of the robot, respectively.

The kinematic models are said to be nonholonomic because with slight manipulation we can obtain a differential constraint in (2.5):

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0 \quad (2.5)$$

Since linear approximations are usually regarded as the first step for the analysis and control design of a nonlinear system. Thus if the linearized system is controllable, then the original nonlinear system is at least locally controllable and feedback stabilizable. However, the linearized model is not controllable, because the rank of the controllability matrix is less than the number of its states; See [Feng 2004] for a detailed explanation.

The driftless form of nonholonomic robots (2.6) was investigated for weaker controllability (sufficient) conditions from nonlinear systems theory. Eventually, it was proved using Brockett's theorem [Brockett 1983b] that the system 2.4 is small-time locally controllable [Bloch 2003; Feng 2004]

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} + \omega \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.6)$$

2.2.3 Controllability and stability of nonholonomic robots

Controllability refers to the ability to drive a system from one state to another through the control input. Stability on the other hand is related to the properties of system trajectories around an equilibrium point. An equilibrium point is a vector \mathbf{x}_e such that $f(\mathbf{x}_e) = 0$.

The kinematic model of nonholonomic robot is non-linear of the form $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$, where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^m$. By linearizing the kinematic model around an equilibrium points $\mathbf{x}_e = 0, \mathbf{u}_e = 0, \cos(\theta) = 1$ and $\sin(\theta) = 0$. Therefore, the linearized model (in the form $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$) will be:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2.7)$$

Literarily, A system is said to be Linearly Controllable if the Kalman Rank condition is satisfied. Mathematically,

$$\text{rank}(\mathcal{C}) = \text{rank}([B \ AB \ A^2B \ \dots \ A^{n-1}B]) = n. \quad (2.8)$$

Therefore, the controllability matrix of (2.7), $\mathcal{C} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$

$$\text{rank}(\mathcal{C}) < n. \quad (2.9)$$

Therefore nonholonomic robots are not linearly controllable. Despite that, they may satisfy weaker controllability (sufficient) conditions from non-linear control theory. The Lie bracket of vector fields was usually employed to test whether the nonlinear system is small-time locally controllable.

Let \mathcal{O} be a non empty open subset of \mathbb{R}^n and let $\mathbf{x}_e \in \mathcal{O}$. Assuming some vector fields $f_1, \dots, f_m : \mathcal{O} \rightarrow \mathbb{R}^n$. For a driftless system

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) = \sum_{i=1}^m f_i(\mathbf{x})\mathbf{u}_i, \forall (\mathbf{x}, \mathbf{u}) \in \mathcal{O} \times \mathbb{R}^m \quad (2.10)$$

The control system $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ is small-time locally controllable at $(\mathbf{x}_e, 0) \in \mathbb{R}^n \times \mathbb{R}^m$ if and only if

$$h(\mathbf{x}_e); h \in \text{Lie}\{f_1, \dots, f_m\} = \mathbb{R}^n \quad (2.11)$$

The kinematic model of non-holonomic robot (2.4) can be represented in the Canonical form:

$$\dot{\mathbf{x}} = f_1(\mathbf{x})\mathbf{u}_1 + f_2(\mathbf{x})\mathbf{u}_2$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v \begin{bmatrix} \cos \theta \\ \sin \theta \\ 0 \end{bmatrix} + \omega \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.12)$$

where $f_1(\mathbf{x}) = [\cos \theta, \sin \theta, 0]^T$ and $f_2(\mathbf{x}) = [0, 0, 1]^T$ are the control vector fields. The Lie bracket of f_1 and f_2 can be obtained from the relation:

$$[f_1, f_2](\mathbf{x}) = \frac{\partial f_2}{\partial \mathbf{x}} f_1 - \frac{\partial f_1}{\partial \mathbf{x}} f_2 = \begin{bmatrix} \sin \theta \\ -\cos \theta \\ 0 \end{bmatrix} \quad (2.13)$$

At $\mathbf{x}_e = [0, 0, 0]^T$, $f_1(\mathbf{x}_e) = [1, 0, 0]^T$, $f_2(\mathbf{x}_e) = [0, 0, 1]^T$ and $[f_1, f_2](\mathbf{x}_e) = [0, -1, 0]^T$.

$$\text{rank}\{f_1, f_2, [f_1, f_2](\mathbf{x}_e)\} = 3. \quad (2.14)$$

This means that $h(\mathbf{x}_e) = \mathbb{R}^3$. Therefore the control system (non-holonomic robot) is small-time locally controllable at $(\mathbf{x}_e, 0)$.

However, even though the nonlinear model of a nonholonomic robot is controllable, it doesn't mean it can be stabilized by a smooth time-invariant control policy. Therefore we have to verify its stabilizability.

The problem is state feedback stabilization for a smooth time invariant system concerns about finding a state feedback law of the form $\mathbf{u} = k(\mathbf{x})$, where $k(\mathbf{x})$ is a smooth function of \mathbf{x} such that the closed-loop system $\dot{\mathbf{x}} = f(\mathbf{x}, k) = f(\mathbf{x}^*)$ is asymptotically stable.

Brockett's theorem [Brockett 1983a; Bloch 2003] provides a general condition for smooth feedback stabilization of a nonlinear system. However, Brockett's condition was not satisfied, and thus the linearised model of nonholonomic is not stabilizable.

2.2.4 Discretization methods

The exact discrete time dynamic of nonholonomic mobile robot was derived using direct integration in [Thrun, Burgard, and Fox 2005]. Considering a piecewise control inputs on each interval $[kT_s, (k+1)T_s]$, $k \in \mathcal{N}$, with sampling period of T_s seconds, the exact discrete model is given by:

$$\begin{cases} x_{k+1} &= x_k + \frac{v_k}{\omega_k} (\sin(\theta_k + T_s \omega_k) - \sin \theta_k) \\ y_{k+1} &= y_k + \frac{v_k}{\omega_k} (\cos \theta_k - \cos(\theta_k + T_s \omega_k)) \\ \theta_{k+1} &= \theta_k + T_s (\omega_k) \end{cases} \quad (2.15)$$

Here, the kinematic model of a nonholonomic mobile robot will be discretized using two methods. The performance comparison of these two method would be present in the results section.

2.2.4.1 Euler method

Several works involving discrete NMPC for controlling nonholonomic robots utilized the Euler method also called Euler forward method [Sani, Robu, and Hably 2020; Lim et al. 2008b].

The discrete model form of any system that was discretized using Euler method is given in (2.16).

$$\mathbf{x}_{k+1} = \mathbf{x}_k + T_s f(\mathbf{x}_k) \quad (2.16)$$

Where \mathbf{x}_{k+1} is the approximate next state of the system after one discrete instant, \mathbf{x}_k is the current state, while T_s is the sampling time. Therefore the corresponding discrete model of nonholonomic mobile robot would be given as:

$$\begin{cases} x_{k+1} &= x_k + T_s(v_k \cos \theta_k) \\ y_{k+1} &= y_k + T_s(v_k \sin \theta_k) \\ \theta_{k+1} &= \theta_k + T_s(\omega_k) \end{cases} \quad (2.17)$$

However, the Euler method does not give good approximate solutions of nonlinear ordinary differential equations for a larger sampling time (T_s) because it only uses one slope. It is reasonable to assume that using more estimates of slopes at the mid-point and the end of the interval would result in more accuracy, see [Lin et al. 2010] for more details.

2.2.4.2 Runge-Kutta 4th order method

Runge-Kutta 4th order method (*RK4*) is the most widely known member of the Runge-Kutta family. The *RK4* approximation of $\mathbf{x}(t_{k+1})$ is \mathbf{x}_{k+1} which depends on the current value \mathbf{x}_k and some weighted average of four increments as depicted in figure 2.2. Each increment is a function of the sampling time and an estimated slope specified by a function of the right-hand side of the differential equation. By representing the ordinary differential equation of a system as an initial value problem, we have (2.18).

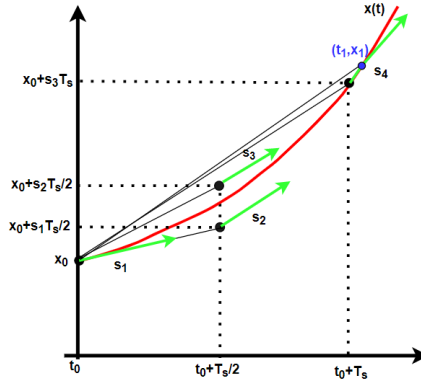


Figure 2.2: Runge-Kutta Discretization

$$\begin{cases} \dot{\mathbf{x}} = f(t, \mathbf{x}) \\ \mathbf{x}(t_0) = \mathbf{x}_0 \end{cases} \quad (2.18)$$

The mathematical representation of RK4 is given as:

$$\mathbf{x}_{k+1} = \frac{1}{6} T_s (s_1 + 2s_2 + 2s_3 + s_4) \quad (2.19)$$

Such that:

$$\begin{cases} s_1 = f(t_k, \mathbf{x}_k) \\ s_2 = f(t_k + \frac{T_s}{2}, \mathbf{x}_k + T_s \frac{s_1}{2}) \\ s_3 = f(t_k + \frac{T_s}{2}, \mathbf{x}_k + T_s \frac{s_2}{2}) \\ s_4 = f(t_k + T_s, \mathbf{x}_k + T_s s_3) \end{cases} \quad (2.20)$$

The first slope s_1 is at the beginning of the interval (this is Euler's method). The slopes s_2 and s_3 are both at the mid-point of the interval while the s_4 is at the end of the interval.

Consequently, the corresponding discrete model of nonholonomic mobile robot would be given as:

$$x_{k+1} = x_k + \frac{1}{6}T_s(s_1 + 2s_2 + 2s_3 + s_4) \quad (2.21)$$

where $s_1 = v_k \cos \theta_k$; $s_2 = f(x_k + \frac{T_s}{2}s_1)$; $s_3 = f(x_k + \frac{T_s}{2}s_2)$; $s_4 = f(x_k + T_s s_3)$.

$$y_{k+1} = y_k + \frac{1}{6}T_s(s_1 + 2s_2 + 2s_3 + s_4) \quad (2.22)$$

where $s_1 = v_k \sin \theta_k$; $s_2 = f(y_k + \frac{T_s}{2}s_1)$; $s_3 = f(y_k + \frac{T_s}{2}s_2)$; $s_4 = f(y_k + T_s s_3)$.

$$\theta_{k+1} = \theta_k + \frac{1}{6}T_s(s_1 + 2s_2 + 2s_3 + s_4) \quad (2.23)$$

where $s_1 = \omega_k$; $s_2 = f(\theta_k + \frac{T_s}{2}s_1)$; $s_3 = f(\theta_k + \frac{T_s}{2}s_2)$; $s_4 = f(\theta_k + T_s s_3)$.

2.2.5 Obstacle's detection and avoidance

Unmanned vehicles are used increasingly in complex environment to carryout sophisticated tasks autonomously, obstacles detection and avoidance is unavoidable in control design. Path and motion planning method was sparingly used to navigate mobile systems, for example [Margraff, Stéphant, and Labbani-Igbida 2020; Fuseiller et al. 2018], in many cases, mobile robots operating in a dynamic environment had to be enhanced with obstacle avoidance techniques for the safety of the objects around and that of the robot. In this thesis, the second approach of integrating obstacles avoidance with the control design was used. Some of the widely accepted obstacle avoidance techniques are summerized in [Bhavesh 2015]:

- Bug Algorithms: These are the simplest algorithms [Bhavesh 2015]. The robot moves on the shortest path from its current position towards the goal until it comes across an obstacle. The algorithm forces the robot to move tangentially around the obstacle's surface until it returns to its original path. The primitive bug algorithms make the robot circumnavigate the whole obstacle before returning to its original path.
- Artificial Potential Field (APF) Methods: In APF methods, the robot, obstacle, and the goal are considered as electric charges such that the robot and the obstacle have the same polarity so that repulsive force is created while the goal is assumed to have opposite polarity with the robot so that attractive force will be created [Lumelsky and Skewis 1990].

-
- Bubble Band Technique: In these methods as firstly proposed in [Borenstein and Koren 1988], the robot is surrounded by a "bubble" containing the maximum available free space which the robot can pass in any direction without collision.
 - Vector Field Histogram: This method was firstly proposed in [Quinlan and Khatib 1993] to deal with the issue of sensor noise through constructing a polar histogram using the most recent readings of the sensor. In the histogram, a probability of an obstacle's presence in a particular direction is plotted against the angle associated with the sonar sensor readings. A local occupancy grid map of the environment around the robot is created to compute the probabilities.

The polar histograms are used to determine all the passages large enough to avoid collision with the obstacle. The passage to be followed by the robot is selected by evaluating the cost function defined for each passage which is a function of the alignment of the robot's path with the goal and the difference between the current wheel orientation and the new direction. The passage has a minimum cost function is selected.

In this work, bug-type algorithm is selected due to its simplicity. The obstacles are assumed to be spherical with radius R_{obs} and each being positioned in a point described by its Cartesian coordinates (x_{obs}, y_{obs}) . Collision with the obstacles is avoided by including the following function as an inequality constraint in the NMPC formulation.

$$\sqrt{(x_{rob} - x_{obs})^2 + (y_{rob} - y_{obs})^2} \geq (R_{obs} + R_{rob}) \quad (2.24)$$

where x_{rob} and y_{rob} are the position of the robot (pursuer or evader) in x-y plane.

Detection of the obstacles can be achieved using sensors to give in real-time the size and the positions of the obstacles. In mobile robotics, LIDAR (acronyms for light detection and ranging) sensor is usually employed to detect the position and the size of an obstacle as in [Catapang and Ramos 2016]. Also, a set of camera system can be used for localization and detection of robots and obstacles.

2.3 Model predictive control

Model predictive control (MPC) is a feedback implementation of optimal control using finite prediction horizon and online optimization. MPC is also known as receding horizon control (RHC) where a future control sequence minimizing an objective function is minimized over a finite horizon. The advantages of MPC comprise its ability to control multi-variable coupled dynamical systems, handle constraints on the states, handle constraints on control inputs, handle nonlinearities in the systems model conceptually. In addition, MPC has a systematic design approach and has well-understood tuning parameters, i.e prediction horizon length, and weighting matrices. [Wang 2009; Alami 2006; Rawlings, Mayne, and Diehl 2019].

The system model is central to the design of MPC, thus if a linear model is used in the design, it's regarded as Linear Model Predictive Control (LMPC); otherwise called Nonlinear

Model Predictive Control if a nonlinear model is used. It should also be pertinent to know that MPC can be formulated using continuous or discrete-time.

MPC has a wide area of application in both industrial and research community. For example, MPC was employed in power electronic devices in [Cortes et al. 2008; Kouro et al. 2009], heating systems in [Široký et al. 2011], process control in [Bartee et al. 2009; Mhaskar 2006], wind turbines in [Lio, Rossiter, and Jones 2014], marine surface vessels in [Fahimi 2007; Oh and Sun 2010], unmanned aerial vehicles in [Alexis, Nikolakopoulos, and Tzes 2011; Dalamagkidis, Valavanis, and Piegł 2011; Alexis et al. 2011; Bouffard, Aswani, and Tomlin 2012; Kang and Hedrick 2009; Eklund, Sprinkle, and Sastry 2012; Chemori and Marchand 2008], projectiles and missiles guidance in [Gross and Costello 2014; Oza and Padhi 2012], parallel robots in [Vivas and Poignet 2005], holonomic robots in [Kanjanawanishkul and Zell 2009] and nonholonomic robots with point-stabilization in [Gu and Hu 2005; Kuhne, Lages, and Silva 2005; Worthmann et al. 2016; Worthmann et al. 2015], trajectory tracking in [Mehrez, Mann, and Gosine 2013; Xie and Fierro 2008; Mehrez, Mann, and Gosine 2015; Dixon et al. 2000; Michalek and Kozłowski 2010; Lee et al. 2001] and path following problems in [Aguiar, Hespanha, and Kokotović 2008; Nielsen, Fulford, and Maggiore 2010].

2.3.1 Mathematical formulation of MPC

In this section, the mathematical formulation of MPC was presented. As highlighted in the previous section, MPC can be formulated as LMPC or NMPC. To formulate LMPC, we consider a discrete linear time-invariant system of the form:

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k) \quad (2.25)$$

where k is the discrete time instant, $\mathbf{x} \in \mathbb{R}^n$ is the state, $\mathbf{u} \in \mathbb{R}^m$ is the control input of the system. $A \in \mathbb{R}^{n \times n}$ and $B \in \mathbb{R}^{n \times m}$ are the state and input matrices of the system respectively.

The LMPC can be implemented by solving the following open loop optimal control problem(OCP):

$$\min_{\mathbf{u} \in \mathbb{R}^{n\mathbf{u} \times N}} J_N(\mathbf{x}_0, \mathbf{u}) \quad (2.26)$$

Subject to

$$\begin{cases} \mathbf{x}(0) = \mathbf{x}_0, \\ \mathbf{x}(k+1) = A\mathbf{x}(k) + B\mathbf{u}(k); & k \in \{0, 1, \dots, N-1\}, \\ \mathbf{x}_{min} \leq \mathbf{x}(k) \leq \mathbf{x}_{max} & k \in \{1, 2, \dots, N\}, \\ \mathbf{u}_{min} \leq \mathbf{u}(k) \leq \mathbf{u}_{max} & k \in \{0, 1, \dots, N-1\}, \end{cases} \quad (2.27)$$

The objective function $J_N(\mathbf{x}_0, \mathbf{u})$ is generally defined by:

$$J_N(\mathbf{x}_0, \mathbf{u}) = \sum_{k=0}^{N-1} V(\mathbf{x}(k), \mathbf{u}(k)) + W(\mathbf{x}(N)) \quad (2.28)$$

The term $V(\mathbf{x}(k), \mathbf{u}(k))$ is called the running cost which can be computed by penalizing the deviation of the system's state $\mathbf{x}(k)$ and control input $\mathbf{u}(k)$ from the reference state $\mathbf{x}_r(k)$ and reference control input $\mathbf{u}_r(k)$ respectively. Generally, the running cost are defined as:

$$V(\mathbf{x}(k), \mathbf{u}(k)) = \|\mathbf{x}(k) - \mathbf{x}_r(k)\|_Q^2 + \|\mathbf{u}(k) - \mathbf{u}_r(k)\|_R^2 \quad (2.29)$$

Where $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$ are positive definite symmetric weighting matrices. N is the prediction horizon assuming that the length of the prediction and control horizon is the same. As previously explained, in point stabilization problem, the state reference $\mathbf{x}_r(k)$ is a fixed value, thus the control, reference $\mathbf{u}_r(k) = 0$. In the case of trajectory tracking problem, the state reference $\mathbf{x}_r(k)$ is time-varying, therefore the deviation of control input from the reference can be penalized due to computational advantages such as rendering the optimal control problem easier, avoiding control values with expensive energy [Mehrez 2017].

The term $W(\mathbf{x}(N))$ are referred to as terminal cost which is used for stability purpose. It can be computed by penalizing the last entry from the state prediction $\mathbf{x}(N)$ from its reference $\mathbf{x}_r(N)$. Terminal cost can be defined as:

$$W(\mathbf{x}(N)) = \|\mathbf{x}(N) - \mathbf{x}_r(N)\|_P^2 \quad (2.30)$$

Where $P \in \mathbb{R}^{n \times n}$ is a positive definite weighting matrix. The solution of the optimal control problem 2.26 is the optimal control sequence of the form:

$$\mathbf{u}^* = (\mathbf{u}^*(0), \mathbf{u}^*(1), \dots, \mathbf{u}^*(N-1)) \quad (2.31)$$

On the other hand, the formulation of NPMPC uses nonlinear systems dynamics of the form

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \quad (2.32)$$

The NMPC can be implemented by solving the following open loop optimal control problem (OCP):

$$\min_{\mathbf{u} \in \mathbb{R}^{n_{\mathbf{u}} \times N}} J_N(\mathbf{x}_0, \mathbf{u}) \quad (2.33)$$

Subject to

$$\begin{cases} \mathbf{x}(0) = \mathbf{x}_0, \\ \mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)); & k \in \{0, 1, \dots, N-1\}, \\ \mathbf{x}_{min} \leq \mathbf{x}(k) \leq \mathbf{x}_{max} & k \in \{1, 2, \dots, N\}, \\ \mathbf{u}_{min} \leq \mathbf{u}(k) \leq \mathbf{u}_{max} & k \in \{0, 1, \dots, N-1\} \end{cases} \quad (2.34)$$

2.4 Control design and implementation

In this section, the mathematical formulation of NMPC, the solution, and coding approach, and the simulation environment for implementing the controller was presented. The control system architecture is depicted in Figure 2.3.

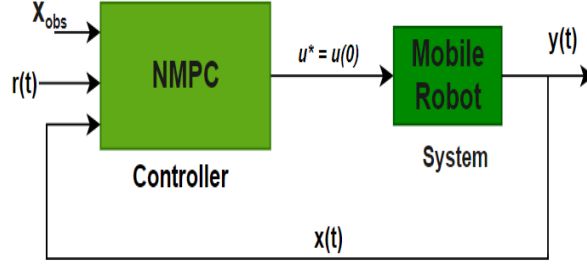


Figure 2.3: Control system architecture

2.4.1 Point stabilization with static obstacles avoidance using NMPC

To stabilize a nonholonomic mobile robot to a particular reference state $\mathbf{x}_r = [x_r, y_r, \theta_r]^T (m, m, rad)^T$, from a given initial state $\mathbf{x}_0 = [x_0, y_0, \theta_0]^T (m, m, rad)^T$, while avoiding static obstacles with parameters $[x_{obs}, y_{obs}, d_{obs}]^T (m, m, m)^T$, using NMPC, the following optimal control problem will be formulated:

$$\min_{\mathbf{u}} J = \|\mathbf{x}(N) - \mathbf{x}_r(N)\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}(k) - \mathbf{x}_r(k)\|_Q^2 + \|\mathbf{u}(k)\|_R^2 \quad (2.35)$$

subject to:

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (2.36a)$$

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k)), \quad k = 0, 1, \dots, N-1 \quad (2.36b)$$

$$\sqrt{(x(k) - x_{obs}(k))^2 + (y(k) - y_{obs}(k))^2} \geq \frac{(d_{obs} + d_{rob})}{2}, \quad obs = 1, \dots, M \quad (2.36c)$$

$$x_{min} \leq x(k) \leq x_{max} \quad (2.36d)$$

$$y_{min} \leq y(k) \leq y_{max} \quad (2.36e)$$

$$v_{min} \leq v(k) \leq v_{max} \quad (2.36f)$$

$$\omega_{min} \leq \omega(k) \leq \omega_{max} \quad (2.36g)$$

where M is the number of obstacles, d_{rob} represents the diameter of the robot. The first part of the cost function is added for stabilizing the controller while the second part is the quadratic running cost with $\mathbf{u}_r = 0$ because we are tracking a static reference. The constraint in (2.36a) represents the initialization with a new measurement or estimate after every sampling time.

2.4.2 Point stabilization with dynamic obstacles avoidance using NMPC

For dynamic obstacles with a time-varying parameter $[x_{obs}(t), y_{obs}(t), d_{obs}(t)]^T (m, m, m)^T$, The controller can be designed to solve the following optimal control problem:

$$\min_{\mathbf{u}} J = \|\mathbf{x}(N) - \mathbf{x}_r(N)\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}(k) - \mathbf{x}_r(k)\|_Q^2 + \|\mathbf{u}(k)\|_R^2 \quad (2.37)$$

subject to:

$$\mathbf{x}(0) = \mathbf{x}_0 \quad (2.38a)$$

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k)), \quad k = 0, 1, \dots, N-1 \quad (2.38b)$$

$$\sqrt{(x(k) - x_{obs}(t))^2 + (y(k) - y_{obs}(t))^2} \geq \frac{(d_{obs}(t) + d_{rob})}{2} \quad (2.38c)$$

$$x_{min} \leq x(k) \leq x_{max} \quad (2.38d)$$

$$y_{min} \leq y(k) \leq y_{max} \quad (2.38e)$$

$$v_{min} \leq v(k) \leq v_{max} \quad (2.38f)$$

$$\omega_{min} \leq \omega(k) \leq \omega_{max} \quad (2.38g)$$

Similarly, M is the number of obstacles, d_{rob} represents the diameter of the robot. The first part of the cost function is added for stabilizing the controller while the second part is the quadratic running cost with $\mathbf{u}_r = 0$ because we are tracking a static reference. The constraint in (2.38a) represents the initialization with a new measurement or estimate after every sampling time.

The weight matrices are tuned and the best values that stabilize the controller are chosen. The Q and R matrices are found to be diagonal matrices with diagonal elements defined as $(1, 1, 0.001)$ and $(1, 1)$ respectively. The weight on the terminal penalty cost is found to be $1000 * Q$.

2.5 Numerical simulations

In this work, the algorithm was implemented in a MATLAB/Simulink environment. The optimal control problem was converted to a nonlinear programming problem using a multiple shooting approach (where both the states and the control variables are considered as optimization parameters). An open-source symbolic framework for algorithmic differentiation and nonlinear optimization known as CasADi [Andersson et al. 2019] was interfaced for the computations. The output of the controller is sent to the system for an update whereas the state measurement, the reference position, and the obstacles parameters are feed to the controller at each sampling instant for re-computation of the new control strategy.

2.5.1 Discretization method performance comparison

The result using the Euler discretization method is presented in Figure 2.4. Although the controller was able to move from its initial pose to the goal position, it should be noted that the controller's output was not smooth with the Euler method, which is not suitable for physical implementation on robots. We then perform the same experiment with the RK4 discretization method. The result is presented in Figure 2.5 and shows a smooth trajectory of the control profiles much better than the Euler method.

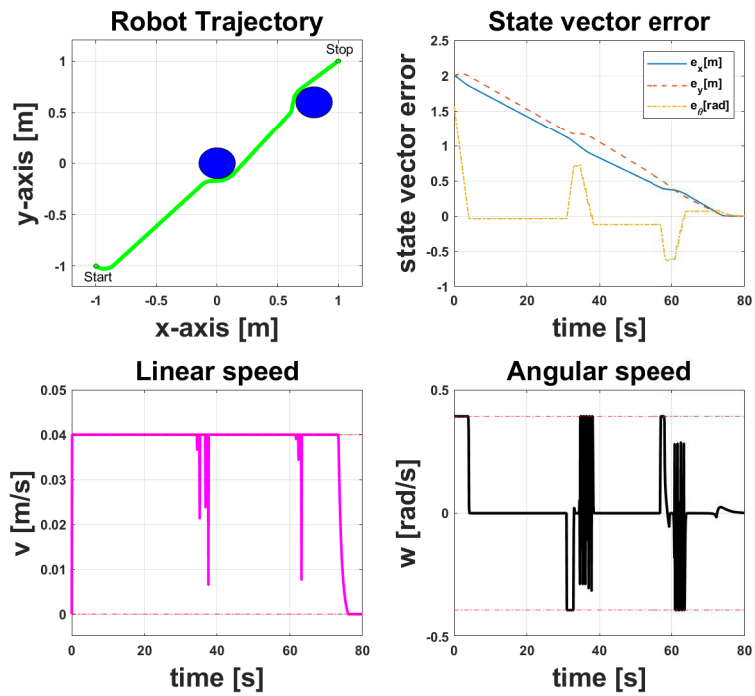


Figure 2.4: Euler Discretization

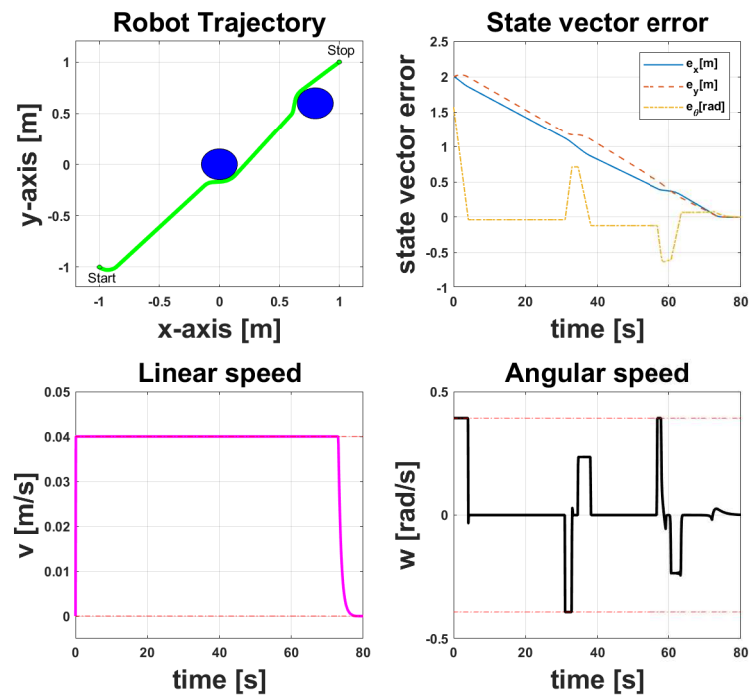


Figure 2.5: Runge-Kutta 4th Order Discretization

2.5.2 Static obstacles avoidance

To achieve point stabilization with static obstacles avoidance using discrete NMPC, the robot starts from an initial pose $\mathbf{x}_0 = [-1m, -1m, -\frac{\pi}{4}rad]$ and aimed to reach a goal pose $\mathbf{x}_g = [1m, 1m, \frac{\pi}{4}rad]$ encountering two static obstacles at $(0m, 0m)$ and $(0.8m, 0.6m)$. The diameter of each obstacle is $0.3m$ while the diameter of the robot is $0.04m$. The sampling time is chosen to be $0.1sec$ and a prediction horizons of lengths $N = [5, 10, 20]$ were selected leading to prediction horizon times of $T = [0.5, 1, 2]secs$, respectively. The robot's actuator saturation limits are selected randomly such that the linear velocity v ranges from 0 to $0.04m/s$ and the angular velocity ω ranges from $-\frac{\pi}{8}rad/s$ to $\frac{\pi}{8}rad/s$.

To examine the effects of prediction horizons on the computation time, we compare the results in Figures 2.6, 2.7 and 2.8. The controller was able to move the robot from its initial state to the final state while avoiding obstacles as we can see in the robot trajectories and state error vector of each figure. The difference was only visible on the control action profiles. The longer the prediction horizon, the smoother the control action profile. In Figure 2.9, the average computation time was displayed on boxplot for the three prediction horizons. It can be seen that the longer the prediction horizon the higher the computation time and vice-versa.

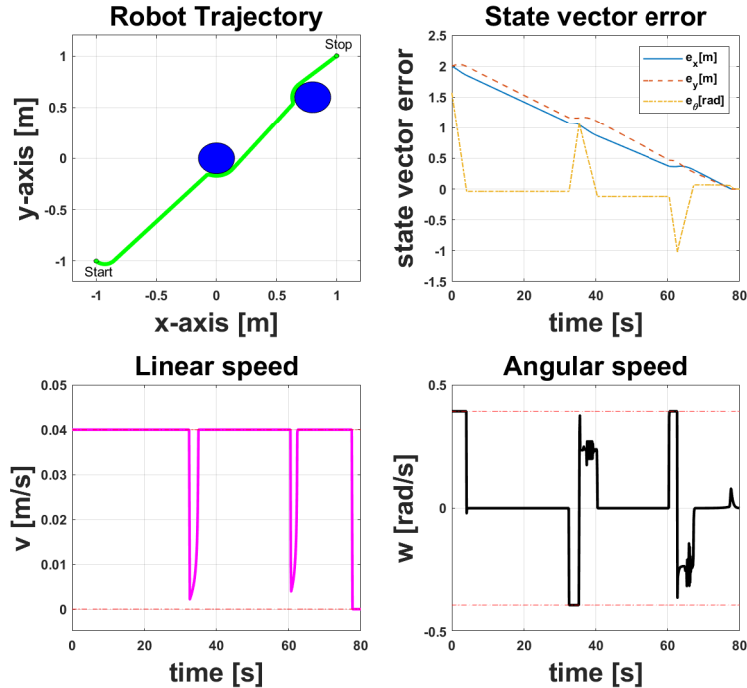


Figure 2.6: Runge-Kutta 4th Order Discretization Method, N=5

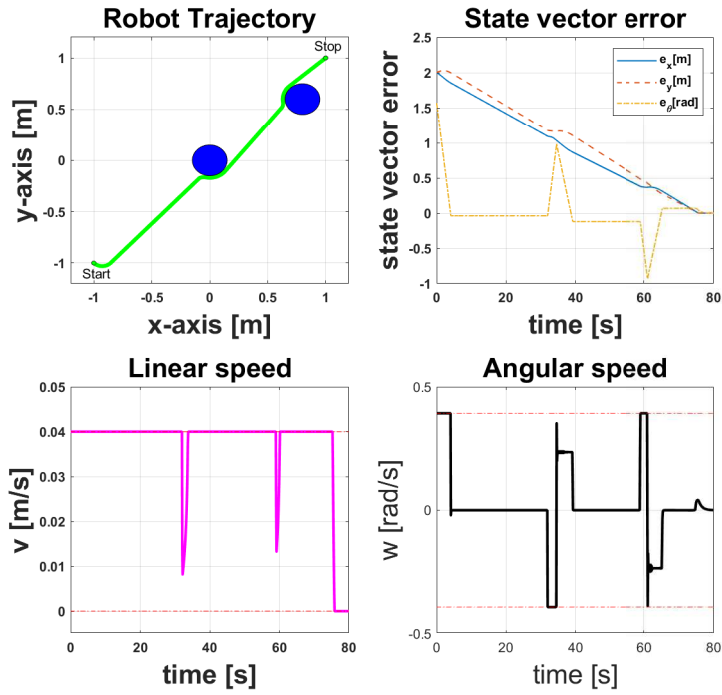


Figure 2.7: Runge-Kutta 4th Order Discretization Method, N=10

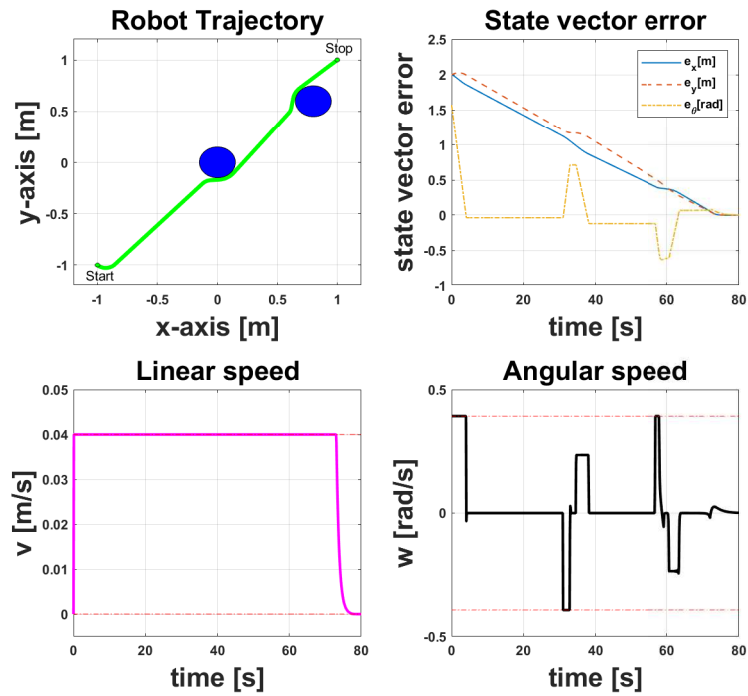


Figure 2.8: Runge-Kutta 4th Order Discretization Method, N=20

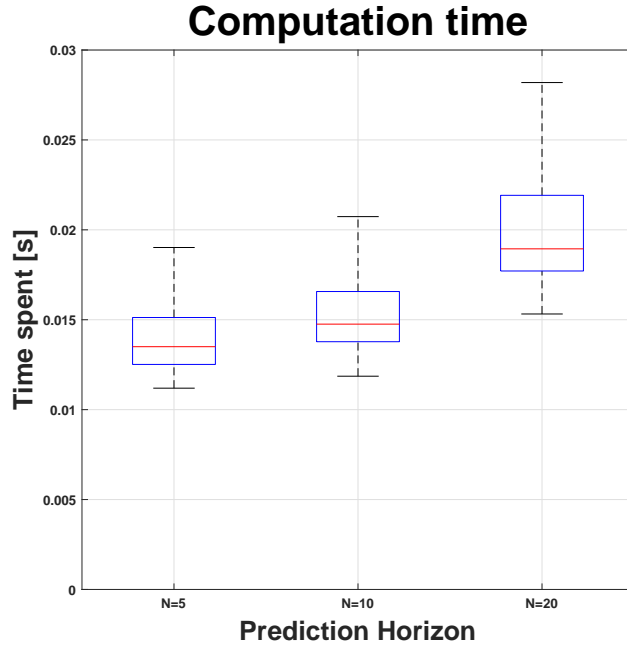


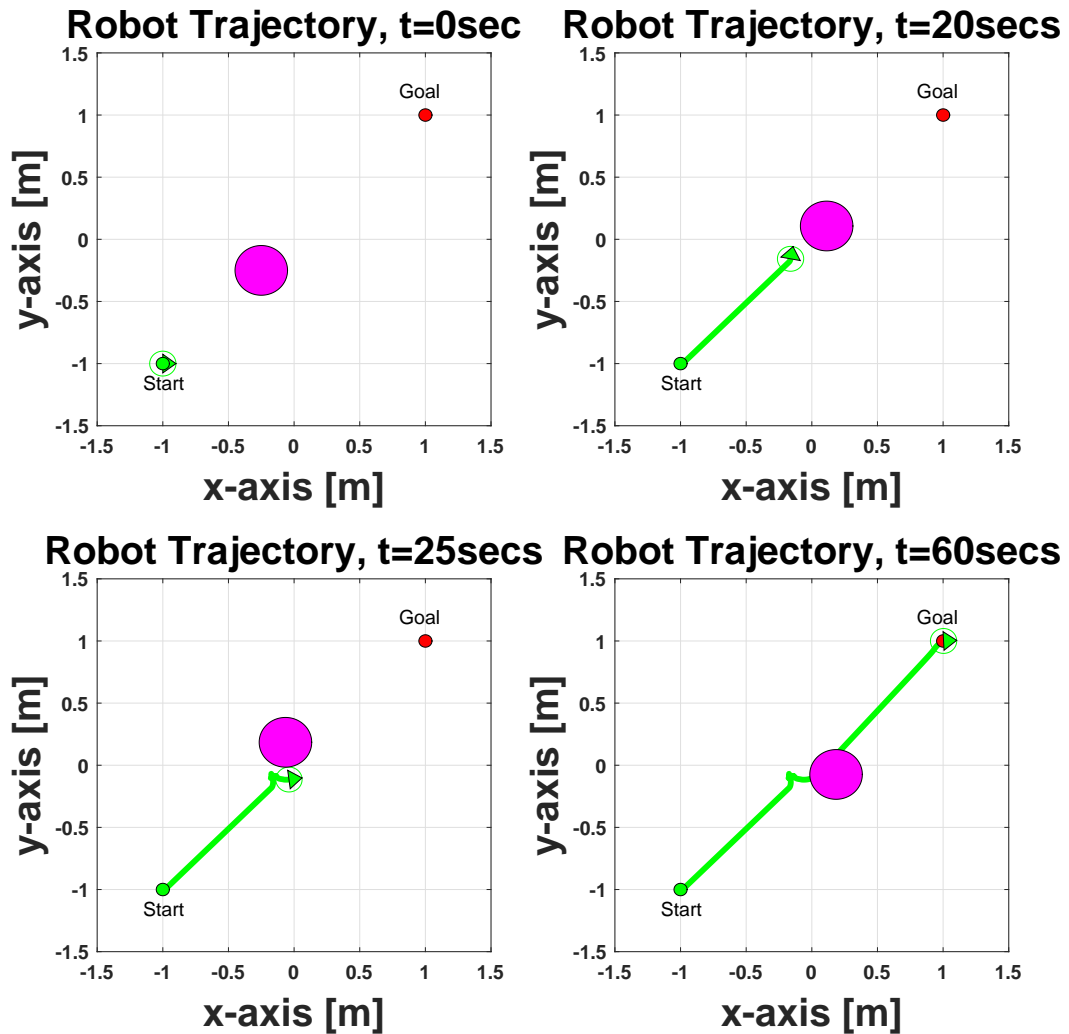
Figure 2.9: Computation time

2.5.3 Dynamic obstacles avoidance

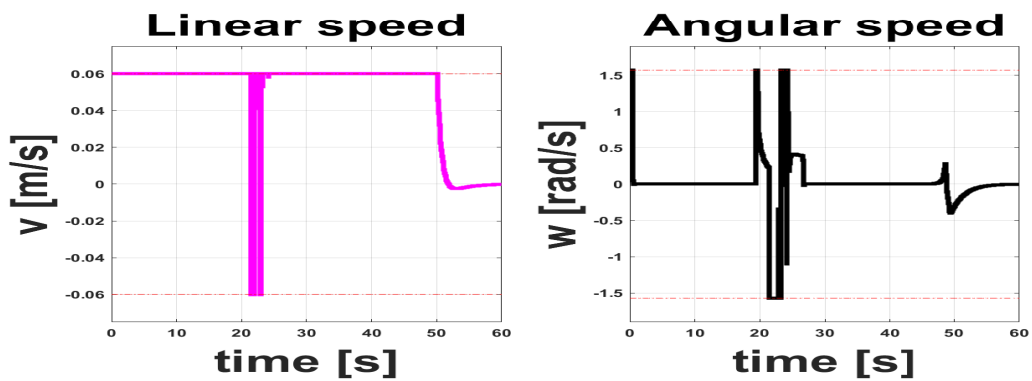
Simulation results of dynamic obstacles avoidance for mobile robots are presented in two scenarios. The first scenario with one single dynamic obstacle is depicted in Figure 2.10 where the robot autonomously moved from its initial pose $[-1m, -1m, 0rad]$ and aimed to reach its goal pose $[1, 1, 0rad]$ while encountering a dynamic obstacle moving in a counter-clockwise direction between the start and the goal positions. The obstacle was approximated as a circle with a radius of $0.2m$.

The trajectory of the robot and the obstacles at four-time instances are depicted in Figure 2.10a. At $t = 0$, the initial posture of the robot and the obstacles as well as the initial position and the goal position are marked. After $t = 20secs$, the robot moved from its initial position and encountered the obstacle at around $0.15m, 0.18m$ mark, i.e the obstacle has moved from its initial position to the current position. It should be noted here that the obstacle is moving in a counter-clockwise direction with a speed of $0.04m/s$ and a turning rate of $\pi/20$ rad/s. The robot avoided collision with the obstacle even though the controller does not know the information about the movement of the obstacle and then moved towards its goal.

The control profiles are presented in Figure 2.10a showed that the controller respected the actuator saturation as the limits on both linear speed and angular speed are within the range. Also, it can be noticed that some slight oscillations occurred at around $22secs$ because the controller struggled to respect the constraints for obstacles avoidance as the robot approached the moving obstacle.



(a) State Trajectory



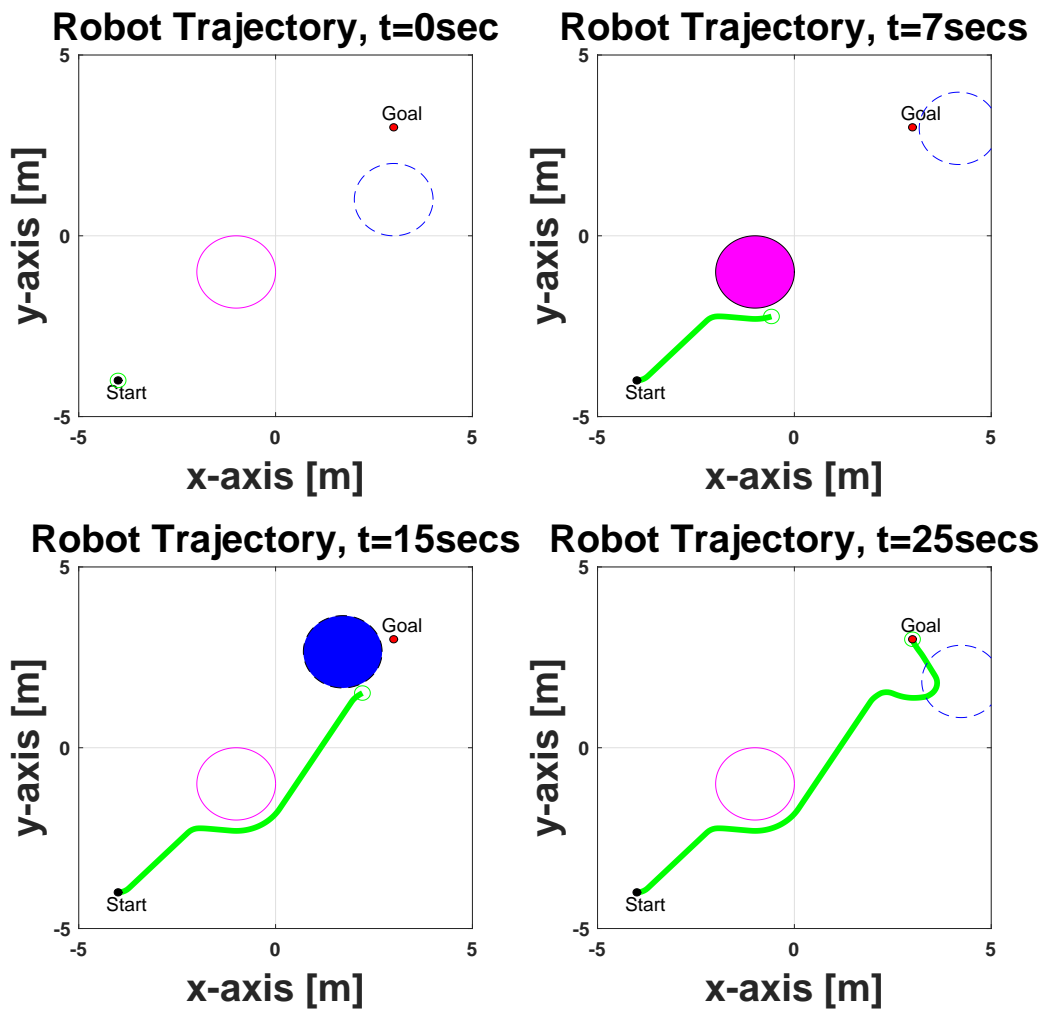
(b) Control Profile

Figure 2.10: Simulation: Dynamic Obstacles Avoidance

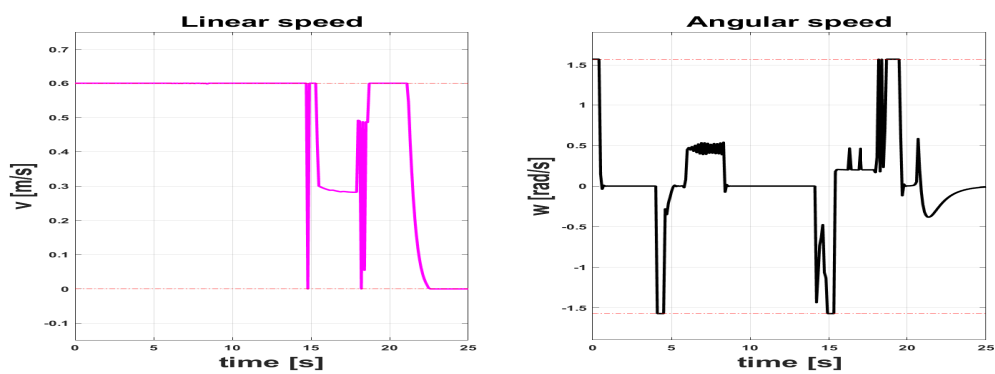
The second scenario, a bit complex, was composed so that the robot must avoid static and dynamic obstacles moving circularly around the goal point. In Figure 2.11, dynamic obstacles avoidance was presented for a prediction horizon of 10. The initial pose of the robot is $[-4m, -4m, 0rad]$, the goal position is $[3m, 3m, \frac{\pi}{2}rad]$, the static obstacle parameter is $[-1m, -1m]$ with a radius of $1m$, while the initial position of the dynamic obstacle is $[3m, 1m]$ with a radius of $1m$. The robot's state trajectory was displayed at four different instances in Figure 2.11a. At $t = 0$, the initial positions of the robot marked Start, the goal position marked Goal, and the obstacles are at their initial positions marked, $[-1m, -1m]$ and $[3m, 1m]$, respectively. After $7secs$, the robot encountered the static obstacle, encircled it, and moved towards the goal point. After $15secs$, the robot encountered the moving obstacle and then moved away from it and continue towards the goal point. The robot finally reaches its target and stabilizes at the required orientation.

The scenario was repeated in Figures 2.12 and 2.13 with a prediction horizon of 20 and 30, respectively. It can be observed that with a higher prediction horizon, detection of the obstacle came faster and thus avoidance starts early.

In Figure 2.14, the comparison of prediction horizons was presented and their effects on the computation time was observed. With the prediction horizon of length 10, the average computation time is 0.035secs. When the prediction horizon is increased to 20, the average computation time was found to be 0.04secs, while when the prediction horizon is increased to 30, the average computation time was found to be 0.043secs.

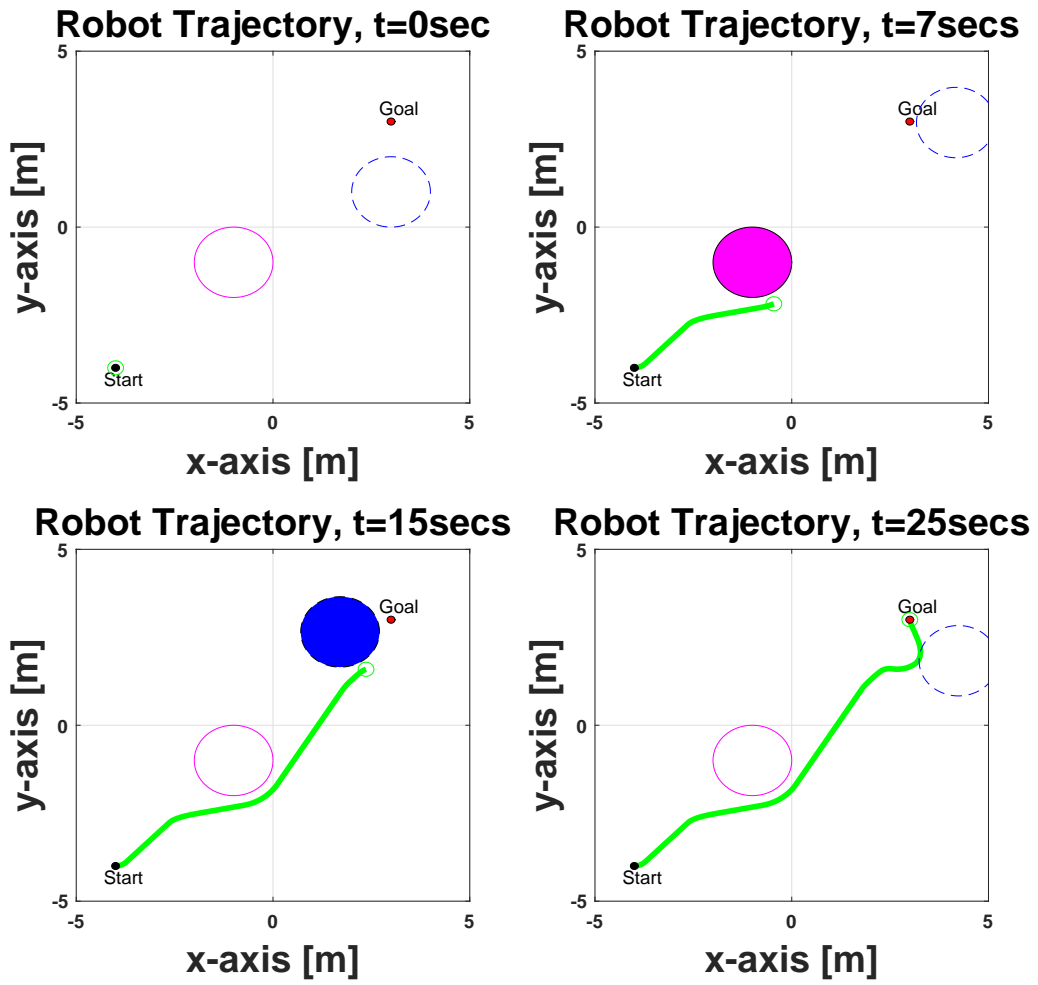


(a) State Trajectory

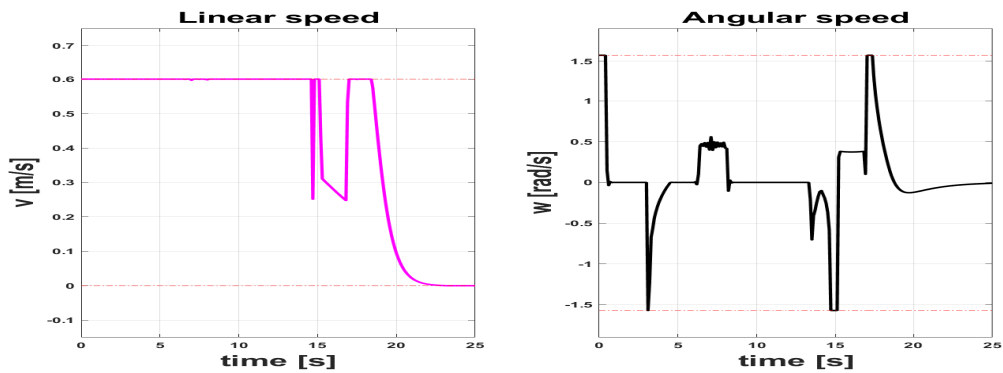


(b) Control Profile

Figure 2.11: Dynamic Obstacles Avoidance, Circular movement, $N = 10$

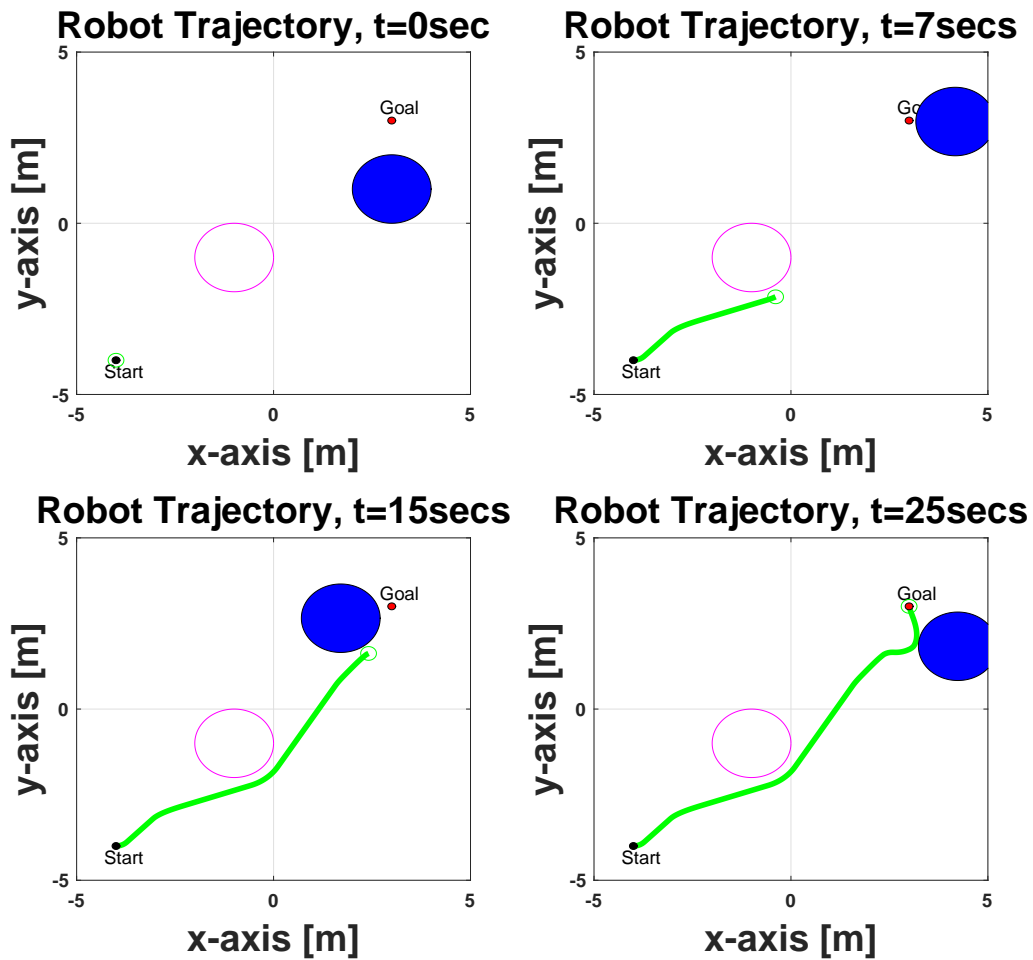


(a) State Trajectory

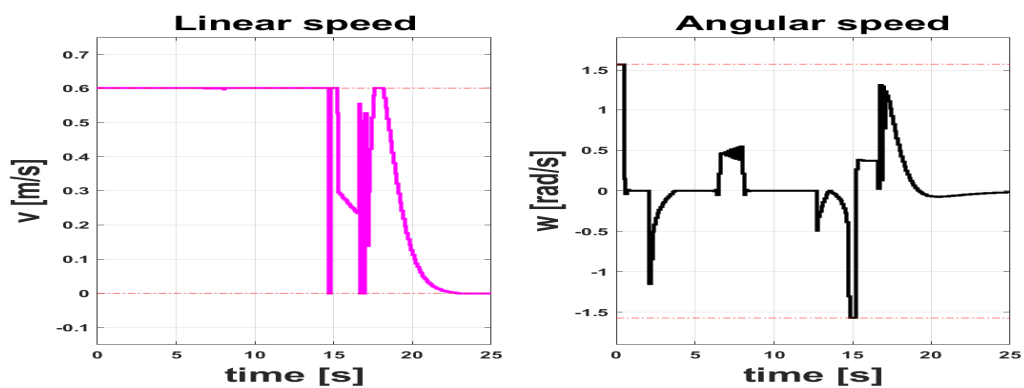


(b) Control Profile

Figure 2.12: Dynamic Obstacles Avoidance, Circular movement, $N = 20$



(a) State Trajectory



(b) Control Profile

Figure 2.13: Dynamic Obstacles Avoidance, Circular movement, $N = 30$

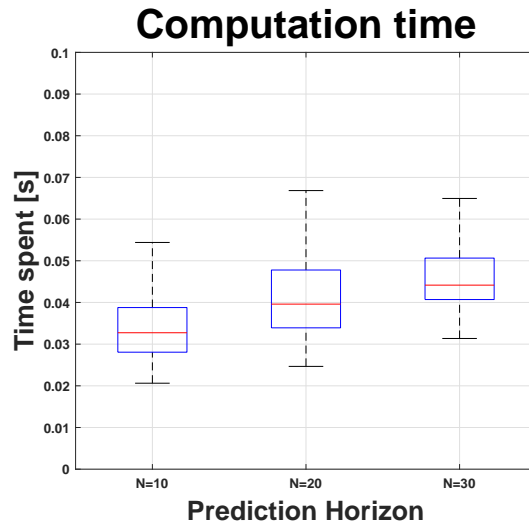


Figure 2.14: Computation time

2.6 Conclusion

This chapter dealt with static and dynamic obstacles avoidance for point stabilization problems using discrete NMPC. Two discretization methods are compared the Euler and RK4 method, the latter achieved better performance than the former, especially for a larger sampling time. The obstacle avoidance was integrated as an inequality constraint during the formulation of the NMPC. Stability was achieved using terminal cost inclusion. The effect of a long prediction horizon on the controller computation time was analyzed, showing that a higher prediction horizon leads to higher computation costs. The results obtained showed that the proposed approach can be used to handle both static and dynamic obstacles moving with unknown velocities without the need for measurement or additional computation.

Pursuit-Evasion Games Using Nonlinear Model Predictive Control

This chapter extends the results of the previous chapter by involving a moving target. Rather than stabilizing the robot to point, the robot was made to pursue another robot that is controlled by a conflicting objective. This scenario is called the pursuit-evasion problem and has differed from the conventional trajectory tracking problems where a mobile robot follows a moving target whose behavior is known. The two-player pursuit-evasion game is tackled such that the objective of the pursuer is to capture the evader by minimizing the distance and orientation between them. The objective of the evader on the other hand is to escape from the pursuer by maximizing the distance and orientation between them. Nonlinear model predictive control approach is used to solve the problem assuming the states of each player can be measured or estimated by the opponent. The stability of the controllers is achieved by including a stabilizing cost function in the formulation. The stability method is compared with a recent method that involve adding monotonically increasing weight in the cost function instead of a stabilizing cost function or constraints. Various numerical simulations are conducted in the presence and absence of obstacles as well as in the presence of measurement noise. It was shown that the proposed control approach was effective for solving the problem.

3.1 Overview of pursuit-evasion games

The pursuit-evasion game is a problem where an autonomous agent or group of agents engages another agent or group of agents that have conflicting objectives. The game has received attention in both the cooperative and non-cooperative control literature. It provides a general framework that mathematically formalizes important applications in different areas such as navigation, surveillance, analysis of biological behaviors, conflict, and combat operations [Marden and Shamma 2018; Weintraub, Pachter, and Garcia 2020]. The goal is to determine a strategy that enables an autonomous agent to perform a set of actions against the opponent. For example, the pursuer aims at determining a strategy that will result in capture or interception of the evader, while the evader aims to find a strategy that could make it escape or maximize the time of capture. Both teams are therefore playing their best strategies which is the worst-case scenario from the opponent's point of view.

Pursuit-evasion differential game was discovered in [Isaacs 1956] as a result of an aspiration

to design the best strategies which optimize a certain objective against the worst possible behaviors of the opponent and also provide robustness with respect to all possible behavior of the opponent. The underlying assumption was that each player is well-informed about the opponent's behavior. The level of information each player has on the opponent is very important in determining its best strategy. This information on the opponent can be full, partial, deterministic, or stochastic. Therefore this would be the pivotal focus of this thesis.

Broadly, the pursuit-evasion game can be categorized into two classes. The most popular category is the zero-sum game which is the case of a pursuit-evasion problem between two autonomous players; the pursuer and the evader playing against each other. The second category is the case of a multiple-player pursuit-evasion game in which a team of pursuers plays against the team of evaders.

3.1.1 Two player pursuit-evasion games

In a two-player pursuit-evasion game, a single pursuer plays against a single evader to capture the latter. The goal of the evader is to avoid being captured by the pursuer. The payoff, cost, or utility of this game is time-of-capture or time-to-capture can be formulated as a common performance functional shared by the two players. The resulting problem is known as min-max or a zero-sum game. The optimal solution for the performance function is one in which the strategy of the pursuer seeks to minimize it while the strategy of the evader is to maximize it. The dynamics of all players is regarded as a constraint that can be linear or nonlinear. In [Isaacs 1956], the concepts of game theory, control theory, and calculus of variations were employed to solve a dynamic conflict between autonomous players. The optimal strategy pair is the Nash equilibrium of the game, the value function of the payoff is the value of the game which is used to determine the winning player. It was argued in [Ho, Bryson, and Baron 1965] that the min-max solution proposed by Isaac did not make extensive use of classical variational technique, thus describing the work to resemble a dynamic programming approach to optimization problems. Thus the paper shows that variational techniques can be applied to solve differential games. In [Tolwinski 1989], the author solved zero-sum differential games by approximating the dynamic programming equation using a sequence of finite-state Markov games. The approach called the modified policy iteration method was employed to solve a combat problem related to the two-car game of Isaacs.

The reachable set-based approach was employed to find the optimal solutions of pursuit-evasion differential game e.g in [Mitchell, Bayen, and Tomlin 2005], where an algorithm was presented for numerically computing the backward reachable set for a two-player, nonlinear differential game with a general target set. The algorithm is based on the formulation of reachability in terms of the viscosity solution of a time-dependent Hamilton Jacobi Isaacs Partial Differential Equation. The authors in [Sun et al. 2017] employed a reachability-based approach to solving pursuit-evasion games between two players in the presence of dynamic environmental disturbances such as winds, sea currents, etc. Termination of the game was achieved by including reachable sets. In [Dong, Zhang, and Jia 2012], an improved artificial potential field method is used to solve pursuit-evasion game problems in the presence of static

obstacles. A hybrid algorithm based on the differential game and an improved potential field is developed to allow for flexibility depending on the environment of the game. Proportional navigation guidance-based method was borrowed from the missile navigation and target tracking to compare the performance of pursuer-centric against evader-centric strategies for wheeled holonomic mobile robot applications in [Kumar and Ojha 2019].

Predictive control-based approaches have also been proposed to solve pursuit-evasion games. In [Eklund, Sprinkle, and Sastry 2012], the nonlinear model predictive control approach was used to solve the pursuit-evasion game in 3-D space between two homogeneous, fixed-wing aircrafts such that each aircraft predicts both its optimal trajectory and that of its opponent. The approach takes into consideration the hard constraints on the linear and angular velocity of each player. The game is said to be symmetric such that a pursuer can become an evader and vice-versa, and have assumed that each player has full information of the game. The authors in [Tzannetos, Marantos, and Kyriakopoulos 2016] dealt with a pursuit-evasion game between two heterogeneous players; the unmanned aerial vehicle and the unmanned ground vehicle using nonlinear model predictive control. Full information was assumed, and each player predicted its opponent's strategy before determining its optimal strategy. The paper considered the relative distance between the two players as the states of the game and thus it cannot incorporate states and inputs constraint as well as obstacles avoidance.

3.1.2 Multi-player pursuit-evasion

Multi-player pursuit-evasion games can be categorized according to the number of players in each team. The general category is the multiple pursuers against multiple evaders while particular categories involve multiple pursuers against a single evader and a single pursuer against multiple evaders. Safe reachable area minimization method where multiple pursuers cooperate to capture a single evader was proposed in [Huang et al. 2011; Zhou et al. 2016], where a decentralized control scheme was presented based on the Voronoi partition of domain of the game, such that the pursuers jointly minimize the area of evader's Voronoi cell. The approach was applied to both holonomic and non-holonomic mobile robots in [Kothari, Manathara, and Postlethwaite 2014; Kothari, Manathara, and Postlethwaite 2017]. In [Bakolas and Tsiotras 2010] Voronoi partition approach was also applied for multiple evader pursuit-evasion problems. The complexity of computing the Voronoi cells especially for nonholonomic systems opens the door for another approach for handling cooperative pursuit problems. In [Garcia et al. 2017], a game of two pursuers against a single evader was proposed based on the Hamiltonian formulation and based on the geometric properties of the game. Cooperation in this method occurs only when none of the two pursuers have the geometric advantage to capture the evader alone, thus they cooperate by finding the point of intersection between two Apollonian circles. Other works by the same authors on the same holonomic system are presented in [Garcia et al. 2019; Von Moll et al. 2018]. Cooperative defense strategy against predator by two evaders has been presented in [Fuchs, Khargonekar, and Evers 2010] while strategies for capturing multiple evaders by multiple pursuers has been presented in [Pier-son, Wang, and Schwager 2017; Souza et al. 2021]. However, apart from the computational

complexity and applicability of this method to holonomic systems only, constraints on the states and inputs of the systems cannot be imposed and thus obstacles avoidance was not incorporated.

In this chapter, the pursuit-evasion game will be handled using NMPC approaches for both players such that each player can measure the states of the opponent. We will investigate the stability of the controllers with terminal cost inclusion and compare it with a recent stabilizing method that employs monotonically increasing cost function instead of terminal cost. The robustness of the controller would be verified by adding some measurement noise and obstacles.

The remaining parts of this chapter are as follows: The problem statement would be presented in section 3.2. Controllers design using NMPC technique with stabilizing costs and weights would be discussed in section 3.3. Numerical results would be presented in section 3.4 while conclusions would be made in section 3.5.

3.2 Problem statement

Consider a pursuit-evasion game between two non-holonomic unicycle robots, where the game is to be conducted inside the polytope, Ω in the presence and absence of obstacles. The aim of the pursuer is to capture the evader in the shortest possible time by moving towards the evader while the aim of the evader is to maximize the capture time. Both players must be able to avoid obstacles in the environment. The kinematic model of each player is given in (3.1).

$$\begin{cases} \dot{x}_i &= v_i \cos \theta_i \\ \dot{y}_i &= v_i \sin \theta_i \\ \dot{\theta}_i &= \omega_i \end{cases} \quad (3.1)$$

where the subscript i stands for i^{th} player such that $i \in \{p, e\}$ denotes pursuer and evader respectively. The states $(x_i, y_i) \in \Omega$ and θ_i are the position and orientation of i^{th} player respectively, while $(v_i, \omega_i) \in U_i$ represents the linear and the angular speeds respectively, which acts as a control inputs for the i^{th} player. U_i denotes the set of feasible control inputs for the i^{th} player and is assumed to be constrained.

Each player repeatedly computes its optimal strategy against the opponent until the game is terminated. The termination of the game occurs either when the game time elapses (meaning that the evader escaped) or when the evader has been captured. The capture condition is defined by computing the relative distance between the two players at every time instant using (3.2).

$$D(t) = \sqrt{(x_p(t) - x_e(t))^2 + (y_p(t) - y_e(t))^2} \quad (3.2)$$

The instantaneous distance is compared with a threshold distance which is defined to the sum of the radii of the two robots, where R_p and R_e denotes the radius of the pursuer and the

evader respectively.

$$D_{th} = R_p + R_e \quad (3.3)$$

The capture condition, which is winning from the pursuer's point of view and lost from the evader's point of view is defined in (3.4).

$$D(t) \leq D_{th}, \quad \text{for } t \geq 0 \quad (3.4)$$

The time at which the capture condition holds is defined as the capture time. The pursuit problem can be defined as:

Assuming full state feedback of both players are deterministic, design an online control algorithm that could move the pursuer towards the evader while avoiding obstacles such that constraints on states and control variables are respected, assuming that the evader is playing its best strategy.

Conversely, the evasion problem can similarly be defined as:

Assuming full state feedback of both players are deterministic, design an online control algorithm that could move the evader away from the pursuer while avoiding obstacles such that constraints on states and control variables are respected, assuming that the pursuer is playing its best strategy.

3.3 Nonlinear model predictive control

Since each player in pursuit-evasion game changes strategy at every decision instant to defeat its opponent, MPC technique can be a good candidate to solve the problem. In this context, the pursuer will solve the minimization problem while the evader will solve the maximization problem. The ability of MPC to predict the future behavior of the system using its model and make it possible to detect the presence of an obstacle and to avoid it as a constraint. However, the length of the prediction horizon to be selected must consider the property of the localization system.

The pursuer's NMPC is thus an online optimizing controller which takes in the evader's trajectory (current states at every decision time) $\mathbf{x}_e = [x_e, y_e, \theta_e]^T(m, m, rad)^T$ and the pursuer's initial state $\mathbf{x}_p = [x_p, y_p, \theta_p]^T(m, m, rad)^T$ as well as obstacles parameters $[x_{obs}, y_{obs}, d_{obs}]^T(m, m, m)^T$ to provide an optimal sequence of control actions over a finite horizon window. Only the first part of the sequence would be applied to the systems while the remaining values would be discarded. The procedure would then be repeated at the next decision time which takes in the updated measure of the variables. In essence, the pursuer would solve the following optimal control problem:

$$\min_{\mathbf{u}_p} J = \sum_{k=0}^{N-1} \|\mathbf{x}_p(k) - \mathbf{x}_e(k)\|_Q^2 + \|\mathbf{u}_p(k)\|_R^2 \quad (3.5)$$

subject to:

$$\mathbf{x}_p(0) = \mathbf{x}_p \quad (3.6a)$$

$$\mathbf{x}_p(k+1) = f_p(\mathbf{x}_p(k), \mathbf{u}_p(k)), \quad k = 0, 1, \dots, N-1 \quad (3.6b)$$

$$\sqrt{(x_p(k) - x_{obs})^2 + (y_p(k) - y_{obs})^2} \geq R_p + R_{obs}, \quad obs = 1, \dots, M \quad (3.6c)$$

$$x_{min} \leq x_p(k) \leq x_{max} \quad (3.6d)$$

$$y_{min} \leq y_p(k) \leq y_{max} \quad (3.6e)$$

$$v_{min} \leq v_p(k) \leq v_{max} \quad (3.6f)$$

$$\omega_{min} \leq \omega_p(k) \leq \omega_{max} \quad (3.6g)$$

where M is the number of obstacles, R_p represent the radius of the pursuer. The resulting optimal control action would be $\mathbf{u}_p = [\mathbf{u}_p(0), \mathbf{u}_p(1), \dots, \mathbf{u}_p(N-1)]$ and only $\mathbf{u}_p(0)$ would be applied to the pursuer.

The evader's controller on the other hand, will be a similar online optimizing controller which takes in the pursuer's trajectory (current states at every decision time) $\mathbf{x}_p = [x_p, y_p, \theta_p]^T(m, m, rad)^T$ and the evader's initial state $\mathbf{x}_e = [x_e, y_e, \theta_e]^T(m, m, rad)^T$ as well as obstacles parameters $[x_{obs}, y_{obs}, d_{obs}]^T(m, m, m)^T$ to provide an optimal sequence of control actions over a finite horizon window. Also, the first part of the sequence would be applied to the systems while the remaining values would be discarded. The procedure would then be repeated at the next decision time which takes in the updated measure of the variables. Consequently, the pursuer would solve the following optimal control problem:

$$\max_{\mathbf{u}_e} J = \sum_{k=0}^{N-1} \|\mathbf{x}_e(k) - \mathbf{x}_p(k)\|_Q^2 + \|\mathbf{u}_e(k)\|_R^2 \quad (3.7)$$

subject to:

$$\mathbf{x}_e(0) = \mathbf{x}_e \quad (3.8a)$$

$$\mathbf{x}_e(k+1) = f_e(\mathbf{x}_e(k), \mathbf{u}_e(k)), \quad k = 0, 1, \dots, N-1 \quad (3.8b)$$

$$\sqrt{(x_e(k) - x_{obs})^2 + (y_e(k) - y_{obs})^2} \geq R_e + R_{obs}, \quad obs = 1, \dots, M \quad (3.8c)$$

$$x_{min} \leq x_e(k) \leq x_{max} \quad (3.8d)$$

$$y_{min} \leq y_e(k) \leq y_{max} \quad (3.8e)$$

$$v_{min} \leq v_e(k) \leq v_{max} \quad (3.8f)$$

$$\omega_{min} \leq \omega_e(k) \leq \omega_{max} \quad (3.8g)$$

M is the number of obstacles, R_e represent the radius of the evader. The resulting optimal control action would be $\mathbf{u}_e = [\mathbf{u}_e(0), \mathbf{u}_e(1), \dots, \mathbf{u}_e(N-1)]$ and only $\mathbf{u}_e(0)$ would be applied to the evader.

3.3.1 Stability of NMPC

Even though the formulation of NMPC seems to be quite intuitive, the stability is not automatically guaranteed since the control sequence is obtained from a finite optimal control problem [Feng 2004]. The formulated NMPC problems in (3.5) and (3.7) could lead to divergent responses. The literature is rich with so many methods for stabilizing an MPC controller. Lyapunov stability is a tool used to analyze the stability of an MPC problem based on the monotonic property of the cost function. It was historically established that the value function of a finite horizon optimal control problem is universally accepted as a natural Lyapunov function for stability analysis when a terminal equality constraint is employed [Mayne et al. 2000]. Thus the cost function is employed as a Lyapunov function. Therefore, to guarantee stability, a global optimum must be found at each time step. The optimization problem with terminal equality constraint can be solved, but the computation time for finding the global optimum is very expensive. Thus, even if a feasible solution exists, convergence to that solution is not guaranteed. Later, proposals were made to modify the open-loop optimal control problem by including terminal cost or terminal constraint set to tackle both global optimality and feasibility problem. In this work, we will compare the two stability methods which comprise the terminal cost function and a contraction-based method.

3.3.1.1 Stabilizing terminal cost

One of the preliminary proposal for modifying open-loop optimal control problem to ensure closed-loop stability was the addition of a terminal cost function. In the context of MPC, the terminal cost function is non-trivial and has no terminal constraint [Mayne et al. 2000]. This method was proposed in 90's by [Mosca 1991] for predictive control of unconstrained linear system. The appropriate terminal cost function selected was $F(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q_f \mathbf{x}$ where the matrix Q_f is selected so that the sequence of P_i obtained by solving the Riccati difference equation in reverse time (i is time-to-go) with terminal condition $Q_0 = Q_f$ is *monotonically non-increasing*, that is ($Q_{i+1} \leq Q_i, \forall i \geq 0$). The authors proved that this choice yields a value function $V_N^0 = \frac{1}{2}\mathbf{x}Q_N\mathbf{x}$ and a receding horizon controller $K_N = K_{MPC}\mathbf{x}$ that is stabilizing. This interesting method of establishing stability can be extended to nonlinear systems. In the context of our PEG which employed NMPC, the pursuer's OCP formulated in (3.5) can be modified to include stabilizing terminal cost function as:

$$\min_{\mathbf{u}_p} J = \|\mathbf{x}_p(N) - \mathbf{x}_e(N)\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_p(k) - \mathbf{x}_e(k)\|_Q^2 + \|\mathbf{u}_p(k)\|_R^2 \quad (3.9)$$

Similarly, the evader's OCP (3.7) can be reformulated to include the stabilizing terminal cost cost function as:

$$\max_{\mathbf{u}_e} J = \|\mathbf{x}_e(N) - \mathbf{x}_p(N)\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_e(k) - \mathbf{x}_p(k)\|_Q^2 + \|\mathbf{u}_e(k)\|_R^2 \quad (3.10)$$

3.3.1.2 Stability of NMPC without stabilizing constraints or cost

Due to the feasibility issues related to the terminal constraints which generally require long prediction horizons to be used in the Model Predictive Control formulations. In addition, the presence of this constraint makes the computation more complex. Many researchers are now reluctant to include stability-related constraints in MPC formulation. In [Alamir and Bornard 1995], it was proved that stability can be achieved without terminal stability-related constraints or cost by using a sufficiently long prediction horizon. However, the underlying argument was that with a sufficiently long prediction horizon, the optimal decisions necessarily lead to open-loop trajectories with terminal appropriate properties. It is therefore an interesting task to employ formulations that involve short prediction horizons with no stability-related constraints or cost.

A contractive MPC formulation in which a constraint is added to the MPC formulation to force both the actual and the predicted state to contract to achieve stability. The contractive stability approach was proposed and proved in [Oliveira Kothare and Morari 2000]. However, the inclusion of contraction constraint in the definition of the optimization problem often leads to non-standard features, such as a need for the multi-step open-loop application of control sequences or the use of a multi-step memorization level. This can cause unfeasibility in the presence of uncertain disturbances. Thus, the author [Alamir 2017] proposed a new contraction based approach which does not involve the inclusion of contraction constraint.

A simple novel approach for stabilizing nonlinear systems using monotonically increasing weighting profiles without the addition of any stability-related cost or constraints was proposed in [Alamir 2018] and proven under some mild assumptions. For a given nonlinear system of the form $\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k))$, Lets denote the control profiles as $\mathbf{u} = [\mathbf{u}(0), \mathbf{u}(1), \dots, \mathbf{u}(N-1)]$ defined over a prediction horizon of length N and $\mathbf{x}^{\mathbf{u}}(k)$ to represent the corresponding state trajectory.

Based on the trajectories \mathbf{u} and $\mathbf{x}_{\mathbf{u}}$, the following cost function can be formulated.

$$J_m(\mathbf{u}) = \sum_{k=1}^N \left(\frac{k}{N}\right)^m l(\mathbf{x}_{\mathbf{u}}(\mathbf{x}(k))) \quad (3.11)$$

For some integer m , it proven that $\mathbf{x} = 0$ is an asymptotically stable equilibrium for the closed-loop system given by $\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}^*(k))$.

We wish to investigate this stability approach by applying it to the pursuit-evasion problem. The Pursuer's OCP in (3.5) can be reformulated to include the monotonically increasing function.

$$\min_{\mathbf{u}_p} J = \sum_{k=0}^{N-1} (k/N)^m \|\mathbf{x}_p(k) - \mathbf{x}_e(k)\|_Q^2 + \|\mathbf{u}_p(k)\|_R^2 \quad (3.12)$$

And the corresponding evader's OCP can be reformulated as

$$\max_{\mathbf{u}_e} J = \sum_{k=0}^{N-1} (k/N)^m \|\mathbf{x}_e(k) - \mathbf{x}_p(k)\|_Q^2 + \|\mathbf{u}_e(k)\|_R^2 \quad (3.13)$$

3.4 Numerical simulations

The game was set up and implemented in MATLAB/Simulink environment. The controllers were coded using CasADi software, an open-source symbolic framework for automatic differentiation and optimal control [Andersson et al. 2019]. The game is played in the presence and absence of an obstacle. In each game, the relative distance between the two players are computed at every time instant as in (3.2), while the capture time is when (3.4) is satisfied.

The diameter of each robot is $d_{rob} = 20cm$, the constraints on the pursuer’s control variables are $|v_p| = 0.4m/s$, $|\omega_p| = \pi/3$ while the constraints on the evader’s control variables are $|v_e| = 0.4m/s$, $|\omega_e| = \pi/4$. The prediction horizon $N = 10$ while the sampling time $T_s = 0.1$. The states constraints are the dimension of the polytope $10m$ by $10m$, therefore the constraints on both $x - axis$ and $y - axis$ from $-5m$ to $+5m$ and is applied to both players.

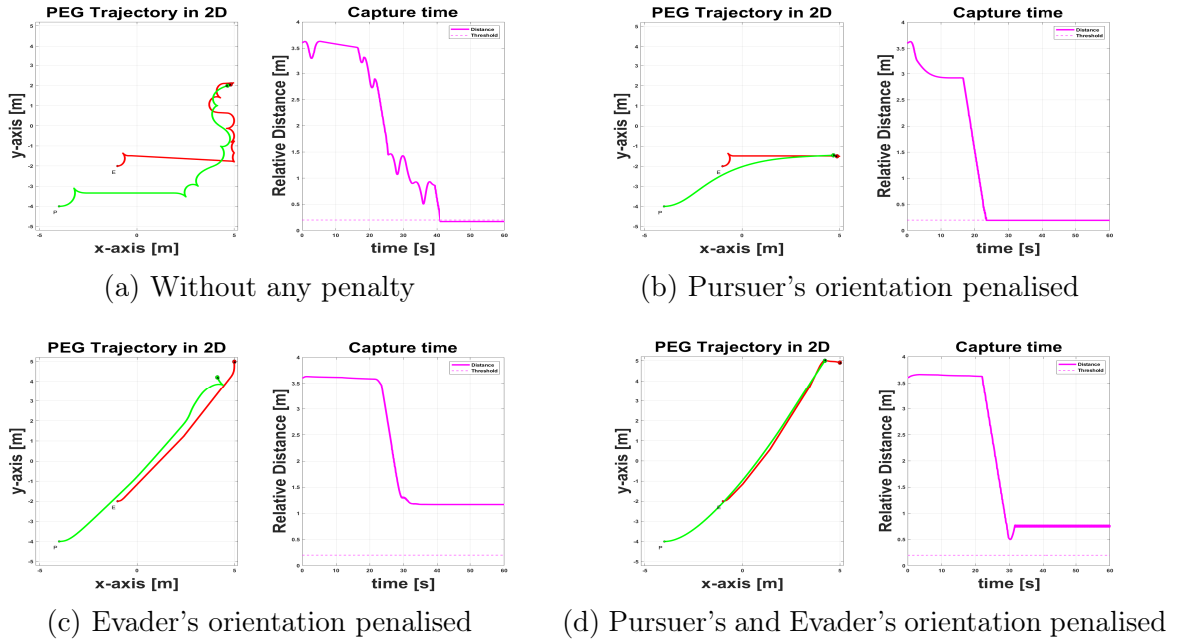


Figure 3.1: Penalizing the orientation angle in PEG

To show the effect of penalizing the weighting matrices in the NMPC formulation, the pursuit-evasion game was played without any stabilizing constraints, cost, or function in Figure 3.1, the letter ‘P’ represents the pursuer while the letter ‘E’ represents the evader. When the weighting matrices for each player was selected as an identity matrix ($R = diag([1; 1])$, $Q = diag([1; 1; 1])$), the game behave as shown in Figure 3.1a. It can be seen that both players respond to changes in the opponent’s orientation even if they are far apart. Therefore there is a need to reduce the weight on the orientation angle. The best value selected after many trails is ($R = diag([1; 1])$, $Q = diag([1; 1; 0.001])$). In Figure 3.1b, the pursuer’s orientation was penalized and it can be seen that the pursuer does not respond to slight changes in the evader’s orientation which explains why it captured the evader in a shorter time.

Conversely, in Figure 3.1c, only the evaders' orientation was penalized, thus this created an advantage over the pursuer that ultimately leads to non-capture in the game. When the orientation of both the pursuer and the evader is penalized, it can be observed in Figure 3.1d there is an improvement in the game trajectory.

3.4.1 PEG with stabilizing terminal cost function

The approach was improved by adding a stabilizing cost function to the NMPC formulation so as to enhance the pursuit-evasion game. The terminal penalty, ($Q_N = 1000 * Q$) was selected after several trails to stabilize the controllers. The effect of adding the stabilizing cost function with an appropriate penalty was shown in Figure 3.2.

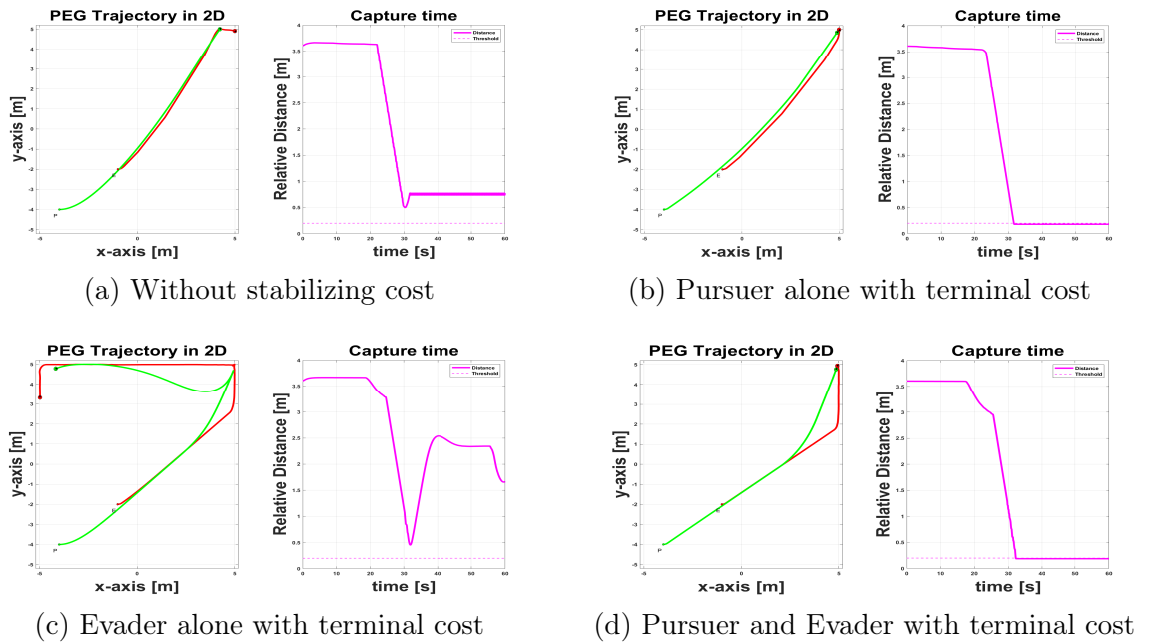


Figure 3.2: PEG with stabilizing terminal cost

Figure 3.2a depicts the game trajectory when no stabilizing cost was added to any player. However, when the terminal cost was added to the pursuer, Figure 3.2b shows the game trajectory and the subsequent capture of the evader infinite time. A contrary situation happened in Figure 3.2c where only the evader's NMPC included the stabilizing terminal cost function. It can be observed that the evader escaped capture by the pursuer. In Figure 3.2d, the controllers of both players included the stabilizing terminal cost function, and that the pursuer eventually captured the evader.

3.4.2 PEG with monotonically increasing weight profile

In this method, we exclude the stabilizing terminal cost function in our formulation but add monotonically increasing weighting profiles that depend on an integer m . The selected value of this parameter after many trials was $m = 3$. Thus for a prediction horizon of length $N = 10$, the stage cost function has been multiplied by a factor between 0 and 1. The result was for this

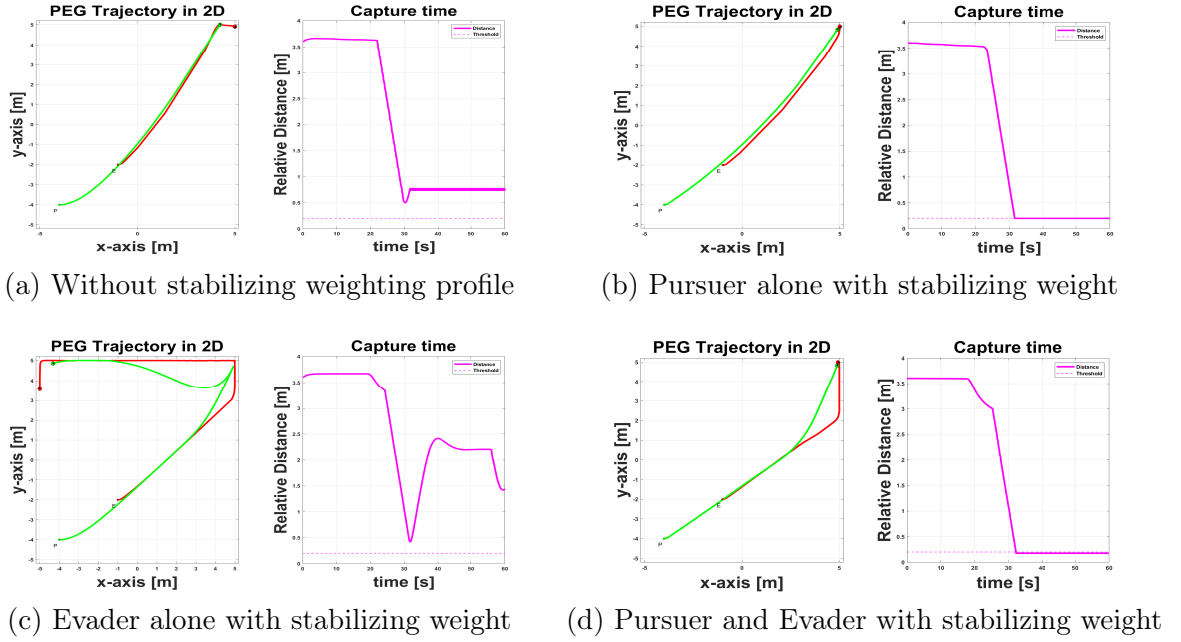


Figure 3.3: PEG with stabilizing weight

method is presented in Figure 3.3 which comprises the case in Figure 3.3a where no stabilizing cost or weight was added to the players. When the pursuer's controller was formulated to include the stabilizing weight, the game trajectory as shown in Figure 3.3b. The inference was the eventual capture of the evader, which does not include any stabilizing cost or weight, as in the previous case. Conversely, when only the evader's controller included the stabilizing weight, the result in Figure 3.3c shows opposite behaviors and there is no capture in finite time. Figure 3.3d shows the situation when the controllers of both players included the stabilizing weight.

3.4.3 Comparison of the performance of the two stabilizing methods

The performance of these two stabilizing methods is compared in terms of computation time, robustness to the presence of noise, and ability to avoid obstacles. The comparison is done from the perspective of the pursuer.

In Figure 3.4, the computation time spent by the pursuer's controller is presented when three games are played. As it can be seen, the average time spend when the pursuer employed

the stabilizing cost was found to be 0.019 secs while the time spent with the stabilizing weight method was 0.022 secs. The conventional methods of using terminal cost have outperformed the new method in this regard.

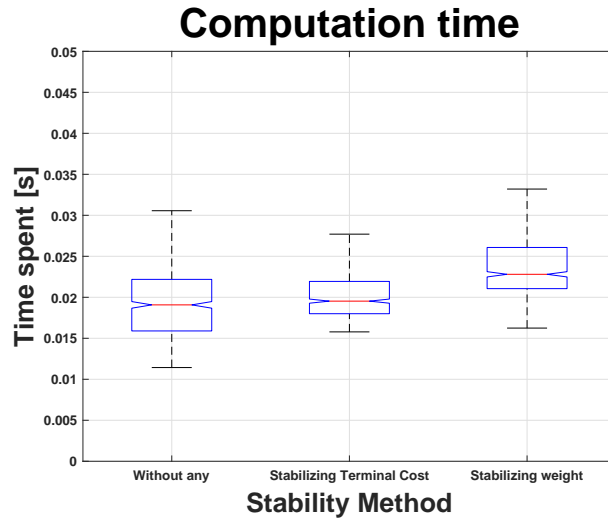
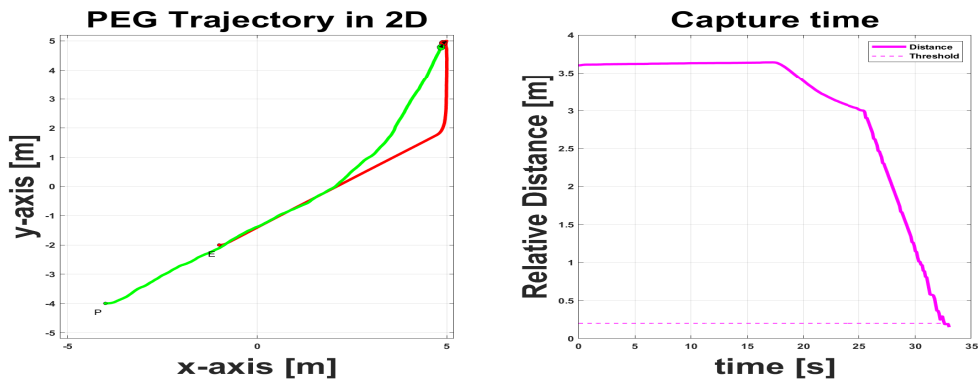


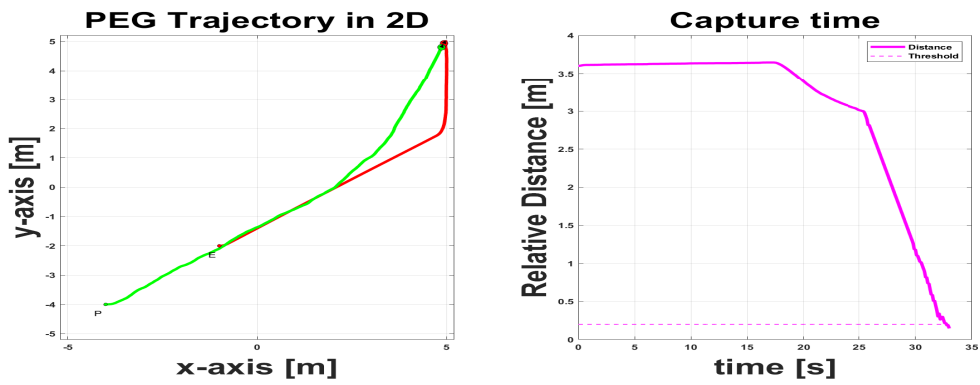
Figure 3.4: Computation time comparison

The game was played with the addition of some noise on the measurement and the result was depicted in Figure 3.5. Similar performance was observed as each method was seen to be robust to the measurement noise.

Finally, some obstacles were added to the game environment at strategic locations so that each player must encounter at least one obstacle during the game. Both methods are seen to be able to avoid obstacles in Figure 3.6.

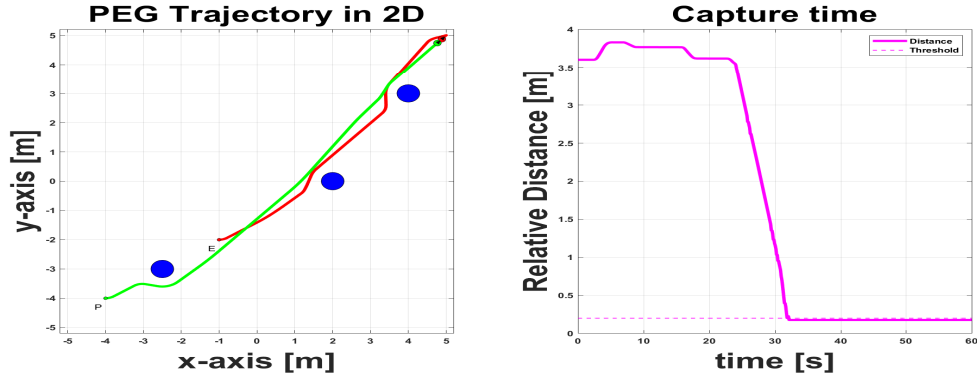


(a) Pursuer with stabilizing terminal cost

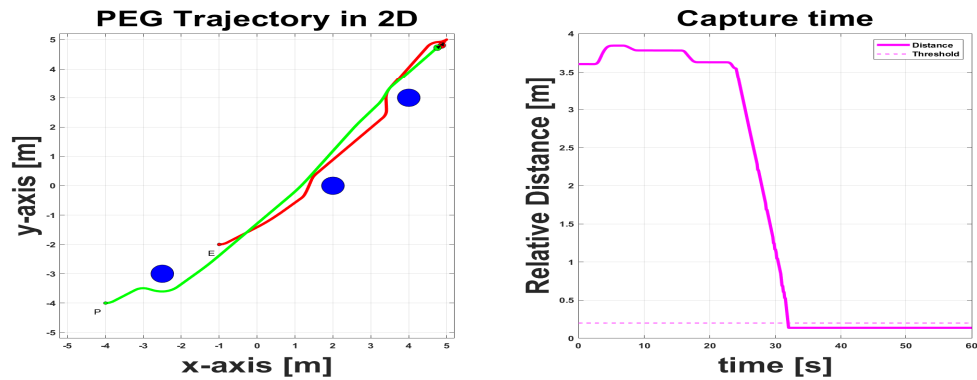


(b) Pursuer with stabilizing weight

Figure 3.5: Effect of noise of PEG using NMPC



(a) Pursuer with stabilizing terminal cost



(b) Pursuer with stabilizing weight

Figure 3.6: PEG with obstacles avoidance using NMPC

3.5 Conclusion

In this chapter, the work of the previous chapter which consists of control of a single robot to a desired set-points was extended. The desired set-point in this chapter is a moving object which is also controlled by an adversarial effect. The resulting pursuit-evasion game was solved using nonlinear model predictive control and applied on nonholonomic mobile robots for the first time. To stabilizing methods of the nonlinear model predictive control formulation, namely; stabilizing terminal cost and a monotonically increasing stabilizing weight are discussed and compared. The performance of the two stabilizing methods is compared in terms of the computation time, robustness to measurement noise, and avoidance of obstacles. It was shown using numerical simulations that the conventional method has outperformed the new method in terms of computational resources required to find an optimal control sequence. It can also be concluded that the new method can be implemented in pursuit-evasion problems with the nonlinear system. Finally, it can also be concluded that the pursuit-evasion game can be solved using the nonlinear model predictive control approach with good performance.

Pursuit-Evasion Games Using Game-theoretic Approaches

In this chapter, a variant of Isaacs' Homicidal Chauffeur problem was posed and solved using game-theoretic techniques where systems are modeled as intelligent rational decision-makers. Each player considers the best strategy of the opponent before deciding its strategy. The problem posed involved a pursuit-evasion game between two nonholonomic mobile robots with the same agility but different speed. The pursuer is said to be faster than the evader to terminate the game. Two game-theoretic solutions are proposed depending on the level of information on the opponent. The first method where each player has complete information on the opponent uses a double optimization approach to predict the opponent's best strategy in advance before computing its best strategy. Due to the high computation time required, the second method was proposed so that each player has incomplete information on the opponent. These game-theoretic approaches are compared with the state-based NMPC approach used in the previous chapter. The performance of the two methods is evaluated in terms of capture time rate, computation time through rigorous simulations in the presence and absence of obstacles.

4.1 Overview of game-theory

Game theory is a mathematical tool for studying strategic interactions among intelligent decision-making entities, known as players or agents, whose individual decisions jointly determine the overall outcome. In other words, game theory mathematically models situations of conflict and cooperation between intelligent rational entities. Due to numerous situations of interactions among decision entities in the real world, game theory found application in several fields such as social science, biology, economics, computer science, and engineering.

The game theory dates back to 1921, when Emile Borel, a French mathematician, envisioned it being used in military and economic fields. Borel's ultimate goal was to determine whether the best strategy for a given game existed and to find it, which led to the publication of several papers on "la Theorie du jeu". However, Borel did not pursue his ideas very far, which is why many historians regard John Von Neumann, who published his first paper in 1928, as the founder of game theory (seven years after Borel). Von Neumann's article "Theory of Parlor Games" made significant contributions to the mathematical respectability of game

theory [Maestre 2010]. The book "Theory of Games and Economic Behaviour" [Von Neumann and Morgenstern 1947], later became an influential and pioneering work of game theory to economics. To better understand the consequence of conflict and cooperation among players, let us recall a popular game in the social domain, "the prisoner's dilemma" which was presented in [Poundstone 1992].

Two criminal gang members are apprehended and imprisoned. Each prisoner is in solitary confinement and has no way of communicating or exchanging messages with the other. The cops admit they don't have enough evidence to convict the couple on the main charge. On a lesser charge, they intend to sentence both to a year in prison. At the same time, the police present each prisoner with a Faustian bargain. If he testifies against his partner, he will be released, while the partner will be sentenced to three years in prison on the main charge. Oh! but there's a catch... Both prisoners will be sentenced to two years in prison if they testify against each other.

Players (A and B)	A rejects the deal	A accepts the deal
B rejects the deal	1 year, 1 year	3 years, 0 year
B accepts the deal	0 year, 3 years	2 years, 2 years

Table 4.1: Prisoner's Dilemma

This game has four possible outcomes, as illustrated in the table 4.1. Each player has two options: accept or reject the offer. At first glance, it may appear that rejecting the deal for a one-year prison sentence is a good idea. The major issue, however, is the fear of serving the maximum sentence of three years in prison if the other player accepts the deal. Because the game is symmetric, only one of the two outcomes in the diagonal will be the game's outcome if both players are intelligent and rational.

4.1.1 Elements of game theory

Some terms acquire special meaning in the context of game theory, even if they are still close to their common meaning. A game can be modeled if the following elements are defined.

1. Players: these are individual agents or systems that participate in the game by making rational and/or intelligent decisions that can be accomplished by optimizing its objective function. The rationality feature motivates the player to always seek the best decision that maximizes profit or minimizes cost. As a result, it is important to recognize that individual rationality is not the same as group rationality. A finite set of players can be represented mathematically as:

$$\mathcal{I} = \{1, 2, \dots, I\} \tag{4.1}$$

2. Strategies are a player's entire plan of action for the game. This means that a strategy determines which action the player must take at any given decision point. A strategy

profile is a collection of possible strategies for a specific player. For each player, $i \in \mathcal{I}$, there is a strategy profile, \mathcal{U}_i . The strategy profile of other players, not i is denoted as \mathcal{U}_{-i} . Thus the joint strategy set for all players in the game can be represented as:

$$\mathcal{U} = (\mathcal{U}_i, \mathcal{U}_{-i}) \quad (4.2)$$

3. **Payoff:** This is the reward a player receives after executing (playing) a specific strategy. It is modeled as a utility function (to be maximized) or cost function (to be minimized) that reflects the player's preference for joint strategies.

The payoff of a player i can be represented in terms of utility function as:

$$\mathcal{J}_i = f(\mathcal{U}_i, \mathcal{U}_{-i}) \quad (4.3)$$

For any two joint strategy sets $\mathcal{U}, \mathcal{U}'$, a player i would strictly prefers \mathcal{U} instead of \mathcal{U}' if and only if

$$\mathcal{J}_i(\mathcal{U}) > \mathcal{J}_i(\mathcal{U}') \quad (4.4)$$

Also, the payoff of a player can be expressed in terms of cost function as:

$$J_i = f(\mathcal{U}_i, \mathcal{U}_{-i}) \quad (4.5)$$

A player i would strictly prefers \mathcal{U} instead of \mathcal{U}' if and only if

$$J_i(\mathcal{U}) < J_i(\mathcal{U}') \quad (4.6)$$

Hence a game can be described by either of the following tuples:

$$\mathcal{G} = \{\mathcal{I}, \mathcal{U}, J\} \quad (4.7a)$$

$$\mathcal{G} = \{\mathcal{I}, \mathcal{U}, \mathcal{J}\} \quad (4.7b)$$

4.2 Game theory and control

Although game theory has been studied primarily as a modeling paradigm in the mathematical and social sciences, there is a strong link to control systems in that a controller can be viewed as a decision-making entity. As a result, game theory is useful in situations involving multiple interacting controllers. This section provides an overview of two specific control theory topics in which game theory has played a significant role.

4.2.1 Cooperative games

Cooperative game theory studies situations of mutual interaction between a set of agents which can negotiate among themselves to follow common binding strategies. As a result of the bargaining process, the set of agents might be divided into several subsets that are called

a coalition. The role of game theory in this field is to investigate which coalitions of agents should be formed and to analyze how the cost or benefit of cooperation should be distributed among coalition members. It is worth noting that the existence of a communication channel is critical to this branch of game theory [Maestre 2010].

A cooperative game is defined in its most basic form by two elements: a set of different players and a function that assigns a value to each of the possible coalitions of players. At this point, it is worth noting that the coalition's value represents the cost of achieving the common goal without the assistance of agents who are not members of the coalition. The study of network influence in cooperative game theory began decades ago with the work of Myerson in [Myerson 1977]. Any two agents can communicate and thus cooperate if they are at least indirectly connected by the network, that is, if there is a path of active links connecting them. Furthermore, a cooperative game can account for the costs of communication. Therefore, it is reasonable to assume that the existence of each link has a fixed cost associated with its use.

4.2.2 Non-cooperative games

A non-cooperative game is one in which two individuals compete against each other. Unlike in cooperative games, the players cannot form a coalition or reach a binding agreement. It is also known as a zero-sum game because the benefit of one player is at the expense of the other. The simplest example of non-cooperative games from the control point of view is to regard the controller as a player and environmental uncertainty as another player since the driving motivation for feedback control system is to assure satisfactory performance of a controller in the face of uncertainties. Non-cooperative games can also be played as a two-player pursuit-evasion game where two players or teams play against each other in a conflicting scenario. Several examples and variants of pursuit-evasion differential games are outlined below:

4.2.2.1 Homicidal chauffeur problem

This is a classic 1-pursuer-1-evader problem that is used in military applications. It was proposed by Isaac in his seminal text [Isaacs 1956] where a hypothetical slow but highly maneuverable holonomic pedestrian is pitted against a driver of a faster but less maneuverable motor vehicle (Dubin's Car). The driver attempts to run over the pedestrian in this somewhat horrific scenario. The question to be answered is: Under what conditions, and with what strategy, can the driver of the car guarantee that he will always catch the pedestrian, or conversely, can the pedestrian guarantee that he will always elude the car, [Isaacs 1956]. And, if the pedestrian's capture is unavoidable, what is the chauffeur's optimal strategy for minimizing the pedestrian's time-to-capture, and what is the latter's strategy for maximizing his time.

Merz's Ph.D. thesis [Merz 1971] is a major work on the Homicidal Chauffeur Differential Game. The work gives great detail and insight into the problem and also proposes several optimal solutions to the Homicidal Chauffeur game. Together with his supervisor, Breakwell,

they presented a complete solution of the game at a conference in [Merz 1974]. Similarly, the Homicidal Chauffeur probe was described and solved using Pontryagin's Minimum Principle in [Marchal 1975].

4.2.2.2 The two car differential games

The "Two Cars" Differential Game is a variation of the Homicidal Chauffeur Differential Game, in which two players compete in a pursuit-evasion game, each driving a car with the smallest turning radius. The author [Meier 1969], studied the problem of two cars in his early work, in which both players had the same minimum turn radius, the pursuer was slower than the evader, and the capture was defined by coming inside the Evader's range. In [Getz and Pachter 1981c; Getz and Pachter 1981b], another investigation into the Two-Car problem was conducted in which capture, escape, and the barrier surfaces between those regions were discussed. Both agents in [Getz and Pachter 1981c] have sector-based capture regions, which is typical of an aerial dogfight; however, in [Getz and Pachter 1981b], the capture regions were different, describing a heterogeneous model of onboard weapon systems. Similarly, the author in [Greenfeld 1987] investigated the two-car problem, endowing the pursuer with a surveillance capability of range, l . The goal was to escape the surveillance region in the shortest amount of time. In [Lewin and Breakwell 1975], a similar problem known as the "Surveillance-Evasion" Differential Game was investigated where the evader wants to get out of the pursuer's detection circle as soon as possible, whereas the pursuer wants the opposite. A complete analysis performed in [Bera, Makkapati, and Kothari 2017] goes into detail of games of kind and degree, with studies on different agent's speeds, capture radius, and maneuverability constraints where a three-dimensional plot of the state space, highlighting the barrier and switching surfaces for the various scenarios was created.

4.2.2.3 Pursuit-evasion game in a constrained environment

To consider differential games more realistically, the introduction of state boundaries and constraints allows for finite spaces and regions to be included in the formulation of the game. The pursuit-evasion differential game can be restricted to a bounded area or obstacles can be used by imposing limitations on physical states. A two-player differential pursuit-evasion game in which an obstacle is added to delay the pursuer or prevent capture entirely was proposed in [Fisac and Sastry 2015]. Also, [Oyler 2016] considered pursuit-evasion games in the presence of obstacles in the form of polygons, line segments, and asymmetric objects that impede the movement of the players. The authors in [Fuchs and Khargonekar 2016] motivated the use of escort regions by manipulating the performance functional to handle pursuer evasion games in the presence of obstacles, while in [Dong, Zhang, and Jia 2012] pursuit-evasion games with obstacles avoidance was handled using artificial potential field methods.

4.2.2.4 Pursuit-evasion with incomplete information

When one or more agents do not have complete information about the state of the game, the problem is referred to as "Differential Games of Incomplete Information." Isaacs stated in his seminal work [Isaacs 1956] that "the ability to pose problems that limited information to individual players appears to be the most vital area for future research." A stochastic approach for modeling partial information that one agent may have relative to another was proposed in [Roxin and Tsokos 1970]. A differential game in which one of the agents is given incomplete information was proposed in [Chernousko and Melikyan 1975] to account for information delays or gaps in gameplay. The authors in [Yavin 1986] proposed an incomplete information pursuit-evasion differential game in which the pursuer's bearing information was limited while the evader had perfect information in the game. In [Giovannangeli et al. 2010] pursuit problem in the presence of convex obstacles was solved using Apollonius circles to provide paths in which the pursuer's visibility of the evader is guaranteed throughout the engagement. A problem where a parameter is unavailable to only one player at the start of the game, and the other has a probability density function describing that parameter was considered in [Hexner 1979]. [Pachter and Yavin 1981] investigated the effects of noise on the Homicidal Chauffeur problem by incorporating stochastics into the pursuit-evasion differential game dynamics. [Basimanebotlhe and Xue 2014] investigated a differential game with a nonlinear stochastic equation in which two players are subjected to noisy measurements.

4.2.2.5 Aerial pursuit-evasion problems

The use of pursuit-evasion differential games in tactical air-to-air applications has been investigated. In [Shinar and Gutman 1979] a closed-form solution to a three-dimensional missile-aircraft pursuit-evasion game was presented. The same authors also used variational methods to investigate a realistic pursuit-evasion engagement involving a missile engaged on an aircraft and air-to-air scenarios [Shinar 1980]. In terms of naval applications, the two-player engagement was framed as a homicidal chauffeur differential game that takes ship dynamics into account in [Pachter 1987]. A realistic 3-dimensional differential game by modeling fighter aircraft was created in [Greenwood 1992] where barrier analysis was included in the dynamics of two fighter aircraft in space, as well as firing envelopes. In [Imado and Kuroda 2005] a differential game involving a pursuit-evasion engagement between a missile and an aircraft was proposed such that a nonlinear miss-distance was used as a payoff functional in the game formulation. The authors in [Shinar, Glizer, and Turetsky 2009b] investigated a pursuit-evasion game in which the dynamics of the pursuer can be changed a finite number of times during the pursuit. In [Shinar, Glizer, and Turetsky 2009a], the evader has a limited number of times during the engagement to change their dynamics. The problem of pursuit or evasion selection when both agents had capture sets but no prior assignment was implemented was investigated in [Merz 1985]. The author's interest in dogfights and aerial combat was in role assignment in pursuit-evasion differential games, as well as the outcome.

4.2.2.6 Other pursuit evasion games with two players

The work in [Leitmann 1968] is another example of posing the pursuit-evasion problems using simple motion kinematics in a differential game. A simple differential game between a pursuer and an evader was proposed in this paper, and variational techniques were used to determine game outcomes where terminal miss distance was used as the payoff/cost functional. The authors in [Calise and Yu 1985] developed a game involving the pursuit-evasion of two aircraft at medium to long-range using simple motion kinematics and expanded control energy. The authors can find trajectories similar to the minimum time intercept using only four states to model the encounter using a reduced-order model with control energy. "Lion and the Man" differential game is a pursuit-evasion differential game in which the lion pursues a man proposed in [Meyers 2012]. The lion and the man are free to change their velocity direction at any time, but the intensity with which they do so is limited. Because the lion is faster than man, the areas of escape and capture are of interest and numerically determined.

4.3 Game-theoretic approaches for solving pursuit-evasion problems

Von Neumann and Morgenstern's works were limited to two-player zero-sum games until a famous US mathematician, John F. Nash, extended the works to N-players in [Nash 1950]. The game-theoretic solution is the Nash equilibrium which is a set of strategy pairs where no player has an incentive to deviate. Thus, it is regarded as the best response by each player.

Since game theory modeled systems as an intelligent rational decision-maker where an agent considers the opponent's strategy before deciding his strategy, the predicted opponent's best response is the worst case from the agent's point of view. In this thesis, we considered the two-player pursuit-evasion problem between two autonomous ground vehicles of nonholonomic type. The goal of the pursuer is to capture the evader while the evader is doing the opposite. The game would be played in a bounded area in the presence and absence of obstacles.

In [Marden and Shamma 2018], pursuit-evasion game was formulated as a zero-sum game with two players, the pursuer and the evader in which the strategy sets $[\mathcal{U}_p, \mathcal{U}_e]$ and the cost functions $[J_p, J_e]$ satisfy the zero-sum property.

$$J_p(U_p, U_e) + J_e(U_p, U_e) = 0 \quad \forall U_p \in \mathcal{U}_p, U_e \in \mathcal{U}_e \quad (4.8)$$

This property depicts the situation in which an increase in the cost of one player results in a corresponding decrease in the cost of the other player. Because of this unique structure, zero-sum games are commonly expressed in terms of a single objective function $\phi(U_p, U_e)$, that is the cost function of the pursuer (the minimizer) and the cost function of the evader (the maximizer).

$$\phi(U_p, U_e) = J_p(U_p, U_e) = -J_e(U_p, U_e) \quad (4.9)$$

Two quantities can be defined:

$$\underline{\phi} = \min_{U_p} \max_{U_e} \phi(U_p, U_e) \quad (4.10a)$$

$$\bar{\phi} = \max_{U_e} \min_{U_p} \phi(U_p, U_e) \quad (4.10b)$$

The quantity $\underline{\phi}$ represents the best guaranteed cost for the pursuer in the worst case scenario of its strategy, U_p , being known to the evader, while the quantity $\bar{\phi}$ has an analogous interpretation for the evader. Zero-sum game have a value:

$$\phi^* = \underline{\phi} = \bar{\phi} \quad (4.11)$$

The Nash equilibrium of a zero-sum game is characterized by the following saddle-point condition. The pair of strategies U_p^* and U_e^* is a Nash equilibrium if:

$$\phi(U_p^*, U_e') \leq \phi(U_p^*, U_e^*) \leq \phi(U_p', U_e^*) \quad \forall U_p' \in \mathcal{U}_p, U_e' \in \mathcal{U}_e \quad (4.12)$$

The inequality indicates that U_p^* is the best response for the pursuer to U_e^* and vice versa. If a zero-sum game has a Nash equilibrium, then it has a value:

$$\phi^* = \phi(U_p^*, U_e^*) \quad (4.13)$$

In this chapter, three game-theoretic algorithms (GTA) are proposed to solve two payer pursuit-evasion game problem.

4.3.1 Game-theoretic algorithm with complete information (GTA-1)

This method solves the pursuit-evasion problem in the context of game theory where each player considers the optimal strategy of the opponent based on the assumption that the player knows full information about the opponent. The solution was based on double optimization such that each player must first compute the worst-case strategy of the opponent, then compute its own strategy based on it. The solution was formulated using two NMPC solving point stabilization problems. The opponent optimal move predicted would be taken as a reference by each player. The computation times for each MPC block are summed at each instant.

4.3.1.1 Pursuer's game-theoretic algorithm (GTA-1)

The pursuer's controller is obtained by solving its min-max problem. This could be achieved in two steps:

Step 1: Computing the evader's best move \mathbf{x}_e^* by solving the following cost function:

$$\max_{\mathbf{u}_e} J = \|\mathbf{x}_e(N) - \mathbf{x}_p\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_e(k) - \mathbf{x}_p(k)\|_Q^2 + \|\mathbf{u}_e(k)\|_R^2 \quad (4.14)$$

subject to:

$$\begin{cases} \mathbf{x}_e(k+1) = f_e(\mathbf{x}_e(k), \mathbf{u}_e(k)), & k = 0, 1, \dots, N-1 \\ \sqrt{(x_e(k) - x_{obs})^2 + (y_e(k) - y_{obs})^2} \geq (R_e + R_{obs}) \\ \mathbf{x}_{e_{min}} \leq \mathbf{x}_e(k) \leq \mathbf{x}_{e_{max}} \\ \mathbf{u}_{e_{min}} \leq \mathbf{u}_e(k) \leq \mathbf{u}_{e_{max}} \end{cases} \quad (4.15)$$

The output of this block is a vector of predicted states of the evader \mathbf{x}_e^* , which is taken as a reference.

Step 2: Compute the pursuer's strategy by the solving the following cost function:

$$\min_{\mathbf{u}_p} J = \|\mathbf{x}_p(N) - \mathbf{x}_e^*\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_p(k) - \mathbf{x}_e^*\|_Q^2 + \|\mathbf{u}_p(k)\|_R^2 \quad (4.16)$$

subject to:

$$\begin{cases} \mathbf{x}_p(k+1) = f_p(\mathbf{x}_p(k), \mathbf{u}_p(k)), & k = 0, 1, \dots, N-1 \\ \sqrt{(x_p(k) - x_{obs})^2 + (y_p(k) - y_{obs})^2} \geq (R_p + R_{obs}) \\ \mathbf{x}_{p_{min}} \leq \mathbf{x}_p(k) \leq \mathbf{x}_{p_{max}} \\ \mathbf{u}_{p_{min}} \leq \mathbf{u}_p(k) \leq \mathbf{u}_{p_{max}} \end{cases} \quad (4.17)$$

The output of this block is the pursuer's Nash equilibrium strategy, i.e. $\mathbf{u}_p^* = [v; \omega]^T$ which stands for the linear and angular speeds respectively.

4.3.1.2 Evader's game-theoretic algorithm (GTA-1)

In this case the evader first predict the optimal strategy and next move of the pursuer by solving minimization problem and then use the first block outputs to compute its optimal strategy by solving maximization cost function with its dynamics. Similarly, the solution is in two steps:

Step 1: Predict pursuer's next move by solving the following cost function:

$$\min_{\mathbf{u}_p} J = \|\mathbf{x}_p(N) - \mathbf{x}_e\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_p(k) - \mathbf{x}_e\|_Q^2 + \|\mathbf{u}_p(k)\|_R^2 \quad (4.18)$$

subject to:

$$\begin{cases} \mathbf{x}_p(k+1) = f_p(\mathbf{x}_p(k), \mathbf{u}_p(k)), & k = 0, 1, \dots, N-1 \\ \sqrt{(x_p(k) - x_{obs})^2 + (y_p(k) - y_{obs})^2} \geq (R_p + R_{obs}) \\ \mathbf{x}_{p_{min}} \leq \mathbf{x}_p(k+1) \leq \mathbf{x}_{p_{max}} \\ \mathbf{u}_{p_{min}} \leq \mathbf{u}_p(k+1) \leq \mathbf{u}_{p_{max}} \end{cases} \quad (4.19)$$

The output of this block is a vector of predicted states of the pursuer, \mathbf{x}_p^* which is taken as a reference.

Step 2: Then compute the evader's strategy by the solving the following cost function:

$$\max_{\mathbf{u}_e} J = \|\mathbf{x}_e(N) - \mathbf{x}_p^*\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_e(k) - \mathbf{x}_p^*\|_Q^2 + \|\mathbf{x}_e(k)\|_R^2 \quad (4.20)$$

subject to:

$$\begin{cases} \mathbf{x}_e(k+1) = f_e(\mathbf{x}_e(k), \mathbf{u}_e(k)), & k = 0, 1, \dots, N-1 \\ \sqrt{(x_e(k) - x_{obs})^2 + (y_e(k) - y_{obs})^2} \geq (R_e + R_{obs}) \\ \mathbf{x}_{e_{min}} \leq \mathbf{x}_e(k) \leq \mathbf{x}_{e_{max}} \\ \mathbf{u}_{e_{min}} \leq \mathbf{u}_e(k) \leq \mathbf{u}_{e_{max}} \end{cases} \quad (4.21)$$

The output of this block is the evader's Nash equilibrium strategy, i.e. $\mathbf{u}_e^* = [v; \omega]^T$ which stands for the linear and angular speeds respectively.

4.3.2 Improved game-theoretic algorithm with complete information (GTA-2)

In this algorithm, the problem was formulated as a Model Predictive Tracking problem instead of a point stabilization problem. Each player firstly predicts a sequence of the opponent's state and control trajectories over a finite horizon window using a multiple shooting procedure and then uses them as a state and control reference.

4.3.2.1 Pursuer's game-theoretic algorithm (GTA-2)

Step 1: Predict the evader's next move by solving the following cost function:

$$\max_{\mathbf{u}_e} J = \|\mathbf{x}_e(N) - \mathbf{x}_p\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_e(k) - \mathbf{x}_p\|_Q^2 + \|\mathbf{u}_e(k)\|_R^2 \quad (4.22)$$

subject to:

$$\begin{cases} X_e(k+1) = f_e(X_e(k), U_e(k)), & k = 0, 1, \dots, N-1 \\ \sqrt{(x_e(k) - x_{obs})^2 + (y_e(k) - y_{obs})^2} \geq (R_e + R_{obs}) \\ \mathbf{x}_{e_{min}} \leq \mathbf{x}_e(k) \leq \mathbf{x}_{e_{max}} \\ \mathbf{u}_{e_{min}} \leq \mathbf{u}_e(k) \leq \mathbf{u}_{e_{max}} \end{cases} \quad (4.23)$$

The output of this block is a vector of predicted sequence of optimal states and controls of the evader $[\mathbf{x}_e^*(k); \mathbf{u}_e^*(k)]$.

Step 2: Compute the pursuer's strategy by the solving the following tracking problem:

$$\min_{\mathbf{u}_p} J = \|\mathbf{x}_p(N) - \mathbf{x}_e^*(N)\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_p(k) - \mathbf{x}_e^*(k)\|_Q^2 + \|\mathbf{u}_p(k) - \mathbf{u}_e^*(k)\|_R^2 \quad (4.24)$$

subject to:

$$\begin{cases} \mathbf{x}_p(k+1) = f_p(\mathbf{x}_p(k), \mathbf{u}_p(k)), & k = 0, 1, \dots, N-1 \\ \sqrt{(x_p(k) - x_{obs})^2 + (y_p(k) - y_{obs})^2} \geq (R_p + R_{obs}) \\ \mathbf{x}_{p_{min}} \leq \mathbf{x}_p(k) \leq \mathbf{x}_{p_{max}} \\ \mathbf{u}_{p_{min}} \leq \mathbf{u}_p(k+1) \leq \mathbf{u}_{p_{max}} \end{cases} \quad (4.25)$$

The output of this block is the pursuer's Nash equilibrium strategy, i.e. $\mathbf{u}_p^* = [v; \omega]^T$ which stands for the linear and angular speeds respectively.

4.3.2.2 Evader's game-theoretic algorithm (GTA-2)

Similarly, full information of the pursuer is assumed and the solution is in two steps:

Step 1: Predict pursuer's next move by solving the following cost function:

$$\min_{\mathbf{u}_p} J = \|\mathbf{x}_p(N) - \mathbf{x}_e\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_p(k) - \mathbf{x}_e\|_Q^2 + \|\mathbf{u}_p(k)\|_R^2 \quad (4.26)$$

subject to:

$$\begin{cases} \mathbf{x}_p(k+1) = f_p(\mathbf{x}_p(k), \mathbf{u}_p(k)), & k = 0, 1, \dots, N-1 \\ \sqrt{(x_p(k) - x_{obs})^2 + (y_p(k) - y_{obs})^2} \geq (R_p + R_{obs}) \\ \mathbf{x}_{p_{min}} \leq \mathbf{x}_p(k) \leq \mathbf{x}_{p_{max}} \\ \mathbf{u}_{p_{min}} \leq \mathbf{u}_p(k) \leq \mathbf{u}_{p_{max}} \end{cases} \quad (4.27)$$

The output of this block is a vector of predicted sequence of optimal states and controls of the pursuer $[\mathbf{x}_p^*(k); \mathbf{u}_p^*(k)]$.

Step 2: Then compute the evader's strategy by the solving the following tracking problem:

$$\max_{\mathbf{u}_e} J = \|\mathbf{x}_e(N) - \mathbf{x}_p^*(N)\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_e(k) - \mathbf{x}_p^*(k)\|_Q^2 + \|\mathbf{u}_e(k) - \mathbf{u}_p^*(k)\|_R^2 \quad (4.28)$$

subject to:

$$\begin{cases} \mathbf{x}_e(k+1) = f_e(\mathbf{x}_e(k), \mathbf{u}_e(k)), & k = 0, 1, \dots, N-1 \\ \sqrt{(x_e(k) - x_{obs})^2 + (y_e(k) - y_{obs})^2} \geq (R_e + R_{obs}) \\ \mathbf{x}_{e_{min}} \leq \mathbf{x}_e(k) \leq \mathbf{x}_{e_{max}} \\ \mathbf{u}_{e_{min}} \leq \mathbf{u}_e(k) \leq \mathbf{u}_{e_{max}} \end{cases} \quad (4.29)$$

The output of this block is the evader's Nash equilibrium strategy, i.e. $\mathbf{u}_e^* = [v; \omega]^T$ which stands for the linear and angular speeds respectively.

4.3.3 Game-theoretic algorithm with incomplete information (GTA-3)

Due to the complications arising from the double optimizations in terms of huge computation time and unrealistic assumption of knowing full information on the opponent, we investigated a situation when each player can only measure the current position (incomplete states) of the opponent. We proposed a limited information model predictive control to tackle the problem. To use a deterministic formulation of a discrete MPC, full states of the game are required. However, the opponent's heading angle is unavailable since each player can only measure the opponent's position. Therefore the heading angle of the opponent needs to be predicted by each player.

4.3.3.1 Prediction of opponent's heading angle

To obtain the heading angle of the opponent of each player, which is not available a priori, we can use the current position of the players to compute the optimal heading of the opponent. For this, we employ the optimal strategies for computing the heading angle of a player as presented in [Kothari, Manathara, and Postlethwaite 2014; Garcia et al. 2017]. The method proposes pursuit and evasion strategies for holonomic systems where the heading angle is considered as a control variable. The paper uses two different approaches to solve a cooperative pursuit problem with two or more players cooperating to capture an evader. The former uses the concept of Voronoi partitions while the latter uses a geometric approach.

The methods consider a game with two pursuers (P1 and P2) and one evader (E), either P1 or P2 alone can engage the evader if it is guaranteed to capture the evader faster than the other. The optimal heading angle of the pursuer (optimal pursuit strategy) and the optimal heading angle of the evader are derived as:

$$\theta_i^* = \arctan\left(\frac{y_e - y_i}{x_e - x_i}\right) \quad (4.30a)$$

$$\theta_e^* = \pi + \arctan\left(\frac{y_e - y_i}{x_e - x_i}\right) \quad (4.30b)$$

In a situation where none of the two pursuers can capture the evader faster than the other, cooperative strategies are proposed. The center of the shared boundary (y_{ci}, x_{ci}) can be obtained either by finding the point of intersection of two Apollonian circles as described in [Garcia et al. 2017] or using a Voronoi partition as described in [Kothari, Manathara, and Postlethwaite 2014].

The optimal heading angle of the pursuer (optimal pursuit strategy) and the optimal heading angle of the evader are derived as:

$$\theta_i^* = \arctan\left(\frac{y_{ci} - y_i}{x_{ci} - x_i}\right) \quad (4.31a)$$

$$\theta_e^* = \pi + \arctan\left(\frac{y_{ci} - y_e}{x_{ci} - x_e}\right) \quad (4.31b)$$

4.3.3.2 Pursuer's game-theoretic algorithm (GTA-3)

The pursuer's controller can be designed to solve the pursuit problem using NMPC techniques. Since the states of the pursuer and the position of the evader can be measured at every decision instant, the pursuer can predict the optimal heading angle of the evader.

$$\theta_e^* = \pi + \arctan\left(\frac{y_e - y_p}{x_e - x_p}\right) \quad (4.32)$$

The optimal control inputs can be obtained by solving the following minimization cost function:

$$\min_{\mathbf{u}_p} J = \|\mathbf{x}_p(N) - \mathbf{x}_e^*\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_p(k) - \mathbf{x}_e^*\|_Q^2 + \|\mathbf{x}_p(k)\|_R^2 \quad (4.33)$$

Subject to:

$$\mathbf{x}_p(k+1) = f_p(\mathbf{x}_p(k), \mathbf{u}_p(k)), \quad k = 0, 1, \dots, N \quad (4.34a)$$

$$\sqrt{(x_p(k) - x_{obs})^2 + (y_p(k) - y_{obs})^2} \geq (R_p + R_{obs}) \quad (4.34b)$$

$$\mathbf{x}_{p_{min}} \leq \mathbf{x}_p(k) \leq \mathbf{x}_{p_{max}} \quad (4.34c)$$

$$\mathbf{u}_{p_{min}} \leq \mathbf{u}_p(k) \leq \mathbf{u}_{p_{max}} \quad (4.34d)$$

where \mathbf{x}_e^* is a vector that concatenates the evader's measured position and its predicted orientation that is computed by the pursuer using (4.32). The first segment of the cost function (4.33) was added to stabilize the controller by tuning the weighting matrix Q_N . The constraint (4.34a) is the pursuer's predicted trajectory over the prediction horizon, (4.34b) is the obstacle avoidance constraint which depends on the position and radius of both the pursuer and the obstacle, respectively, while (4.34c) and (4.34d) are the pursuer's states and input constraints, respectively.

4.3.3.3 Evader's game-theoretic algorithm (GTA-3)

Conversely, the evader's control strategy can be obtained by firstly predicting the pursuer's optimal heading angle.

$$\theta_p^* = \arctan\left(\frac{y_p - y_e}{x_p - x_e}\right) \quad (4.35)$$

Then compute the optimal control inputs by solving the following maximization cost function:

$$\max_{\mathbf{u}_e} J = \|\mathbf{x}_e(N) - \mathbf{x}_p\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_e(k) - \mathbf{x}_p\|_Q^2 + \|\mathbf{x}_e(k)\|_R^2 \quad (4.36)$$

Subject to:

$$\mathbf{x}_e(k+1) = f_e(\mathbf{x}_e(k), \mathbf{u}_e(k)), \quad k = 0, 1, \dots, N \quad (4.37a)$$

$$\sqrt{(x_e(k) - x_{obs})^2 + (y_e(k) - y_{obs})^2} \geq (R_e + R_{obs}) \quad (4.37b)$$

$$\mathbf{x}_{e_{min}} \leq \mathbf{x}_e(k) \leq \mathbf{x}_{e_{max}} \quad (4.37c)$$

$$\mathbf{u}_{e_{min}} \leq \mathbf{u}_e(k) \leq \mathbf{u}_{e_{max}} \quad (4.37d)$$

where \mathbf{x}_p^* is a vector that concatenates the pursuer's measured position and its predicted orientation that is computed by the evader using (4.35). Similarly, the first part of the cost function (4.36) was added to stabilize the controller by tuning the weighting matrix Q_N . The constraint (4.37a) is the evader's predicted trajectory over the prediction horizon, (4.37b) is the obstacle avoidance constraint which depends on the position and radius of both the evader and the obstacle, respectively while (4.37c) and (4.37d) are the evader's states and input constraints, respectively.

4.4 Numerical simulations

The game was set up and implemented in MATLAB/Simulink environment. The controllers were coded using CasADi software, an open-source symbolic framework for automatic differentiation and optimal control [Andersson et al. 2019]. The performance of the developed game-theoretic algorithms are presented in the following sections.

4.4.1 Simulation results of PEG using GTA-1

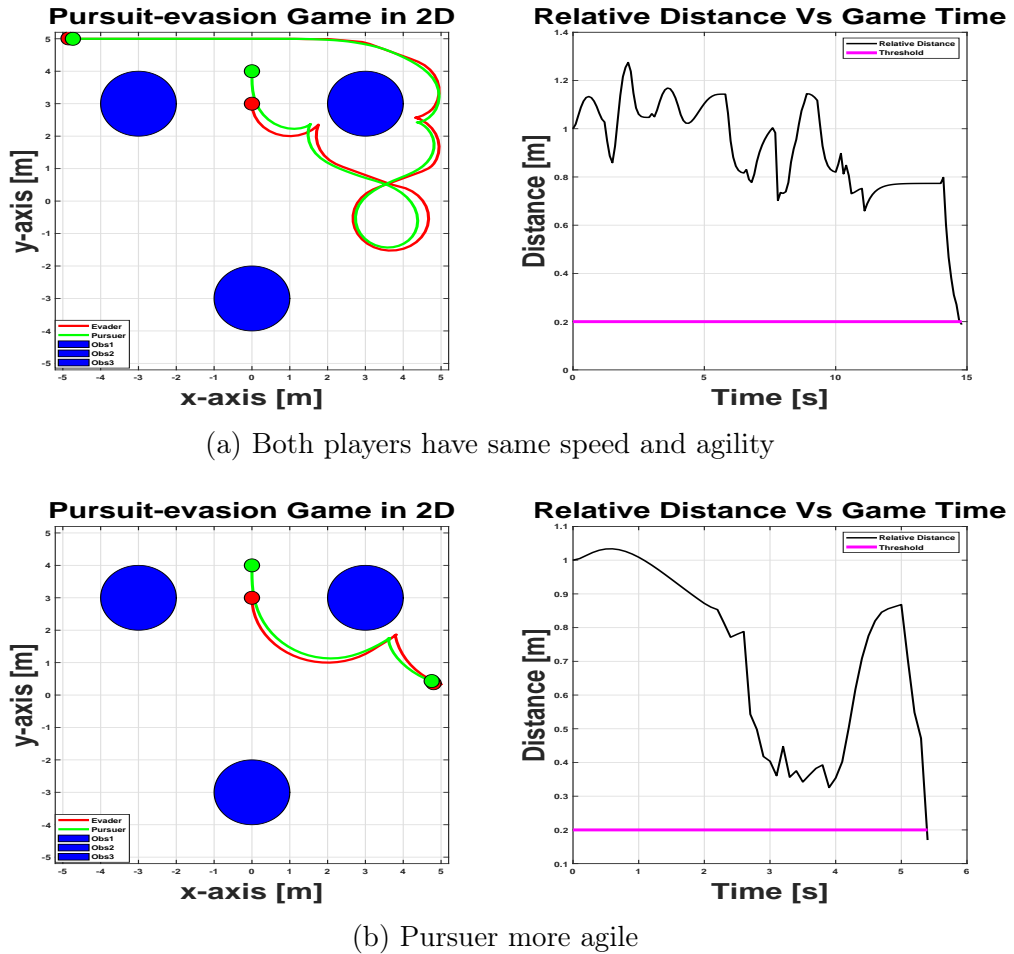


Figure 4.1: Comparing agility effect in pursuit-evasion game

As detailed in section (4.3.1), each player predicts a single optimal step of its opponent before computing its optimal strategy. The prediction horizon of length $N = 10$ was selected after tuning by trial and error, while the sampling time $T_s = 0.1$ to be coherent with the previous chapters. The states constraints are the dimension of the polytope $10m$ by $10m$, therefore the constraints on both x -axis and y -axis from $-5m$ to $+5m$ and is applied to both players. Actuator saturation produces constraints on the control inputs. For nonholonomic

mobile robots, the speed and agility of the robot are proportional to the rotational speed of the left and right wheel.

Figure 4.1 presents some of the preliminary results obtained with this method where the game was played in the presence of three obstacles. In figure 4.1a, both the pursuer and the evader have the same speed and agility while in figure 4.1b the pursuer is more agile. Due to the agility difference between the two figures, the pursuer takes 14.8 seconds to capture the evader in the first case and only 5.4 seconds in the second case. However, the weighting matrices of the controllers that produce these results are not tuned.

When the weighting matrices of the controllers are tuned, the results of the GTA-1 are compared with an NMPC strategy proposed in the previous chapter where each player makes use of the current states of the opponent that is developed in section (3.3). The pursuer has higher speed and maneuverability for terminating the game in a finite time. The constraints on the pursuer's control variable are $|v_p| = 1m/s$, $|\omega_p| = \pi/3$ while the constraints on the evader's control variables are $|v_e| = 0.6m/s$, $|\omega_e| = \pi/4$.

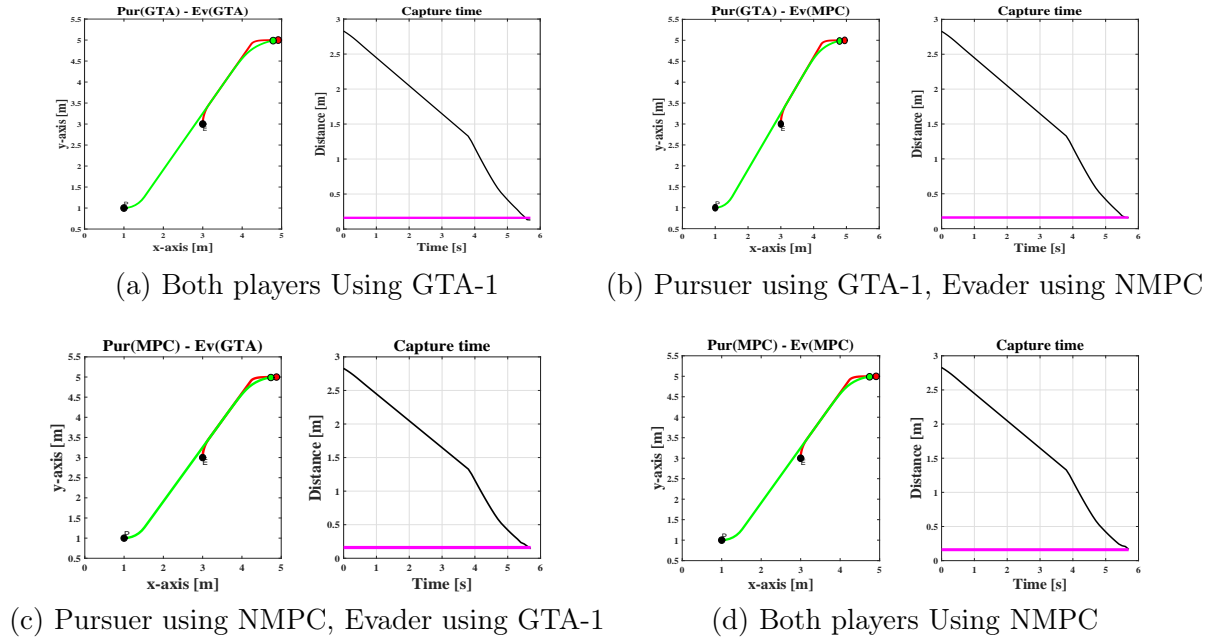


Figure 4.2: PEG with GTA-1 and NMPC in an obstacles free environment, the pursuer marked 'P' is green while the evader marked 'E' is red

In Figure 4.2, PEG is played in a bounded environment that is free of obstacles, with the initial conditions of the pursuer as $(1, 1, 0)$ while the initial condition of the evader is $(3, 3, \pi/2)$. These initial conditions are used for all the games in this scenario. In all the games, the left figure is the trajectory of the game while the figure at the right is the game pay-off. The capture time in all the cases is obtained at the point of intersection of the relative distance $r_{r,d}$ between the two players and the constant threshold distance (sum of the radii of the two robots) $r_{th} = 0.16$. Both players employed GTA-1 in 4.2a, the capture time is 5.5 secs. Despite

that the pursuer deviates from Nash equilibrium by using the NMPC algorithm in 4.2b, the capture time is 5.5secs still. In Figure 4.2c, the evader deviated from the Nash equilibrium and used the NMPC algorithm, but still, the capture time is 5.5 secs. Also in the figure 4.2d, both payers deviated from the Nash equilibrium by employing the NMPC algorithm and still, the capture time is 5.5 secs. Several simulations were conducted using several initial conditions but with the pursuer behind the evader in each case. In all the simulations conducted, the capture time used to be approximately the same for all four games.

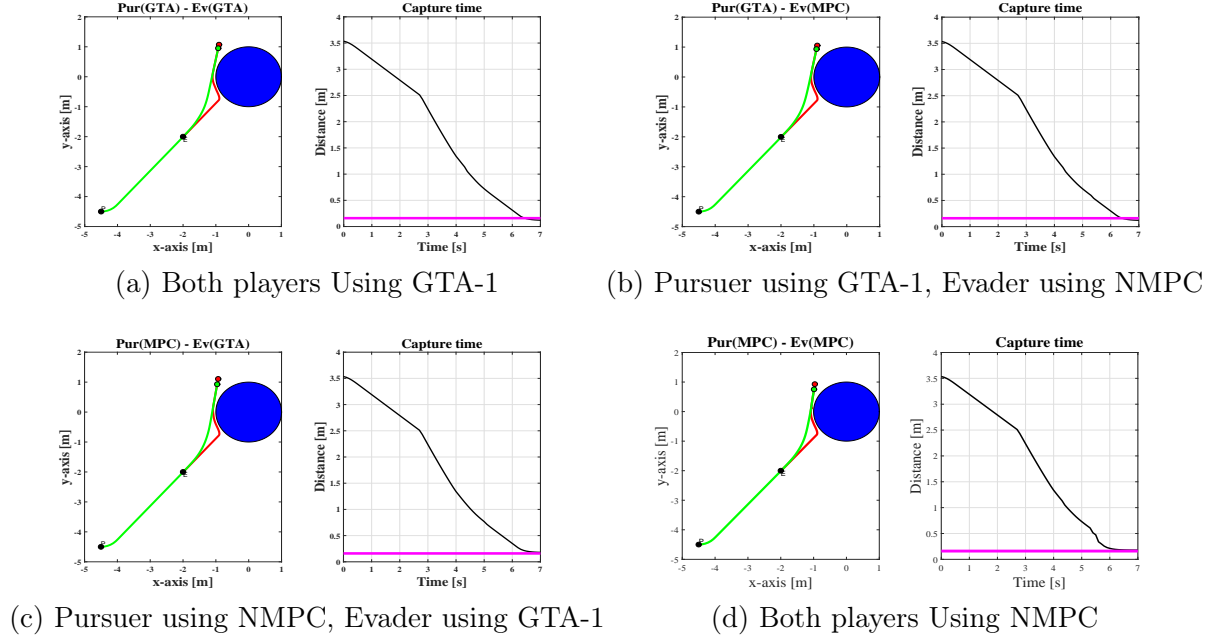


Figure 4.3: PEG with GTA-1 and NMPC in the presence of obstacles, the pursuer marked ‘P’ is green while the evader marked ‘E’ is red

To improve the game condition to a more realistic situation, an obstacle avoidance capability was incorporated. A big spherical obstacle with 2 meters in diameter is placed at the center of the game field (0,0). The players are placed in a strategic position so that the evader must encounter the obstacle before the end of the game. The results are presented in Figure 4.3. The initial conditions of the pursuer is $(-4.5, -4.5, 0)$ while the initial condition of the evader is $(-2, -2, \pi/4)$. These initial conditions are used for all the games in this scenario. In all the games, the left figure is the trajectory of the game while the figure at the right is the game pay-off. The capture time in all the cases is obtained at the point of intersection of the relative distance $r_{r,d}$ between the two players and the constant threshold distance (sum of the radii of the two robots) $r_{th} = 0.16$. Both players used GTA-1 in 4.3a, the capture time is 6.3 secs. Despite that the pursuer deviates from Nash equilibrium by using the NMPC algorithm in 4.3b, the capture time is 6.3secs still. In Figure 4.3c, the evader deviated from the Nash equilibrium and used the NMPC algorithm but still, the capture time is 6.3 secs. Also in the figure 4.3d, both payers deviated from the Nash equilibrium by employing the NMPC algorithm and still, the capture time is 6.3 secs.

These surprising results show that predicting only one optimal step of the opponent is not enough to increase the chance of winning of any player. This could be explained due to the closeness of the current state and the predicted state at each time instant. On the other hand, this could be seen as a plus to the NMPC approach due to its ability to change strategy at each time instant (via online optimization).

4.4.2 Simulation results of PEG using GTA-2

As detailed in section (4.3.2), each player predicts N optimal steps and their corresponding optimal controls of its opponent before computing its optimal strategy. The results of the GTA-2 are compared with an NMPC strategy where each player makes use of the current states of the opponent that is developed in section (3.3).

The diameter of each robot is $20cm$, the constraints on the pursuer's control variables are $|v_p| = 0.6m/s$, $|\omega_p| = \pi/3$ while the constraints on the evader's control variables are $|v_e| = 0.4m/s$, $|\omega_e| = \pi/4$. A faster pursuer is selected to ensure that the game terminates in final time. The prediction horizon $N = 10$ while the sampling time $T_s = 0.1$. The states constraints are the dimension of the polytope $10m$ by $10m$, therefore the constraints on both x -axis and y -axis from $-5m$ to $+5m$ and is applied to both players.

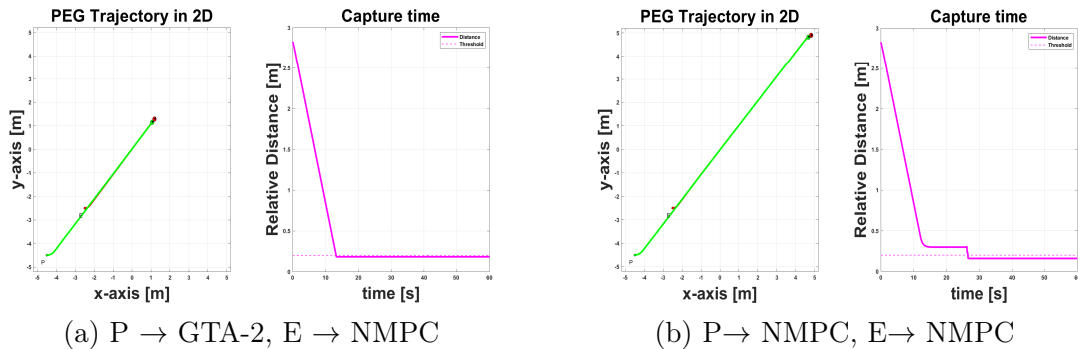


Figure 4.4: Comparing the performance of GTA-2 and NMPC in the absence of obstacles, the pursuer marked 'P' is green while the evader marked 'E' is red

In figure 4.4, the performance of the GTA-2 was shown compared to NMPC by playing the pursuit-evasion game for nonholonomic mobile robots in a plane environment. On the left side of 4.4a, the game trajectory was shown where the pursuer employs the GTA-2 while the evader employed the NMPC base evasion. The figure on the right of 4.4a shows the evolution of relative distance between the player until it coincides with the threshold, the capture time was $12.5secs$. The game was played again under the same initial condition with the pursuer employing NMPC-based pure pursuit while the evader using NMPC evasion in figure 4.4b. The game trajectory was shown on the left and the evolution of the relative distance between the players was shown on the right. It can be seen that the capture time is 27 seconds which is higher than when the pursuer uses GTA-2.

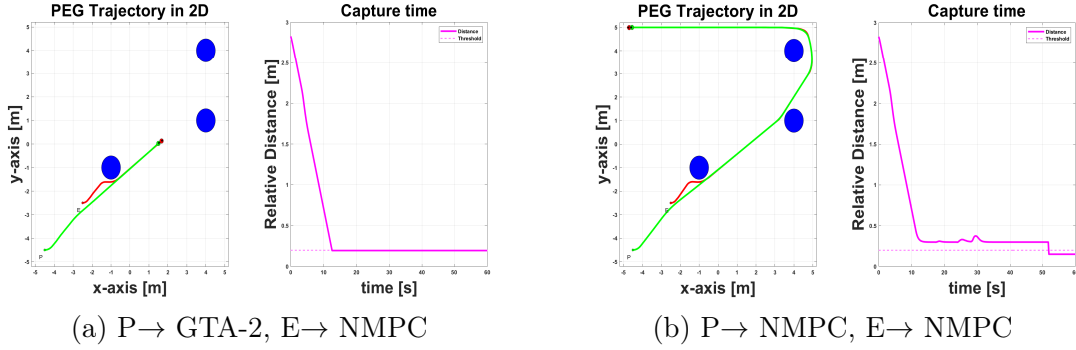


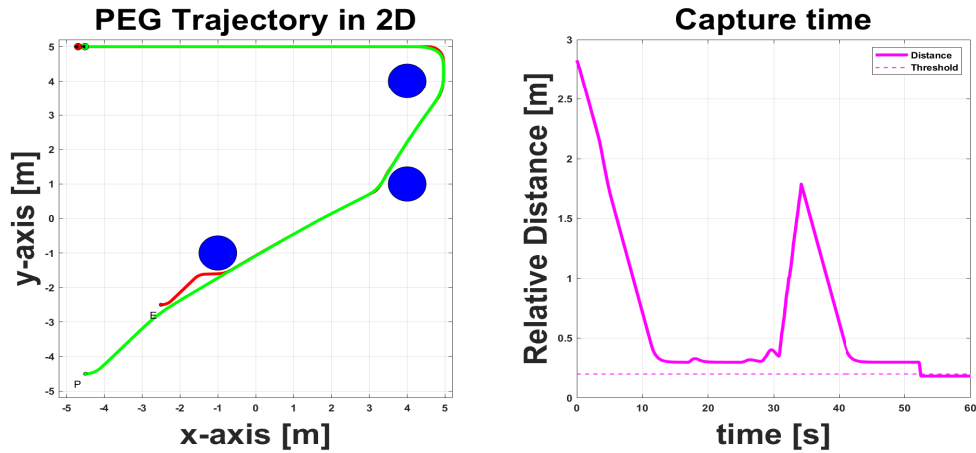
Figure 4.5: Comparing the performance of GTA-2 and NMPC in the presence of obstacles, the pursuer marked ‘P’ is green while the evader marked ‘E’ is red and the blue circles are the obstacles

Similarly, the scenario was replayed in an environment that contains obstacles in Figure 4.5. The parameters of the obstacles are deterministic and can be measured by both players. The first obstacle was placed at $(-1m, -1m)$ the second obstacle was placed at $(4m, 1m)$ and the third obstacle was placed at $(4m, 4m)$. The obstacle positions are selected and placed along the game trajectory so that the players must encounter them during the play. The diameter of all the obstacles was chosen as $1m$. On the left side of 4.5a, the game trajectory was shown where the pursuer employs the GTA-2 while the evader employed the NMPC base evasion. Both the pursuer and the evader avoided the first obstacle before the pursuer captures the evader. The figure on the right of 4.5a shows the evolution of relative distance between the player until it coincides with the threshold, the capture time was $12secs$. The game was played again under the same initial condition with the pursuer employing NMPC-based pure pursuit while the evader using NMPC evasion in figure 4.5b. It can be seen that both players avoided all the three obstacles before the pursuer eventually captures the evader. The game trajectory was shown on the left and the evolution of the relative distance between the players was shown on the right. It can be seen that the capture time is $52secs$ which is higher than when the pursuer uses GTA-2. The consistency in these results shows that the proposed approach can avoid obstacles and shortens the capture time of the pursuer.

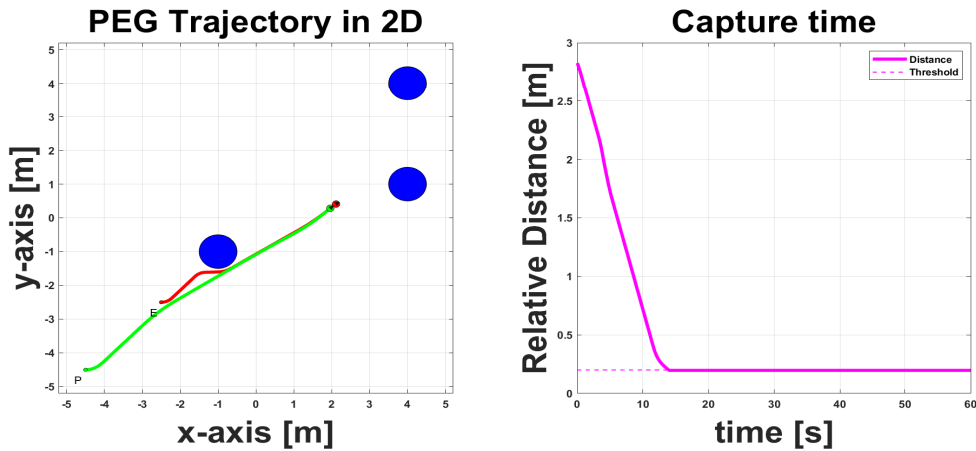
4.4.3 Simulation results of PEG using GTA-3

The solution approach for PEG problem using game-theoretic algorithm with incomplete information (GTA-3) was detailed in section (4.3.3). The diameter of each robot is $20cm$, the constraints on the pursuer’s control variables are $|v_p| = 0.6m/s$, $|\omega_p| = \pi/3$ while the constraints on the evader’s control variables are $|v_e| = 0.4m/s$, $|\omega_e| = \pi/3$. A faster pursuer is selected to ensure that the game terminates in final time. The prediction horizon $N = 10$ while the sampling time $T_s = 0.1$. The states constraints are the dimension of the polytope $10m$ by $10m$, therefore the constraints on both $x - axis$ and $y - axis$ from $-5m$ to $+5m$ and is applied to both players.

Figure 4.6 presented the results of the PEG where the proposed GTA-3 was compared with NMPC from the perspective of the pursuer. The game trajectory and the corresponding capture time were depicted in figure 4.6a when the pursuer employs NMPC, the capture time was found to be $52secs$. In contrast, figure 4.6b shows the game trajectory and the corresponding capture time was shown when the pursuer employs GTA-3, The capture time was found to be $13.9secs$ much lower than with NMPC.



(a) Pursuer using NMPC, Evader using GTA-3.



(b) Pursuer using NMPC, Evader using GTA-3

Figure 4.6: Comparing the performance of GTA-3 and NMPC

To verify the robustness of the proposed approach, a measurement noise with a power of $0.001dB$ was added to the evader's signal in figure 4.7. The game trajectory is presented in 4.7a while the noisy evader states is presented in figure 4.7b. The addition of the noise affected the capture time increases to $16,5secs$. When the noise power is multiplied by 10($0.01dB$) in figure 4.7, the game trajectory is presented in 4.8a while the noisy evader states is presented in figure 4.8b. It can be noticed that a higher value of the measurement noise affected the capture time which increases to $32secs$.

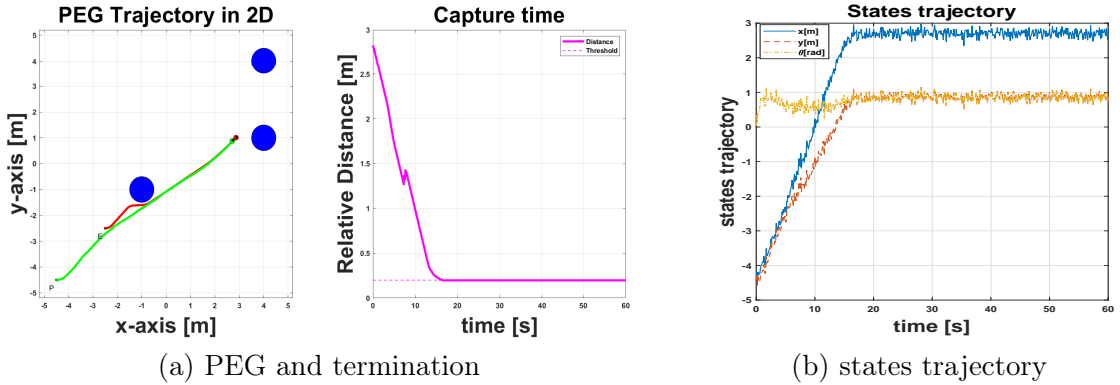


Figure 4.7: PEG with noisy measurements (noise power = 0.001)

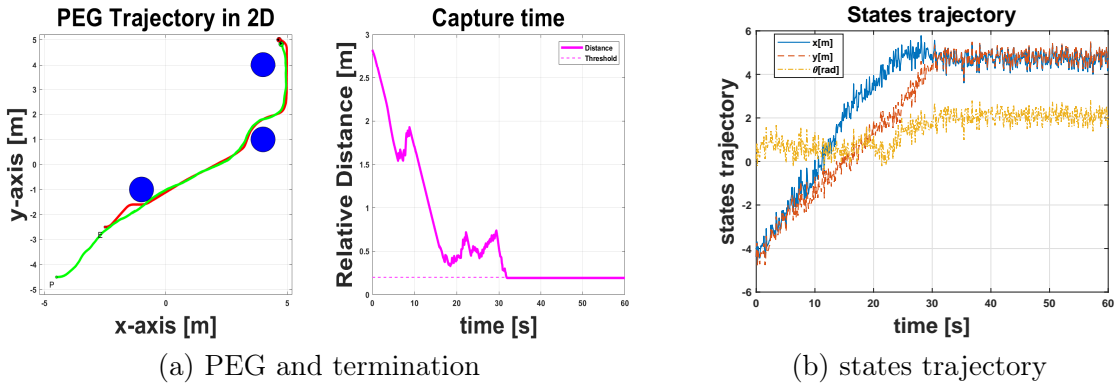


Figure 4.8: PEG with noisy measurements (noise power = 0.01)

4.5 Conclusion

This chapter builds on the previous chapter by implementing several game-theoretic algorithms to solve the pursuit-evasion of the game for nonholonomic mobile robots. These approaches are compared after rigorously revisiting different classes of the pursuit-evasion differential game. Information on the opponent is the key to choosing a game-theoretic strategy. The first method, (GTA-1) assumed that each player knows all the information about the opponent and then employs NMPC to predict the future optimal move of the opponent using point stabilization formulation. The performance of this method resembles the case in the previous chapter when each player has stated information about the opponent only. The second method (GTA-2) computes future trajectories of the opponent in advance before computing its optimal strategy using nonlinear model predictive tracking control. The capture time significantly improved as the pursuer was able to capture the evader in a shorter time. However, the approach results in a very large computation time especially when obstacles avoidance was incorporated. Finally, the third method, (GTA-3) with incomplete information on the opponent was proposed. With only the positions of the opponent known, the heading angle of

the opponent was computed (predicted) in advance by each player using a holonomic optimal strategy. The computation time of this approach is smaller than the previous methods. Table 4.2 presents the comparative analysis of the three game-theoretic approaches to solving the pursuit-evasion game in the presence and absence of obstacles.

	GTA-1	GTA-2	GTA-3	Winner
Capture time	poor	✓	average	GTA-2
Obstacles avoidance	✓	✓	✓	all
Computation time	0.08secs	0.098secs	0.032secs	GTA-3
Information requirement	×	×	✓	GTA-3
Robustness to noise	poor	poor	poor	none

Table 4.2: Comparative analysis

Differential Game of Static Target Defense

This chapter studied the differential game of static target defense which comprises three players; the attacker, the defender, and the target. In a differential game of active defense, the objective of the defender is to intercept an attacker before reaching a target, while the objective of the attacker is to reach the target while avoiding the defender and the objective of the target is to cooperate with the defender to avoid the attacker. This game resembles Issac's game of guarding a target in his book *'The Differential games'*. Several solutions to this game are proposed in the literature but suffer huge drawbacks due to their restriction to holonomic robots with simple equations of motions only and are not inherently compatible with obstacle avoidance. In this chapter, a novel approach called game-theoretic model predictive control was developed to deal with the differential game of static target defense for nonholonomic mobile robots. A single defender is made to pursue the attacker that has the same speed and is equidistant from the target. We have conducted several simulations in the presence and the absence of obstacles. A comparative analysis is conducted using a conventional nonlinear model predictive control approach. Results show that the proposed approach is sufficient and effective in handling differential games of static target defense in complex environments and situations.

5.1 Introduction

The research community has recently shown a keen interest in how autonomous robots can interact and collaborate to complete a given task. This is due to the increasing use of unmanned systems in both civil and military applications. Pursuit-evasion differential games present both interesting and challenging problems in many military applications. As a result, researchers concentrate on studies of pursuer strategies aimed at chasing and capturing the evader in the shortest amount of time. Cooperative pursuit of a single evader by multiple pursuers was proposed using safe-reachable area minimization in [Kothari, Manathara, and Postlethwaite 2017; Zhou et al. 2016; Bakolas and Tsiotras 2010; Pierson, Wang, and Schwager 2017]. The evader's safe-reachable area is computed using the Voronoi partition and then a cooperative strategy would be designed to minimize the safe-reachable area of the evader until it is captured. In [Garcia et al. 2017; Garcia et al. 2019; Von Moll et al. 2018], the cooperative pursuit problem was tackled using a geometric approach. The geometric location

of the players plays a significant role in utilizing either pure pursuit or cooperative pursuit. The method works in such a way that any pursuer that is closer to the evader will be the one to pursue the evader while the other remains stationary. When both players are equidistant from the evader, a cooperative strategy using an Apollonian circle would be employed so that both pursuers will pursue the evader concurrently.

It is pertinent to note in the classic cooperative pursuit problem, the evader is only interested to escape and has no goal or a target location to reach. This is also the same for a two-player pursuit-evasion game as presented in [Dong, Zhang, and Jia 2012; Tzannetos, Marantos, and Kyriakopoulos 2016; Sani, Robu, and Hably 2020]. However, incorporating a target whether as a player (dynamic) or as a goal in the conventional pursuit-evasion problems leads to the formation of a differential game of target defense also known as the game of guarding target. In this case, two pursuit-evasion games are coupled to produce three players (defender, attacker, and target). The objective of the defender is to intercept an attacker before reaching a target, while the objective of the attacker is to reach the target while avoiding the defender and the objective of the target is to cooperate with the defender to avoid the attacker. This scenario is presented as an active target defense differential game in [Liang et al. 2017]. Also, in [Garcia, Casbeer, and Pachter 2019], the closed-form solution differential game of active defense is presented for aircraft and missile applications. A team of slower target aircraft and defender missiles cooperate to intercept a faster attacker missile. The paper also presented the players' optimal strategies and verification of the closed-loop solution using Hamilton-Jacobi-Bellman equations. Other works on the design of cooperative strategy to enable the target to help the defending system to intercept the attacking missile are presented in [Getz and Leitmann 1979; Getz and Pachter 1981a]. These approaches are limited to holonomic robots with simple equations of motion only and did not consider obstacles avoidance.

In this chapter, the differential game of static target defense is studied which was firstly explained by the famous US scientist, Rufus Isaac in his book [Isaacs 1956] as a special case of the differential game of active target defense. The problem involves one defender (D) whose aim is to intercept an attacker (A) and to defend the static target (T). The attacker, on the other hand, is trying to reach the target while avoiding the defender. Both the defender and the attacker know their positions and that of the target. We consider a situation where the speed limits of the players are the same and the distance between each player and the target is equal. This scenario improves our previous work in [Sani, Robu, and Hably 2020] and the works of [Tzannetos, Marantos, and Kyriakopoulos 2016; Dong, Zhang, and Jia 2012] in the sense that the defender's objective is not limited to the pursuit of the opponent but as well includes the protection of the target. Also, the attacker's objective is more specific. We propose a novel approach we termed game-theoretic model predictive control (GTMP) to handle the differential game of static target defense (DGSTD) in both simple and complex environments. From the perspective of the defender, GTMP contains a parameter that makes a trade-off between the attacker's interception and the target defense. GTMP method is compared with the NMPC-based tracking method and simulation results are presented.

The contribution of this chapter is that it provides an intelligent solution to Isaac's problem

of guarding a target that could work for both holonomic and nonholonomic robots. The method is robust to the presence of obstacles. Finally and most importantly, our method can achieve interception in a situation that requires multiple pursuers with a single one.

5.2 The differential game of static target defense

As briefed in the previous section, the DGSTD consist of a static target which could be considered as a position area closest point of a particular area. The objective of the defender is to pursue the attacker while protecting the target and the objective of the attacker, on the other hand, is to reach the target while avoiding the defender. From the perspective of the defender, the DGSTD control system is to employs a single defense system to protect the target against a malicious attacker in a bounded complex environment. To define the termination of the DGSTD, we need to revisit the notions of a game of kind and a game of degree. game of kind has two finite outcomes depending on whether or not the player can achieve its objective. For example, in a pursuit game, the objective might be capture and in a battle game, the objective could be the destruction of the opponent. Game of degree on the other hand has a continuum of possible payoffs. The players then try to maximize or minimize the established pay-off.

Consider a given DGSTD where the initial condition is in a part of the state space where a strategy for the defender exists such that, under optimal play, the defender intercepts the attacker, i.e the attacker was not able to reach the target, then the DGSTD game of degree is played where the attacker tries to minimize the terminal distance between the attacker and the target while the defender tries to maximize it. When our interest lies on whether the defender can capture the attacker or whether the attacker reaches the target, that way, we are dealing with a game of kind. To end our game, we used the concept of a game of kind, which gives birth to two termination criteria. The first condition is when the attacker was intercepted by the defender. This happens when the distance between the defender and the attacker is equal to some threshold value as given in (5.1).

$$C_1 : \sqrt{(x_A - x_D)^2 + (y_A - y_D)^2} = l_1 \quad (5.1)$$

Where $l_1 = R_A + R_D$.

The second termination condition is when the attacker captures the target. This occurs when the attacker eventually reaches the target without being intercepted by the defender as given in (5.2).

$$C_2 : \sqrt{(x_T - x_A)^2 + (y_T - y_A)^2} = l_2 \quad (5.2)$$

Where $l_2 = R_T + R_A$.

Therefore, the termination condition of the game can be defined as (5.3) which is the combination of the two conditions (5.1) and (5.2).

$$C = C_1 \cup C_2 \quad (5.3)$$

5.3 Game-theoretic model predictive control

Since the defender is interested in intercepting the attacker before it reaches the target, it is can be postulated that capturing the attacker and protecting the target are two goals that are linked together. This problem can be solve by combining the notions of game theory ad model predictive control. Game theory is applied when the attacker's intent is known. As a result, to catch the attacker, the defender must track the attacker's location, whereas, to protect the target, the defender must track the target's position. To satisfy both objectives concurrently, the defender is made to track a reference that depends on the parameter α which made a trade-off between the attacker's and the target's positions. This parameter is designed and tuned discretely until optimal value is found.

Consequently, the defender's control is formulated such that at every sampling time, the following OCP is solved:

$$\min_{\mathbf{u}_D} J = \|\mathbf{x}_D(N) - \mathbf{x}_R\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_D(k) - \mathbf{x}_R\|_Q^2 + \|\mathbf{u}_A(k)\|_R^2 \quad (5.4)$$

where,

$$\mathbf{x}_R = (1 - \alpha)\mathbf{x}_A + \alpha\mathbf{x}_T \quad (5.5a)$$

$$\alpha \in (0, 1) \quad (5.5b)$$

Subject to:

$$\mathbf{x}_D(k+1) = f_D(\mathbf{x}_D(k), \mathbf{u}_D(k)), \quad k = 0, 1, \dots, N \quad (5.6a)$$

$$\sqrt{(x_D(k) - x_{obs})^2 + (y_D(k) - y_{obs})^2} \geq (R_D + R_{obs}) \quad (5.6b)$$

$$\mathbf{x}_{D_{min}} \leq \mathbf{x}_D(k) \leq \mathbf{x}_{D_{max}} \quad (5.6c)$$

$$\mathbf{u}_{D_{min}} \leq \mathbf{u}_D(k) \leq \mathbf{u}_{D_{max}} \quad (5.6d)$$

The cost function depends on the parameter α which varies from 0 to 1. As the parameter increases from 0 towards 1, we are giving more weight to protecting the target than pursuing the attacker. The converse is also true. We have included some constraints in the formulation which comprises the defender's model, obstacles' avoidance, states, and input constraints.

On the other hand, the attacker is interested in reaching the target as quickly as possible while avoiding the defender, the problem can be formulated using conventional NMPC such that the target's state is regarded as the reference signal while the defender is taken as a dynamic obstacle. Therefore, the attacker will be solving the following OCP at every time instant.

$$\min_{\mathbf{u}_A} J = \|\mathbf{x}_A(N) - \mathbf{x}_T\|_{Q_N}^2 + \sum_{k=0}^{N-1} \|\mathbf{x}_A(k) - \mathbf{x}_T\|_Q^2 + \|\mathbf{u}_A(k)\|_R^2 \quad (5.7)$$

Subject to:

$$\mathbf{x}_A(k+1) = f_A(\mathbf{x}_A(k), \mathbf{u}_A(k)), \quad k = 0, 1, \dots, N \quad (5.8a)$$

$$\sqrt{(x_A(k+1) - x_D)^2 + (y_A(k+1) - y_D)^2} \geq (R_A + R_D) \quad (5.8b)$$

$$\sqrt{(x_A(k) - x_{obs})^2 + (y_A(k) - y_{obs})^2} \geq (R_D + R_{obs}) \quad (5.8c)$$

$$\mathbf{x}_{A_{min}} \leq \mathbf{x}_A(k) \leq \mathbf{x}_{A_{max}} \quad (5.8d)$$

$$\mathbf{u}_{A_{min}} \leq \mathbf{u}_A(k) \leq \mathbf{u}_{A_{max}} \quad (5.8e)$$

This cost function (5.7) comprises the running cost which is the sum of stages predicted over the prediction horizon and the terminal cost for stabilizing the controller. \mathbf{x}_A and \mathbf{u}_A respectively denote the states and control variables of the attacker whereas \mathbf{x}_T represents the position of the target taken as a reference to go. The constraint given by equation (5.8a) represents the nonlinear model of the robot. Two obstacle avoidance constraints are added to the formulation. The constraint (5.8b) is employed to make the attacker avoid the defender as a dynamic obstacle, while the second constraint (5.8c) is employed to avoid static obstacles that might be encountered. The weight matrices are tuned and the best values that stabilize the controller are chosen. The Q and R matrices are found to be diagonal matrices with diagonal elements obtained as (1, 1, 0.001) and (1, 1) respectively. The weight on the terminal penalty cost is found to be $1000 * Q$.

5.4 Simulation results

Simulations of several scenarios of the differential games of static target defense are implemented in a Matlab/Simulink environment. An open-source symbolic framework for automatic differentiation and optimal control software, CasADi [Andersson et al. 2019] is employed to code the algorithms using a multiple shooting approach to convert optimal control problems into a nonlinear programming problem. The nonlinear model of a nonholonomic robot is used to implement the control where the initial conditions are selected randomly such that both the defender and the attacker are at the same distance from the Target. Also, the speed limits are the same. We have conducted the simulations with and without obstacles and in each case, the GTMPC result was compared with the NMPC approach.

The first set of simulation experiments is conducted without any obstacles in the game environment. Several values of the parameter ranging from ($\alpha = 0.10, 0.25, 0.50, 0.75$ and 0.90) are tested so as to find the best value that guaranteed capture. In Figure 5.1, the defender (green player) employed GTMPC with $\alpha = 0.10$ tried to intercept the attacker (blue player) that is using a conventional NMPC, before it reaches the target (red). The game trajectory presented in Figure 5.1a shows that the defender's attempt was unsuccessful as the attacker reached the target in *31secs*. As it can be seen at the top Figure 5.1b, the game was terminated using the termination conditions presented in (5.1). The control profile of the defender is presented in Figure 5.1c.

Similarly, in Figure 5.2, the defender (green player) employed GTMPC with $\alpha = 0.25$ tried to intercept the attacker (blue player) that is using a conventional NMPC, before it reaches the target (red). The game trajectory presented in Figure 5.2a shows that the defender's attempt was unsuccessful as the attacker has reached the target also in *31secs*. As it can be seen at the top Figure 5.2b, the game was terminated using the termination conditions presented in (5.1). The control profile of the defender is presented in Figure 5.2c.

The same thing happened in Figure 5.3 when the defender (green player) employed GTMPC with $\alpha = 0.50$ tried to intercept the attacker (blue player) that is using a conventional NMPC before it reaches the target (red). The game trajectory presented in Figure 5.3a shows that the defender's attempt was unsuccessful as the attacker has reached the target also in 31secs. As it can be seen at the top Figure 5.3b, the game was terminated using the termination conditions presented in (5.1). The control profile of the defender is presented in Figure 5.3c.

However, in Figure 5.4, the defender (green player) employed GTMPC with $\alpha = 0.75$ tried to intercept the attacker (blue player) that is using a conventional NMPC, before it reaches the target (red). The game trajectory presented in Figure 5.4a shows that the defender's attempt was successful this time as the defender captured the attacker in 28.5secs. As it can be seen at the top Figure 5.4b, the game was terminated using the termination conditions presented in (5.1). The control profile of the defender is presented in Figure 5.4c.

Finally, in Figure 5.5, the defender (green player) employed GTMPC with $\alpha = 0.90$ tried to intercept the attacker (blue player) that is using a conventional NMPC, before it reaches the target (red). The game trajectory presented in Figure 5.5a shows that the defender's attempt was successful this time also as the defender captured the attacker in 28.5secs. As it can be seen at the top Figure 5.5b, the game was terminated using the termination conditions presented in (5.1). The control profile of the defender is presented in Figure 5.5c.

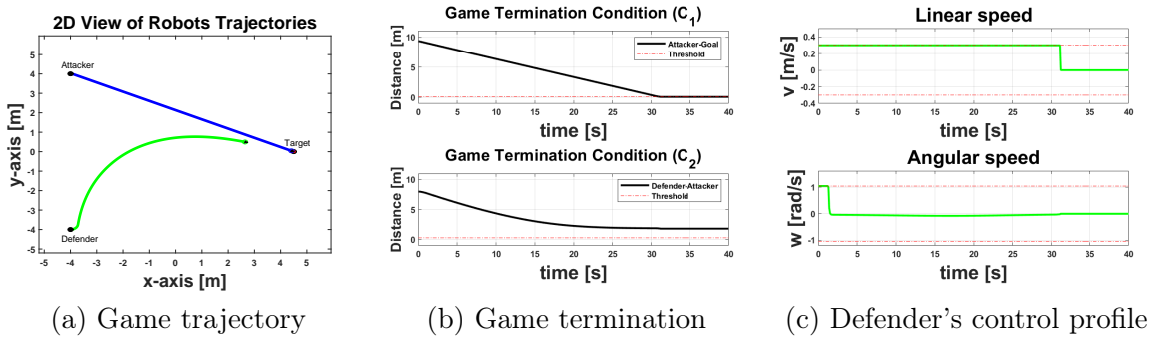


Figure 5.1: Attacker-defender-target game using GT-NMPC, $\alpha = 0.10$

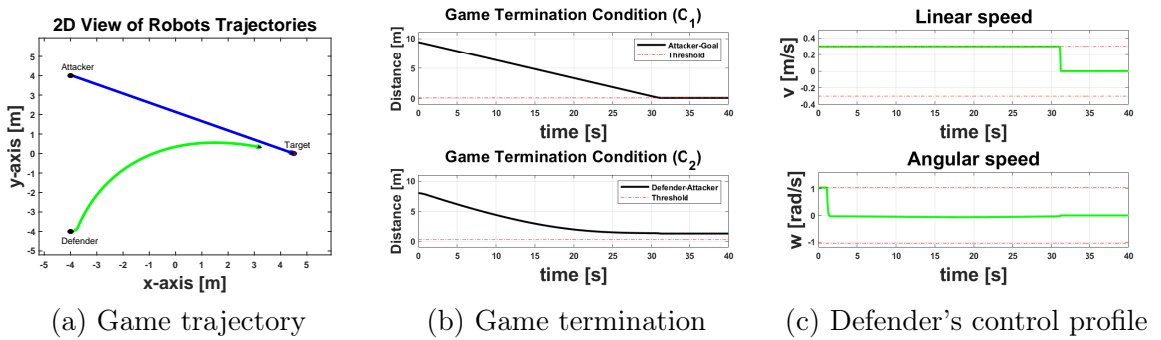


Figure 5.2: Attacker-defender-target game using GT-NMPC, $\alpha = 0.25$

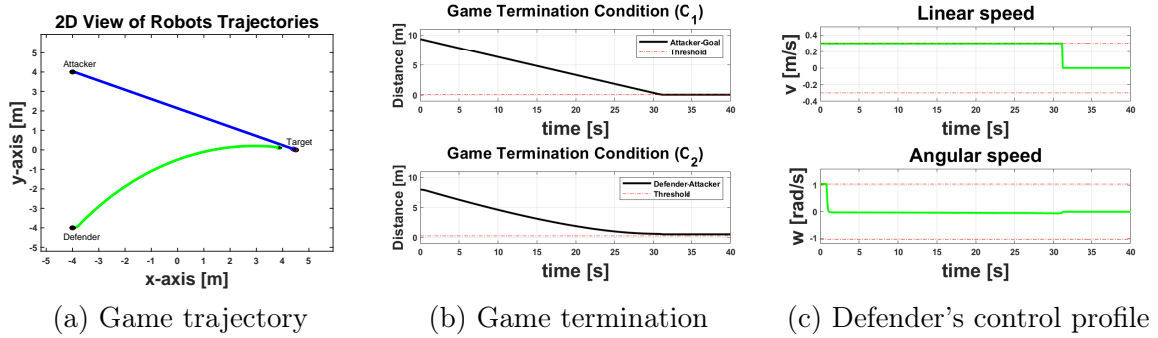


Figure 5.3: Attacker-defender-target game using GT-NMPC, $\alpha = 0.5$

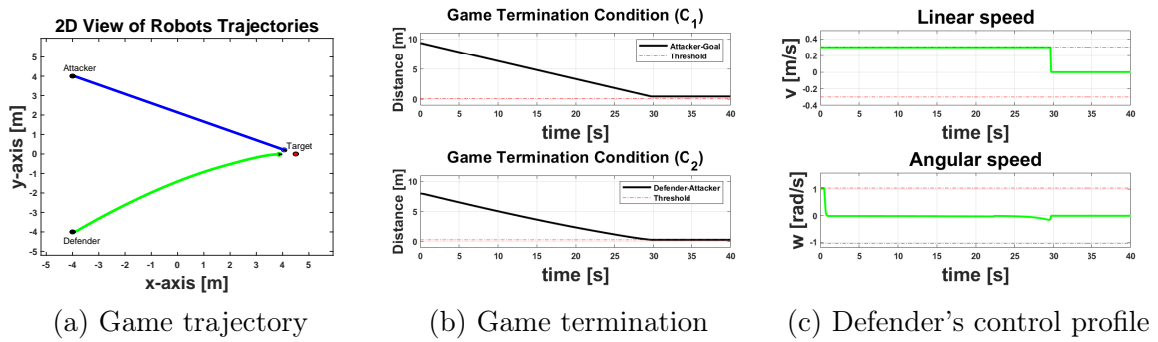


Figure 5.4: Attacker-defender-target game using GT-NMPC, $\alpha = 0.75$

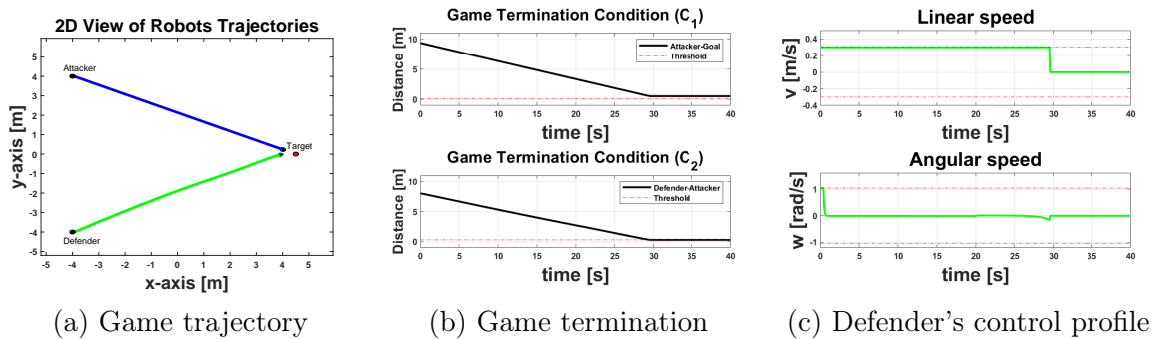


Figure 5.5: Attacker-defender-target game using GT-NMPC, $\alpha = 0.90$

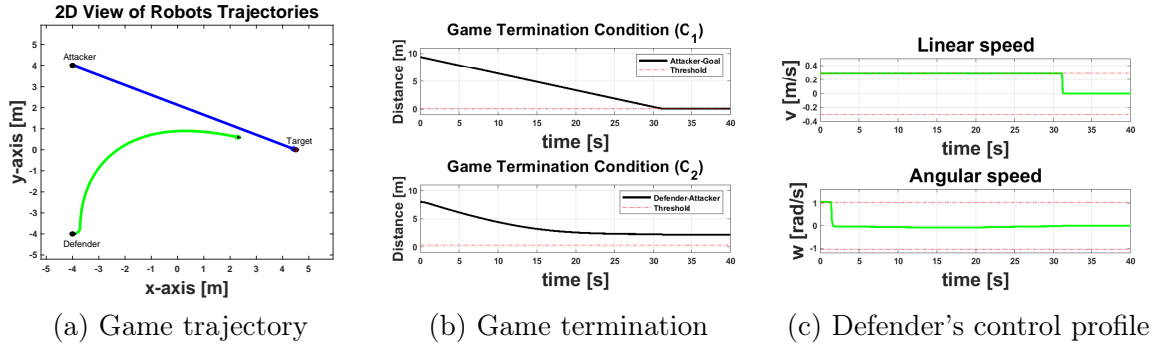


Figure 5.6: Differential game of static target defense using NMPC

The results in Figures 5.4 and 5.5 means that the optimal value of α ranges from 0.75 to 0.90, thus it is compared with the situation when the defender employed the NMPC in Figure 5.6. The game trajectory presented in Figure 5.6a show that the defender's attempt was unsuccessful as the attacker reached the target in 31secs. As it can be seen at the top Figure5.6b, the game was terminated using the termination conditions presented in (5.1). The control profile of the defender is presented in Figure 5.6c.

The second set of simulation experiments are conducted with some obstacles in the game environment. The same value of the parameter $\alpha = 0.90$ is used as in the first scenario. Similarly, in Figure 5.2, the defender employed GTMPC to intercept the attacker before it reaches the target. The game trajectory is presented in Figure 5.7a in which the attacker (blue player) aimed at the target while the defender (green player) pursued the attacker wisely and was able to capture it before reaching the target (red) in the presence of three obstacles (black). The termination conditions are presented in Figure 5.7b which also shows that the second condition given by equation (5.2) holds and the capture time is 17 seconds. The control profile of the defender was presented in Figure 5.7c.

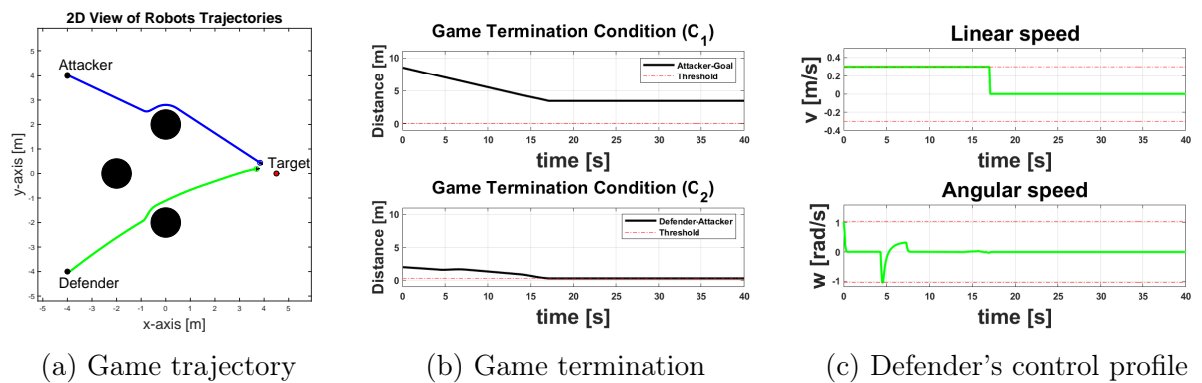


Figure 5.7: Differential game of static target defense with obstacles avoidance using GTMPC, $\alpha = 0.75$

Similarly, the result in Figure 5.7 is compared with the one in Figure 5.8 which employed the NMPC. Figure 5.7a shows the game trajectory in which we can see that the attacker was

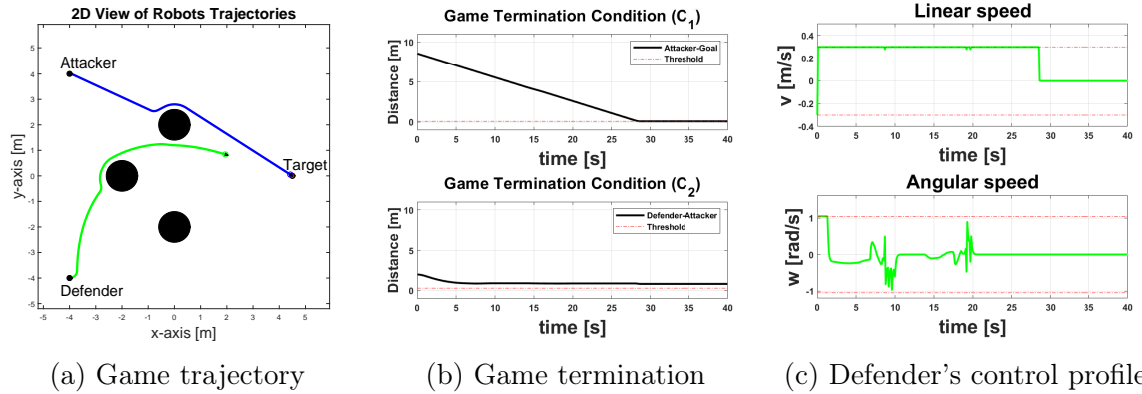


Figure 5.8: Differential game of static target defense with obstacles avoidance using NMPC

able to reach the target without the defender preventing it. This is because the defender's NMPC strategy is just a tracking problem, and it would follow the attacker blindly. Also, the first condition given by equation (5.1) holds to terminate the game at around 28 seconds as presented in Figure 5.8b. Also, the defender's control profile based on NMPC is presented in Figure 5.8c.

5.5 Conclusion

This chapter studied differential games under the context of static target defense and proposed a novel solution using the combination of nonlinear model predictive control and game theory. The game involves a defender whose objective is to intercept an attacker before reaching a target, the attacker's objective is to reach the target while avoiding the defender. A parameter α which provides a trade-off between pursuing objective and the defending objective of the defender was defined and tuned. The best value of the parameter that makes it possible for a single defender to intercept the attacker before reaching the target even if both have the same speed was obtained. The proposed game-theoretic model predictive control was compared with a conventional NMPC technique and applied on a nonholonomic mobile robot. Several simulations in the presence and the absence of obstacles are conducted. It can be concluded that the proposed approach was found to be very effective in handling differential games of static target defense.

Real-time Experimental Validations

6.1 Introduction

This chapter presents the real-time experimental validations of all the control algorithms proposed in this thesis. The controllers are tested on nonholonomic autonomous ground robots with two controlled wheels. The first experiment was about the control of a single robot in the presence and absence of obstacles, the second was about pursuit-evasion games with two robots and the third experiment was about the differential game of static target defense.

6.2 Experimental set-up

Two nonholonomic mobile robots (uni-cycle type) depicted in Figures 6.1a and 6.1b are fabricated at Gipsa-lab and named Robot 1 and Robot 2 respectively. Each robot consists of two controlled wheels on the left and right sides of the robot and a freewheel at the front to support the robot. Two continuous rotation servo motors are coupled to the left and right wheels. Each motor takes in the speed command signal (ω_r or ω_l) which commands the robot how fast to rotate and in which direction (forward or backward). The speed commands of each servo-motors of each robot are simultaneously received wirelessly from the IRC board in Figure 6.1d which communicates with the robot's Spectrum DSMX receiver. The communication between the Simulink and the IRC board is done via the UDP-protocol block in the Simulink.

On the other hand, the motion capture system used for this work was depicted in Figure 6.1c. It is comprised of 8 Miquis M3 cameras from Qualisys company which is used as a sensor to track the movement of the robots. The cameras are synchronized using an integrated Qualisys software which communicates with a Raspberry-pi board. The board contains a python code for converting the quaternions to Euler angles. The states of both robots are then sent to the Simulink in real-time via UDP protocol.

The controllers then use the states and other information to compute the control inputs v and ω which are converted to the angular speeds of the left (ω_l) and right (ω_r) wheels of the

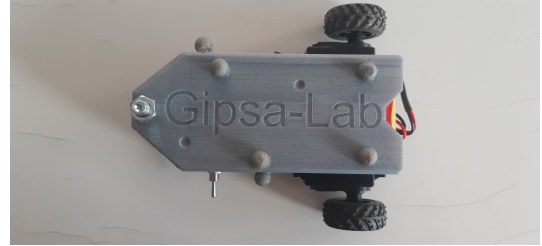
robot by the following equation:

$$\begin{cases} \omega_r = (2v + \omega R_{rob})/2r \\ \omega_l = (2v - \omega R_{rob})/2r \end{cases} \quad (6.1)$$

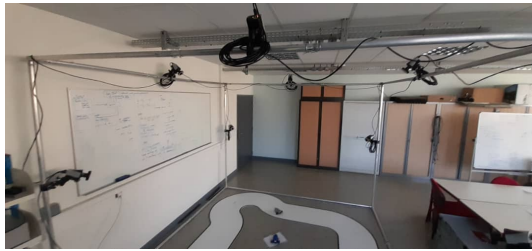
where R_{rob} is the length of the robot's base from the center and r is the radius of the robot's wheels.



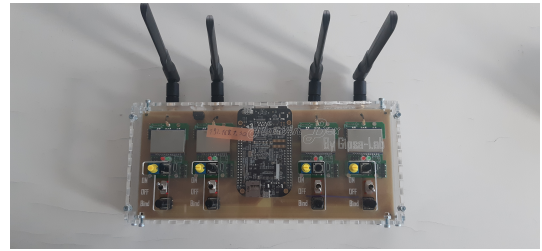
(a) Robot 1



(b) Robot 2



(c) Motion capture system



(d) IRC Communication system

Figure 6.1: Robots, communication and motion capture systems.

6.3 Control of single robot with obstacles avoidance

The set-point stabilization problem of a single robot was tackled using nonlinear model predictive control in chapter 2. The experiment was performed so that a robot moves from an initial state and is stabilized at the referent posture. The goal position $(1; 1; 0)$ was given to the controller by the operator while the current state of the robot is obtained by the feedback mechanism explained. The cost function was defined inside the NMPC block and then give the optimal control variables as an output.

Figure 6.2 presents the results in an obstacles-free environment. The robot makes a turn and heads towards the goal point and stabilized as seen in the figure on the top-left, while the state's error trajectory goes to zero asymptotically in the top-right figure. The two figures at the bottom presented a comparison between the commanded control and the actual controls. The actual linear speed at the bottom-left seems to align with the commanded linear speed well after 10 secs due to system delays. Similarly, the actual angular speed matches the commanded angular speed at the bottom-right corner.

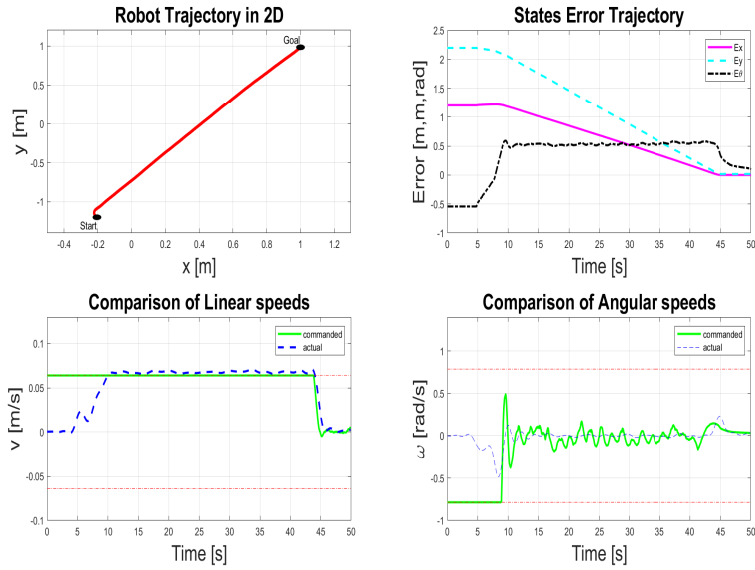


Figure 6.2: Point stabilization using NMPC in an obstacles free-environment

In Figure 6.3, the robot was commanded to go from its initial pose and stabilize at the same set-point while avoiding a static virtual obstacle. The obstacle was approximated by a circle(blue) and positioned along the path of the robot. The controller was able to steer the robot to reach the goal without collision with an obstacle. Similarly, the robot's trajectory and the obstacle are depicted on the top-left side while the state's error trajectory was depicted on the top-right side. The actual and the commanded optimal control inputs (linear and angular speed) are presented at the bottom of the figure.

To implement dynamic obstacles avoidance, we make use of the other robot and approximate it as a circular obstacle. The obstacle was made to rotate between the start and the goal position. Figure 6.4 presents the results of dynamic obstacles avoidance using NMPC with a prediction horizon of length 10. The robot's trajectory was taken at three different instants as depicted at the top part of the figure. The states error trajectory and the comparison of the control inputs are depicted at the bottom. The experiment was repeated with a prediction horizon of lengths 20 and 30 in figure 6.5 and 6.6 respectively. It can be seen that the performance of the NMPC with a prediction horizon of length 30 (prediction time = 3 secs) is better than the NMPC with a prediction horizon of length 20 (prediction time = 2 secs) and 10 (prediction time = 1 secs). This reveals the importance of high prediction horizons. However, care must be taken because the computation time is directly proportional to the prediction horizon.

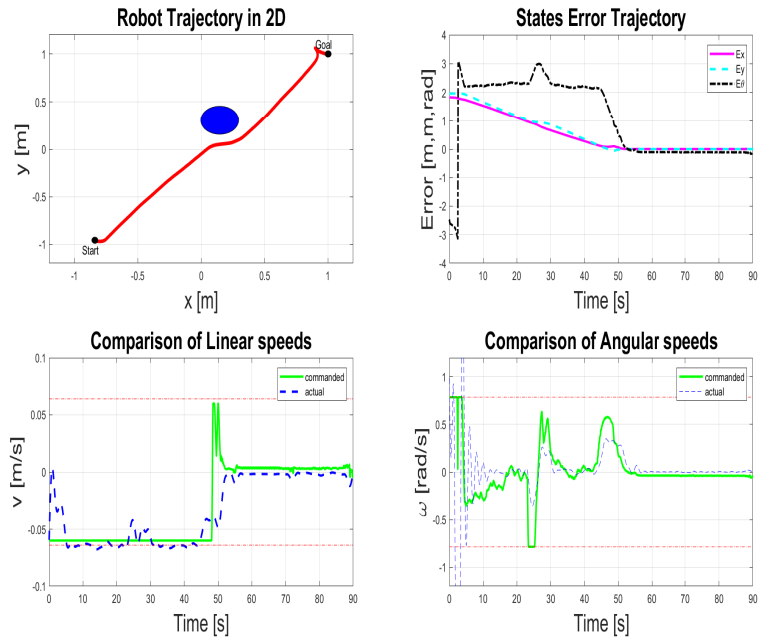


Figure 6.3: Static obstacles avoidance using NMPC

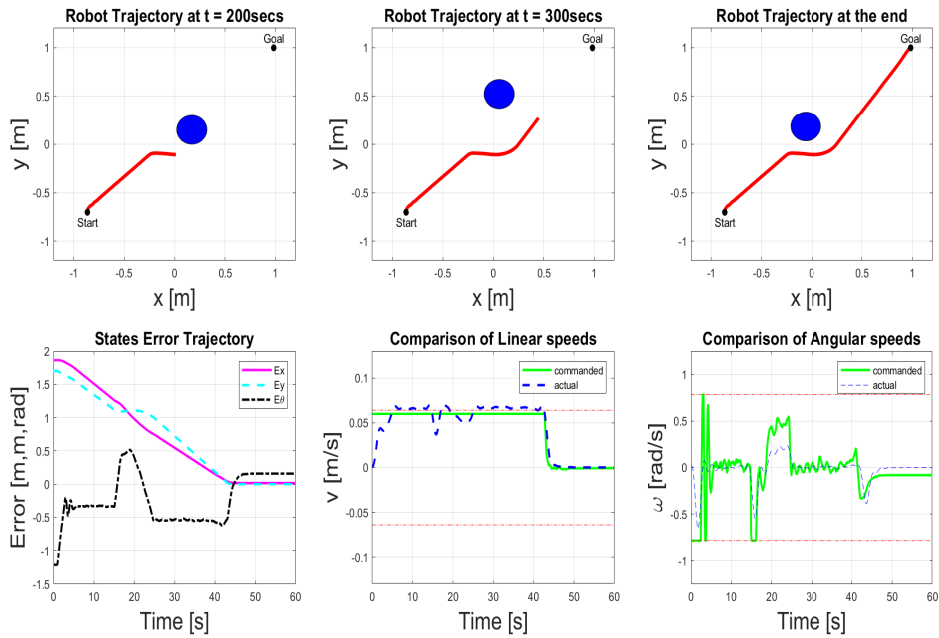


Figure 6.4: Dynamic obstacles avoidance using NMPC, $N = 10$

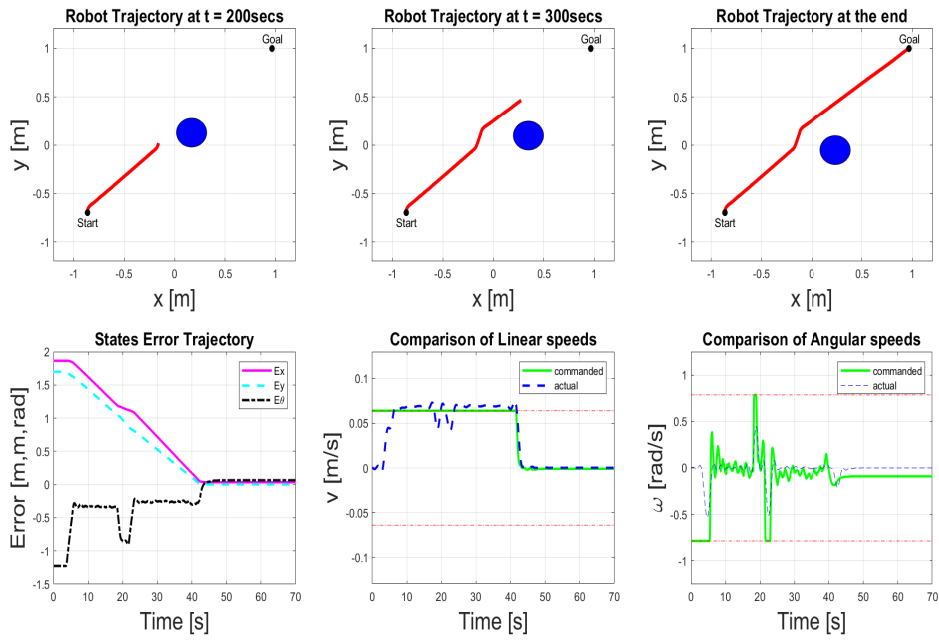


Figure 6.5: Dynamic obstacles avoidance using NMPC, $N = 20$

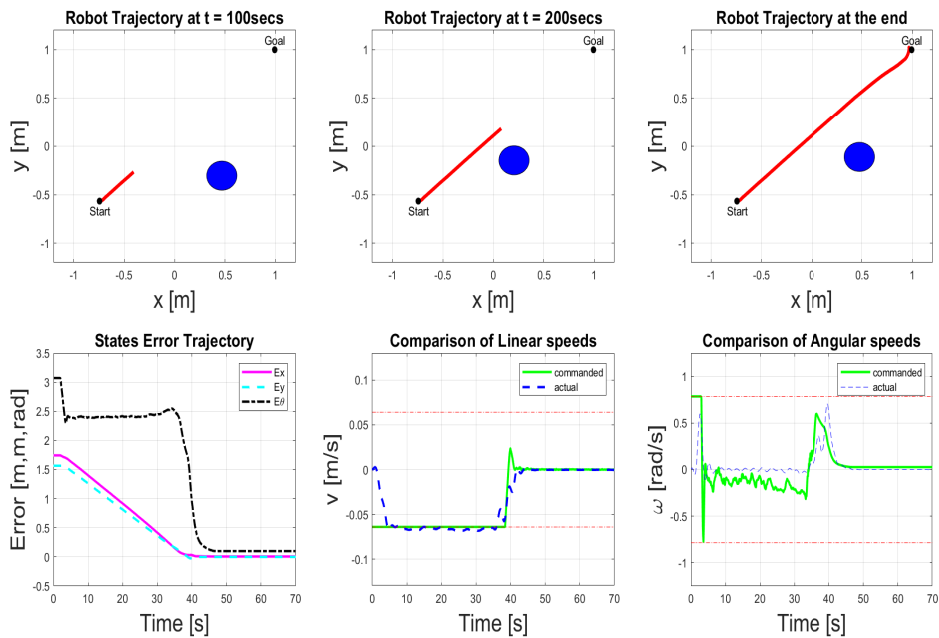


Figure 6.6: Dynamic obstacles avoidance using NMPC, $N = 30$

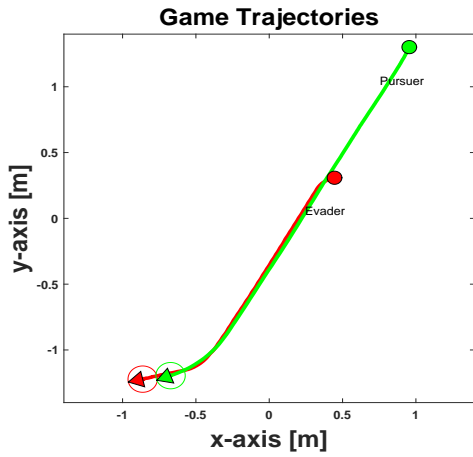
6.4 Pursuit-evasion games

Pursuit-evasion game is played experimentally in a constrained environment whose dimension is 3m by 3m. The pursuer tries to capture the evader, while the evader tries to escape. The pursuer is placed behind the evader in each play. To ensure game termination in finite time, the bounds on the pursuer's speed were selected to be higher than that of the evader. The linear speed bounds of the pursuer are -0.064 to 0.064 m/s while the linear speed bounds for the evader are -0.04 to 0.04 m/s. The game was played with Model Predictive Control based technique and a game-theoretic-based technique.

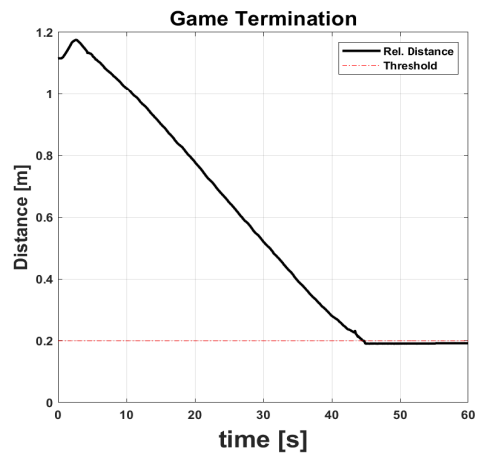
6.4.1 Pursuit-evasion games using nonlinear model predictive controls

In this section, each player localizes itself and measures the states information of the opponent using the camera systems and then solves its corresponding optimization problem. The pursuer solves the minimization problem using discrete NMPC with a sampling time of $0.1secs$, prediction horizon of length 10(i.e for a sampling time of $0.1secs$, the prediction time is 1 sec), and cost function which depends on the tuning parameters Q and R . The evader on the other hand solves the maximization problem using discrete NMPC with similar properties. Figure 6.7 presents a pursuit-evasion game using NMPC in an environment that contains no obstacles. The game trajectory in Figure 6.7a shows that the pursuer optimally tracked the evader while the evader optimally avoided the pursuer until the former captures the latter at $44secs$ as can be seen in Figure 6.7b. The relative distance between the two players coincides with the threshold distance ($0.2m$) which is defined as the sum of the radius of the pursuer ($0.1m$) and the radius of the evader ($0.1m$). The control actions of the pursuer in Figure 6.7c depict the comparisons between the commanded controls and the actual controls from the real data. Similarly, the control actions of the evader were presented in Figure 6.7d.

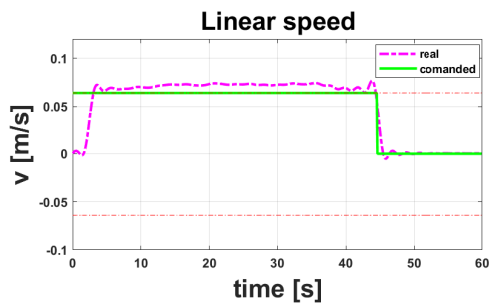
The game was repeated in the presence of virtual obstacles placed at the strategic locations. Figure 6.8 shows that each player pursued its objective and also avoided obstacles. The black circles in Figure 6.8a represent the obstacles and their positions while the green and the red circles represent the pursuer and the evader respectively. The capture time of the game as can be seen in Figure 6.8b was smaller due to the presence of the obstacles which makes the evader strive to avoid them at the beginning of the game, thus giving the pursuer an advantage. The strategy profiles of the pursuer and the evader are shown in the Figures 6.8c and 6.8d respectively.



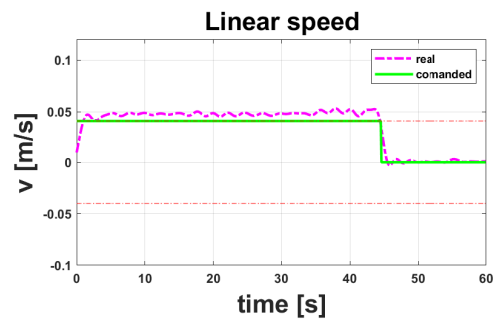
(a) Game Trajectory



(b) Game termination



(c) Pursuer's strategy



(d) Evader's strategy

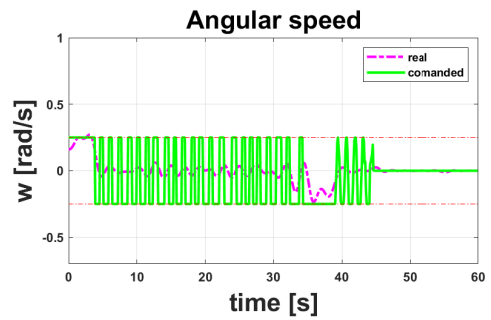
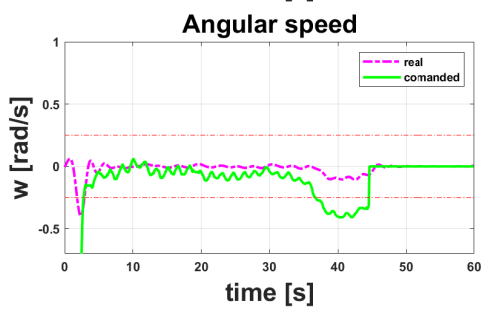
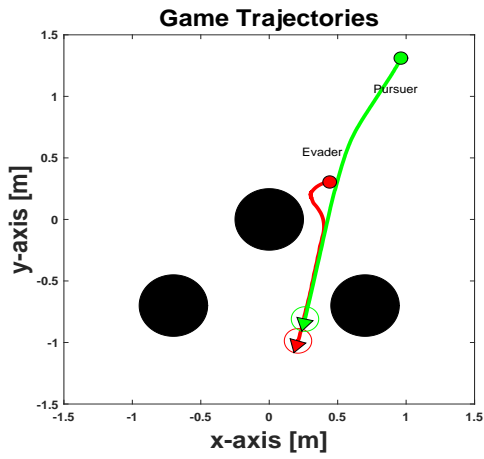
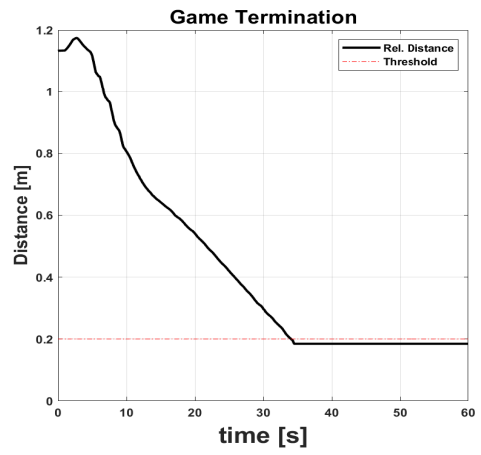


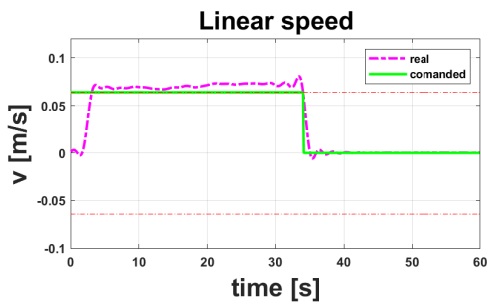
Figure 6.7: PEG using NMPC in an obstacles free environment



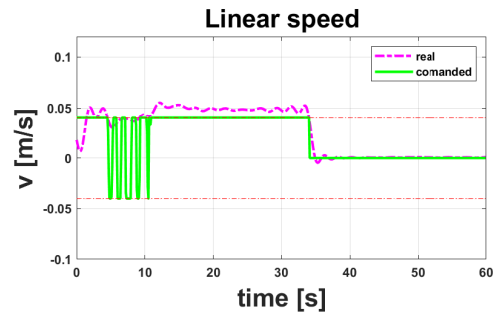
(a) Game trajectory



(b) Game termination



(c) Pursuer's strategy



(d) Evader's strategy

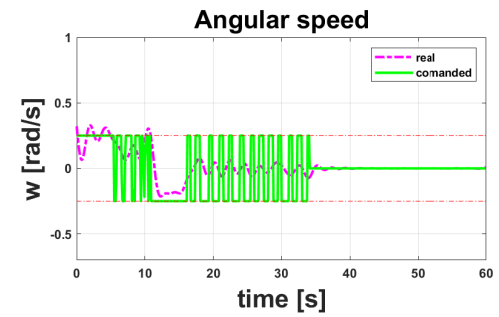
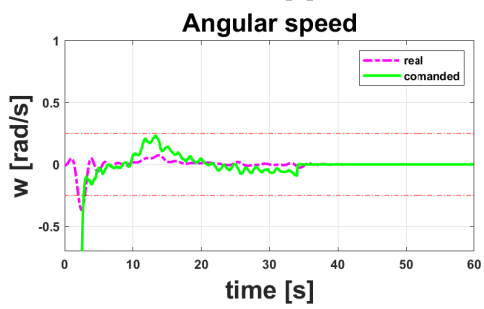


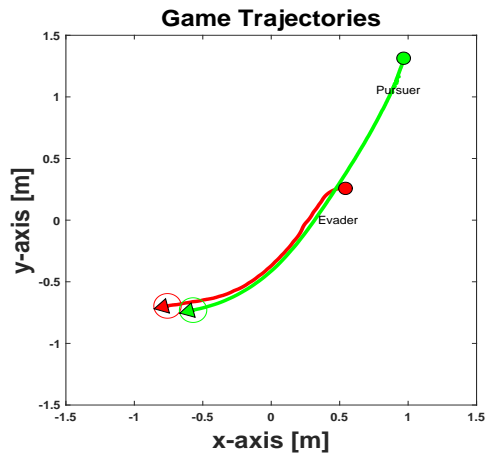
Figure 6.8: PEG using NMPC in the presence of obstacles

6.4.2 Game-theoretic method with incomplete information

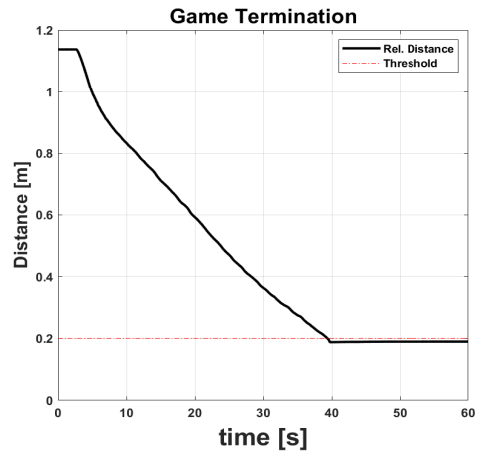
In this section, a game-theoretic approach with incomplete information of the opponent is employed to tackle the pursuit-evasion problem experimentally. As described in chapter 4, each player can measure only the position information of the opponent and predict the optimal orientation angle in advance.

Figure 6.9 presents the results of the pursuit-evasion game in a free bound environment where the pursuer employed the game-theoretic method with limited information and the evader employed the NMPC. The game trajectory depicted in 6.9a shows that the pursuer captured the evader in a short time 39 secs as it can be seen in 6.9b. The comparison between the actual speed and the commanded control profile is for the pursuer and the evader is respectively shown in 6.9c and 6.9d. However, with the incorporation of virtual obstacles (represented by the black circles in 6.10a), the pursuer captured the evader in 31 secs as shown in Figure 6.10.

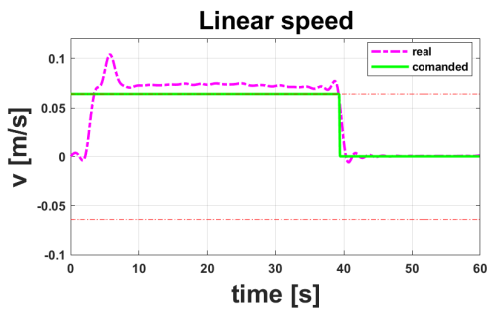
The experiment was repeated with both players using the game-theoretic method with limited information on the opponent. Figure 6.11 shows the result of the pursuit-evasion game in a bounded environment free of obstacles. The game terminates a bit longer (53secs) because the evader uses a better approach as seen in 6.11b. The strategy profile of each player is shown in figures 6.11c and 6.11d. Similarly, with the incorporation of virtual obstacles, both players avoided the obstacles and pursued their goal as shown in Figure 6.12. The game terminated in 48 secs as seen in 6.12d.



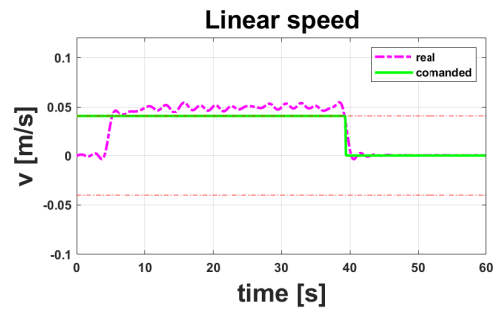
(a) Game Trajectory



(b) Game termination



(c) Pursuer's strategy



(d) Evader's strategy

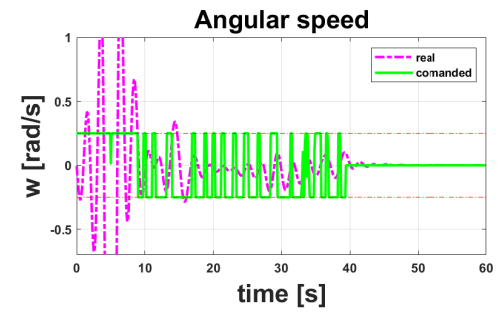
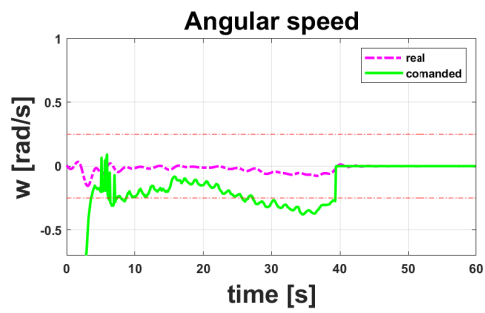
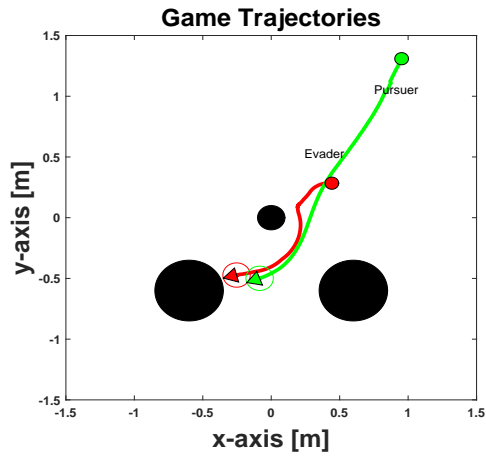
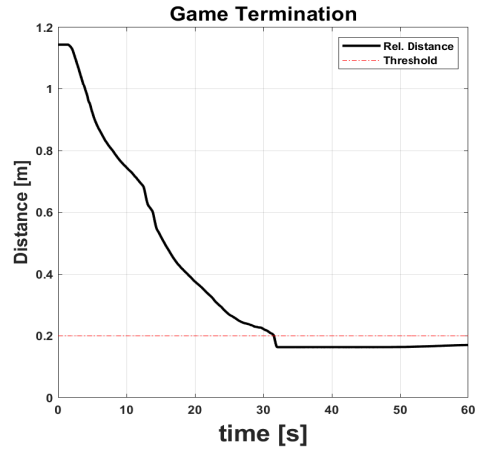


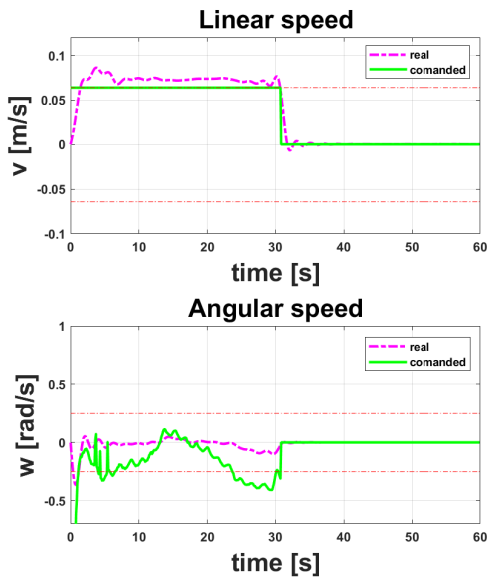
Figure 6.9: PEG with pursuer using LI-NMPC, evader using NMPC in a free-bounded space



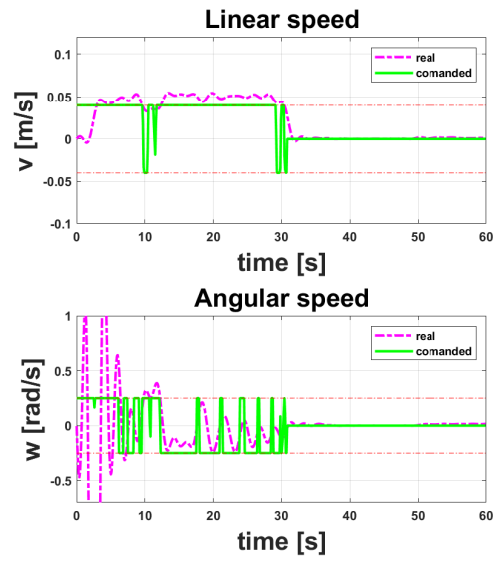
(a) Game Trajectory



(b) Game termination

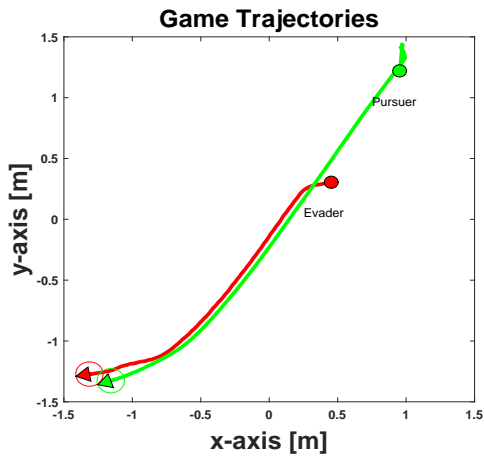


(c) Pursuer's strategy

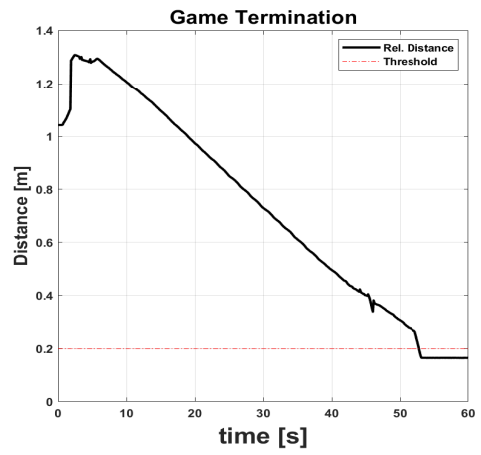


(d) Evader's strategy

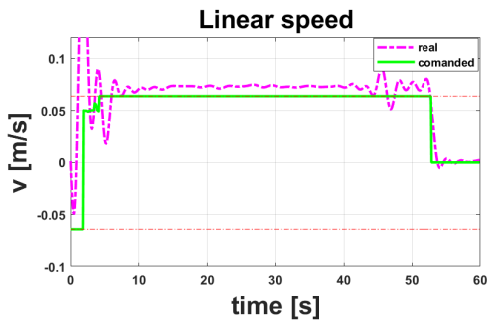
Figure 6.10: PEG with pursuer using LI-NMPC, evader using NMPC in an environment containing obstacles



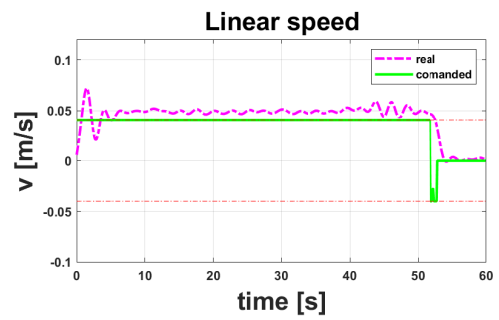
(a) Game trajectory



(b) Game termination



(c) Pursuer's strategy



(d) Evader's strategy

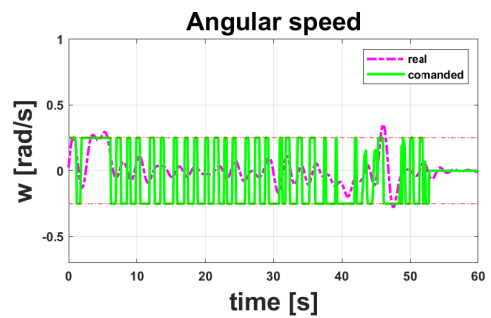
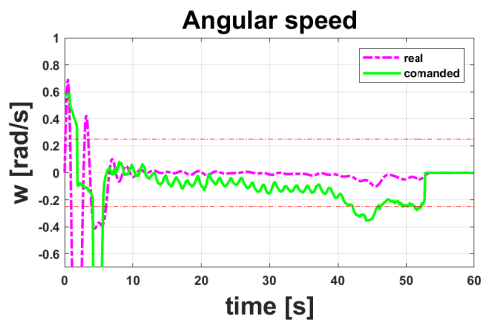
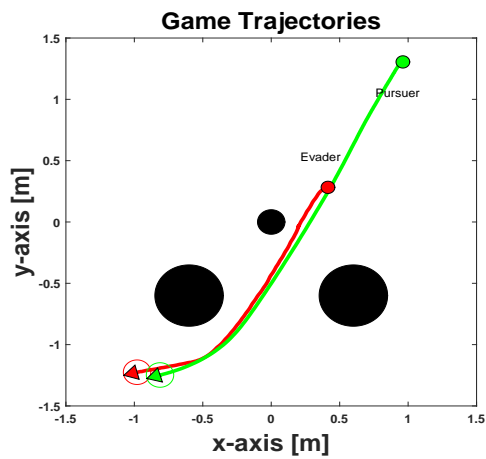
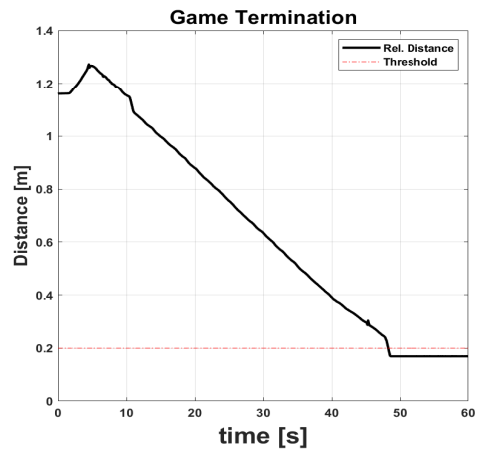


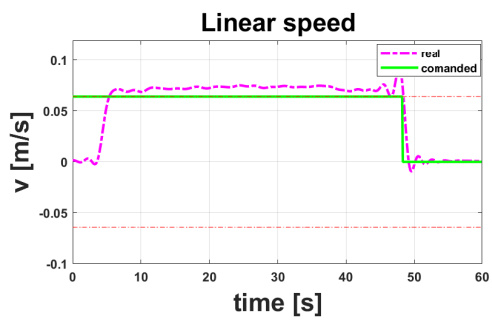
Figure 6.11: PEG with both players using LI-NMPC in a free-bounded space



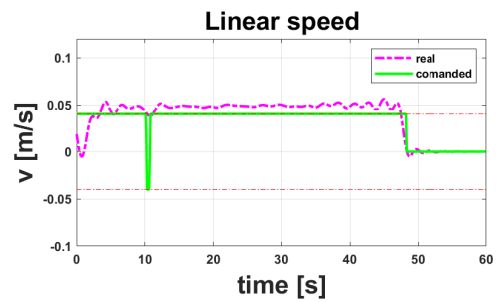
(a) Game trajectory



(b) Game termination



(c) Pursuer's strategy



(d) Evader's strategy

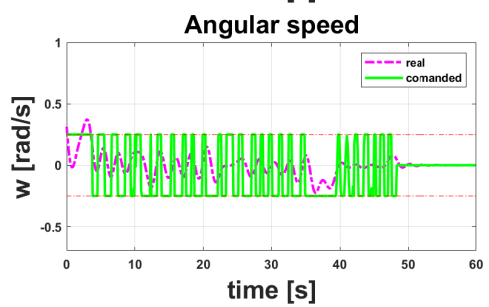
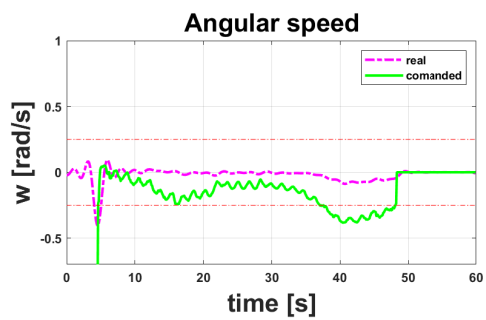


Figure 6.12: PEG with both players using LI-NMPC in an environment containing obstacles

6.5 Differential game of target defense

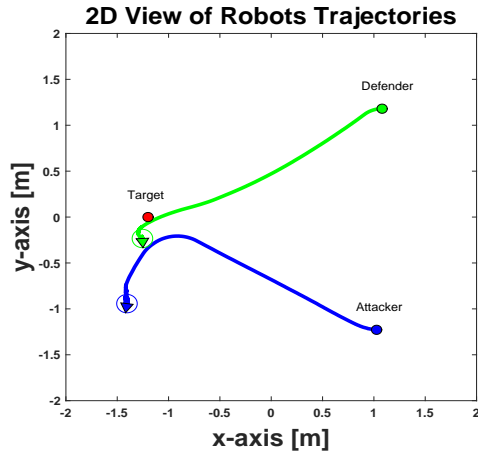
Experimental validation of the controllers dealing with the differential game of static target defense was presented in this section. The game was played by the two robots called a defender and an attacker in the presence and the absence of obstacles. The performance of the proposed controller, GTMPC was compared with the NMPC from the perspective of the defender.

Each experiment can terminate in three ways, i.e when the attacker reaches the target or when the defender captured the attacker, or when the game time elapses (120 secs) which happened when the defender prevented the attacker from reaching the target. Also, the initial positions of each game were selected so that the players are equidistant from the target.

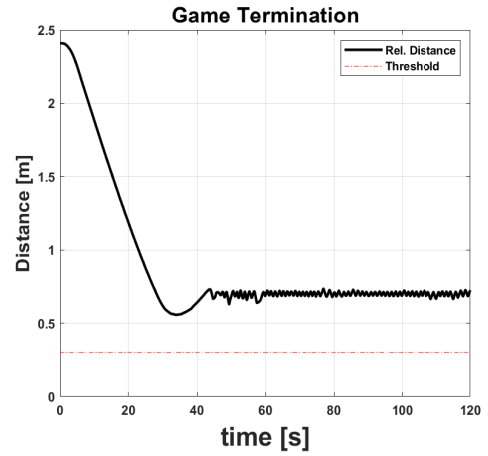
6.5.1 Experimental results of DGSTD in obstacles free environment

GTMPC was implemented on the defender in Figure 6.13 in a bounded-free environment. The game trajectory in figure 6.13a shows that the defender prevented the attacker from reaching the target until the game terminated. The relative distance between the defender and the attacker in figure 6.13b could not intercept the threshold because the defender combined two objectives of pursuing the attacker and protecting the target from attackers inversion. The attacker also considers the defender as a dynamic obstacle, that is why it moves away from the defender to avoid capture. The strategy profiles of the defender and the attacker are shown in figures 6.13c and 6.13d respectively.

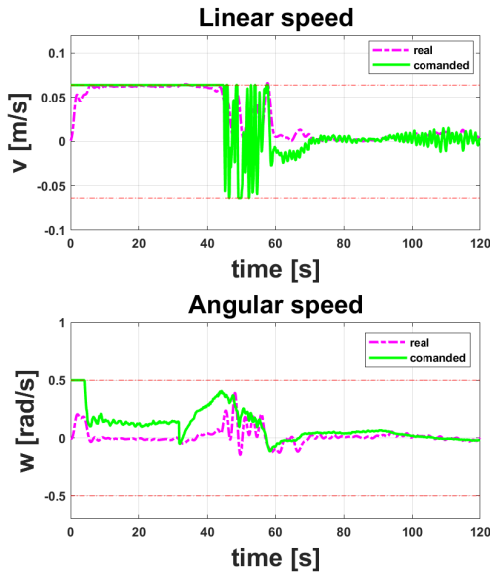
To show the advantage of the proposed controller, the scenario was repeated with the defender employing the conventional NMPC in figure 6.14. The game trajectory in figure 6.14a shows that the defender tracks the attacker but could not intercept or prevent the attacker from reaching the target. This means that the defender blindly tracks the attacker without any form of intelligence, thus unless with speed advantage, the defender can never intercept the attacker in this scenario. The game terminated after 120 secs and the relative distance between the players depicted in figure 6.14b shows that the defender stopped tracking the attacker because the attacker have already reached the target. The control strategy profiles of the defender and the attacker are respectively shown in figures 6.14c and 6.14d.



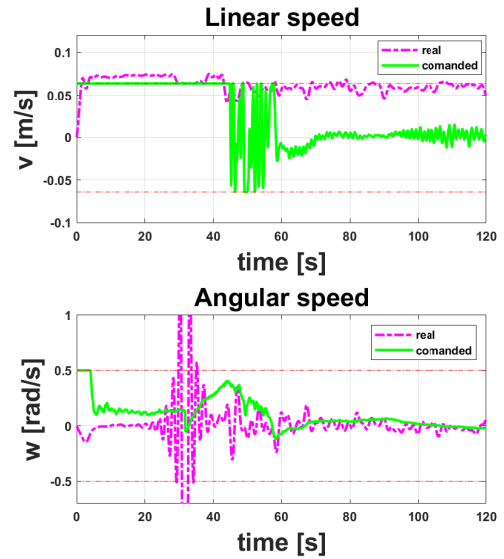
(a) Game trajectory



(b) Game termination



(c) Defender's strategy

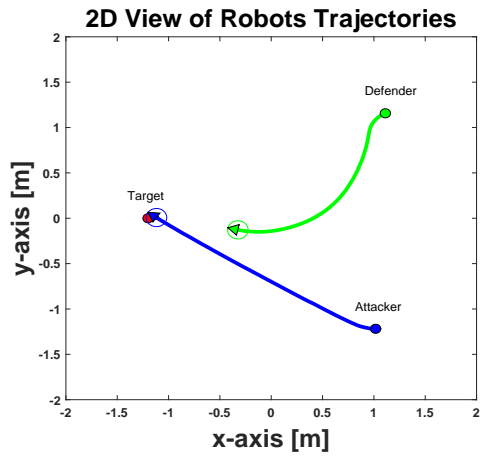


(d) Attacker's strategy

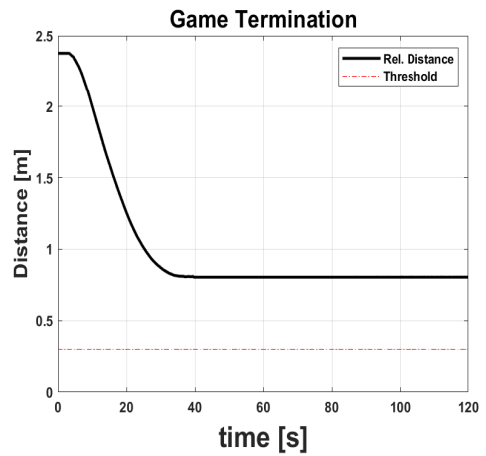
Figure 6.13: Differential game of static target defense using GT-NMPC

6.5.2 Experimental results of DGSTD in an environment with obstacle

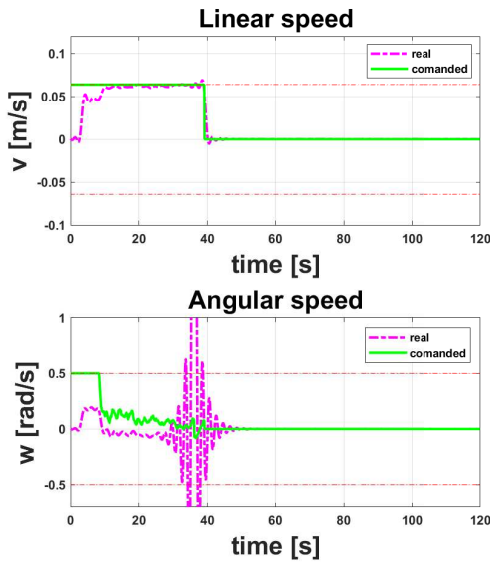
Obstacles avoidance was incorporated in the formulation of the proposed controller. Three virtual obstacles are placed at strategic positions within the game environment as shown in Figure 6.15. The trajectories of the robots in figure 6.15a show that the defender intercepted the attacker before it reaches the target. The game terminated when the relative distance between the players coincides with the threshold value at around 36 secs in figure 6.15b. The strategy profiles of the players are shown in figures 6.15c and 6.15d.



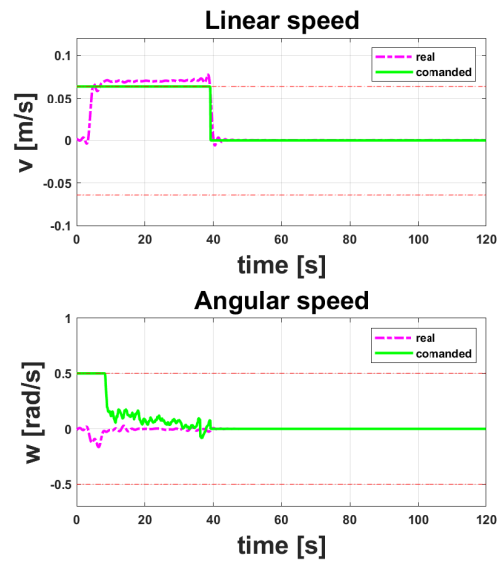
(a) Game trajectory



(b) Game termination



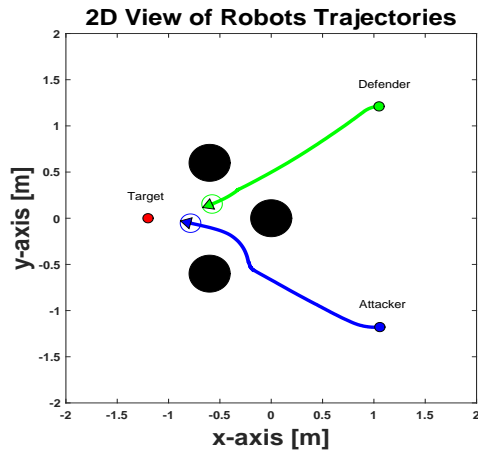
(c) Defender's strategy



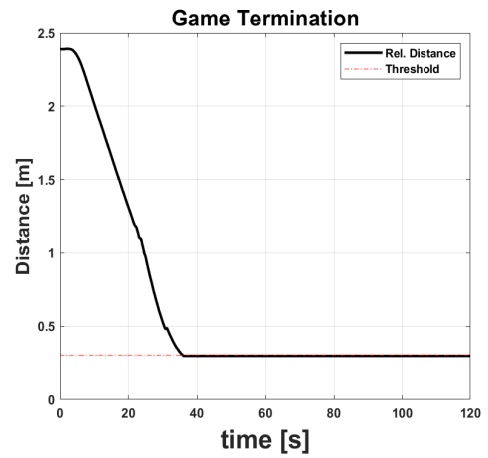
(d) Attacker's strategy

Figure 6.14: Differential game of static target defense using NMPC

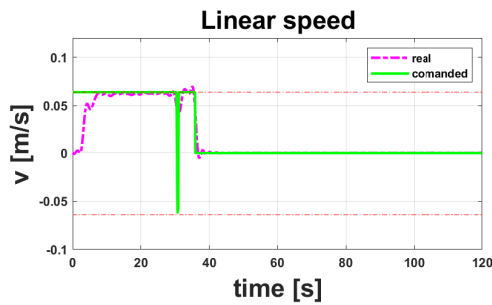
However, when the game was repeated with the defender using NMPC in figure 6.16, the game trajectory in figure 6.16a shows that the defender could not intercept the attacker. The relative distance between the players could not coincide with the threshold as seen in figure 6.16b and the game terminated at 120 secs. The strategy profiles of both players are presented in figures 6.16c and 6.16d respectively.



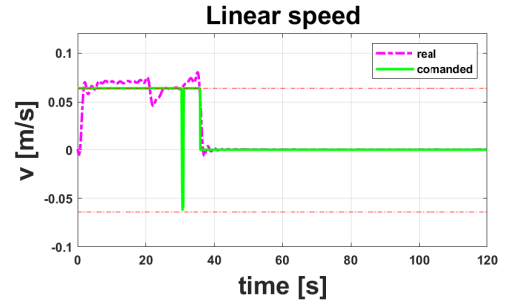
(a) Game Trajectory



(b) Game termination



(c) Defender's Strategy



(d) Attacker's Strategy

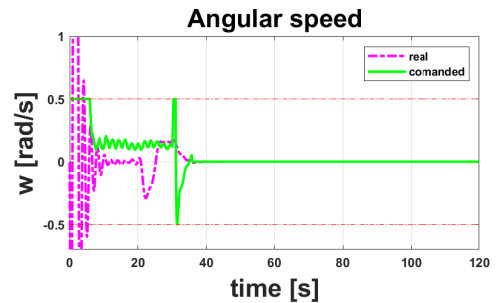
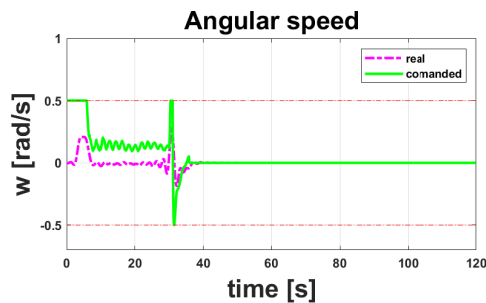
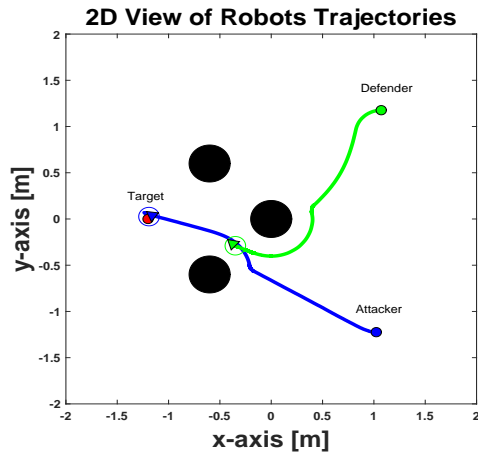


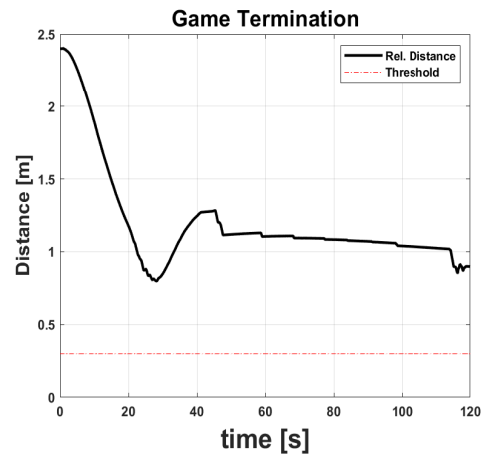
Figure 6.15: Differential game of static target defense using GTMPC

6.6 Conclusion

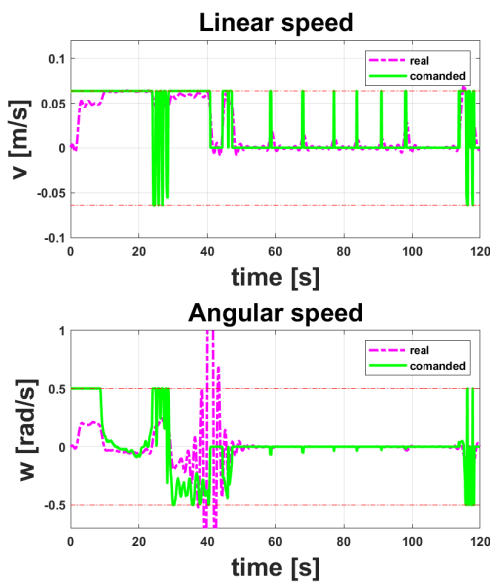
This chapter presented the experimental validations of all the proposed controllers developed in this thesis. The first set of experiments presents the results of dynamic obstacles avoidance using NMPC. A robot was approximated as a circular obstacle that must be avoided by the other robot whose objective was to reach a particular set-point blocked by the obstacle. Our proposed NMPC controller was seen to be capable of steering the robot to stabilize at the referent pose while avoiding dynamic obstacles with unknown (or unmeasured velocity).



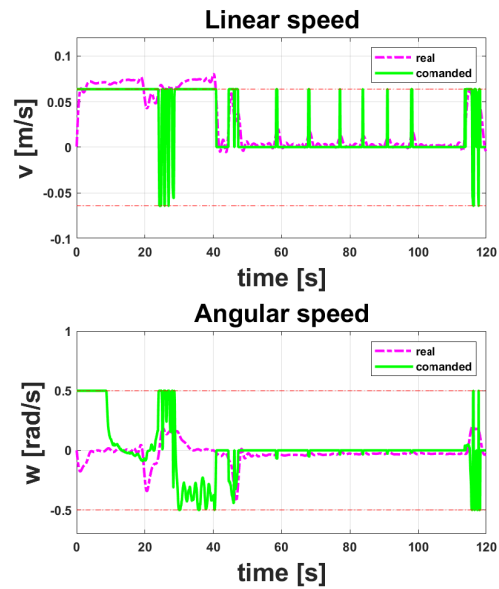
(a) Game trajectory



(b) Game termination



(c) Defender's strategy



(d) Attacker's strategy

Figure 6.16: Differential game of static target defense using NMPC

The second set of experiments is conducted in the context of 2 player pursuit-evasion game. The game was tackled when each player uses the NMPC algorithm and compared it with a proposed game-theoretic method that uses limited information from the opponent. The result of the proposed method outperformed the NMPC-based approach. Finally, the last set of experiments is performed in the context of the differential game of static target defense. A proposed controller, known as the GTMPC was applied to the defender to prevent an attacker from reaching a target. The results presented show that the proposed controller has outperformed the NMPC.

Summary and Outlooks

In this chapter, a summary of the thesis's main contributions and suggestions for some possible extensions to the work are presented.

The main focus of this research study was to develop game-theoretic solutions for non-cooperative control problems and to validate the controllers on a physical robot. The game-theoretic solutions were based on an online optimization method that incorporates obstacles avoidance, state, and input constraints. The online optimization-based solutions developed are the nonlinear model predictive control, the game-theoretic algorithm with complete information, the game-theoretic algorithm with incomplete (limited) information, and the game-theoretic model predictive control.

This PhD thesis research has five key objectives:

Objective I: To design a novel NMPC controller for dealing with the problem of dynamic obstacles avoidance, with unknown speed for a nonholonomic mobile robot.

Objective II: To design two conflicting NMPC controllers for a two-player pursuit-evasion game in the presence of obstacles.

Objective III: To design novel game-theoretic controllers for two-player pursuit-evasion games based on complete and incomplete information about the opponent.

Objective IV: To design a novel game-theoretic model predictive control scheme for tackling differential game of static target defense.

Objective V: To Implement all the controllers on the physical robots for experimental validations.

7.1 Research summary based on objective I

The first objective considered the dynamic obstacle avoidance by a single nonholonomic mobile robot using NMPC. After an extensive literature review, it has been observed state-of-the-art controllers for the dynamic obstacles avoidance take into account the velocity of the obstacle apart from the position and the size. This consideration of the speed of the obstacle introduces additional computations and the need for an additional sensor during physical implementation. Therefore, in this study, we design an NMPC-based controller that steers the robot away from a moving obstacle without the knowledge of the obstacle's speed. This approach depends only on the measurement of the instantaneous position and the size of the obstacle. To this end, a safe distance was added to the sum of the radii of the robot and the obstacle to ensure that the robot was repelled from the obstacle from a far distance. The proposed method reduced the computation time of the MPC which is critical to the physical implementation. The success of this method was verified by both simulations and laboratory experiments. The results show that the proposed controller steers the robot from its initial pose to the reference posture and avoided dynamic obstacles.

It is pertinent to mention here that two discretizing methods for formulating the discrete NMPC are compared. The 4th order Runge-Kutta method has outperformed the Euler method, thus it is used throughout the research.

7.2 Research summary based on objective II

The second objective considered in this thesis was to extend the application of the controller designed in the first objective to handle the pursuit-evasion game between two nonholonomic mobile robots. An extensive literature review on the pursuit-evasion game reveals several offline solutions to the problem where the system was holonomic with simple equations of motions. The methods also do not take into account constraints on the state, inputs, and obstacles avoidance. The few online techniques use double optimization based on unrealistic assumptions. In addition to the resulting large computation time, the approaches also were not implemented on physical systems. In this study, we designed an online solution using NMPC based on the full states information of the opponent only. The pursuer's controller was formulated to minimize cost function subject to its dynamic model, state constraints, input constraints, and obstacle avoidance constraints. The evader's controller was formulated to solve the maximization cost function subject to its dynamic constraints, states constraints, inputs constraints, and obstacle avoidance. The proposed NMPC controllers worked very well in making the pursuer capture the evader that is optimally controlled by a conflicting controller. A new method of stabilizing NMPC without any stabilizing cost or constraint was tested and compared with the conventional approach of utilizing terminal cost. The latter method worked better and was adopted throughout the research.

7.3 Research summary based objective III

The third objective of this thesis is to propose a more intelligent method of solving the pursuit-evasion problem. Game theory was employed from the social sciences so that each player takes into account the strategy of the opponent. Two approaches are then proposed based on the level of opponent information available to each player. The first method considers a situation where each player knows complete information about the opponent and solves the pursuit-evasion problem using double optimization. Each player predicts a sequence of optimal strategies of the opponent and then computes his strategy. When the pursuer employed this method, it captured the evader in a shorter time compared to when it uses the previous states-based NMPC. Due to the double optimization used, the method suffered high computation time which affects the physical implementation. To deal with the problem of computation time, a second method was proposed based on incomplete information from the opponent. Since only the position information of the opponent is known to each player, the optimal heading angle of the opponent is predicted in advance by each player to compute their strategy using Limited Information Model Predictive Control. The method integrated the game-theoretic paradigm with a Model Predictive control technique and has tackled the pursuit-evasion game intelligently in the presence of obstacles. The computation time of the second method is much lower than that of the first method and hence solves the problem of physical implementation.

7.4 Research summary based on objective IV

The fourth objective lies at the intersection of objectives II and III. A differential game of static target defense by a single defender was solved using a novel method called game-theoretic model predictive control. The problem of the differential game of static target defense resembled the game of guarding target discussed in Isaac's book of differential games. A Defender strives to intercept an attacker before it reaches the target, while the attacker tries to reach the target and avoid the defender. While several offline solutions to the problem are presented in the literature for holonomic systems with simple equations of motion. Others considered cooperative strategies between the defender and the target to defeat the attacker, the problem considered in this thesis is for nonholonomic mobile robots in the presence of obstacles where a single defender unilaterally players against the attacker. The solution approach proposed in this thesis is online that combines the concepts of NMPC and game theory as per objectives II and III respectively. The defender uses the knowledge of the attacker's intention to integrate a game-theoretic paradigm by incorporating a trade-off parameter in the control design. The parameter was regarded as a virtual reference that combines the tracking and the intercepting objectives. With the tuned value of the coefficient of parameter, a single defender was able to protect the target from the attacker despite having the same speed limits and being equidistant from each other. The proposed method was compared with other approaches to show its effectiveness.

7.5 Research summary based on objective V

The final objective of this thesis is to implement all the proposed controllers on a physical system for validation. Most of the works on game-theory stop at simulation, both numerical and virtual. The dynamic obstacles avoidance for a set-point stabilization problem was implemented using an NMPC controller which was designed on a remote PC and communicated the optimal strategy to the robot wirelessly. The feedback to the controller was captured using a set of cameras whereas another robot was regarded as a dynamic obstacle. The problem was extended from a point-stabilization problem to a pursuit of an evading robot. The evader is also controlled remotely by a controller whose objective conflicts with the objective of the pursuer. The game was played in a bounded environment with virtual obstacles positioned in strategic locations. Game-theoretic control approaches are then implemented on the two robots based on the level of information of the opponent on each player. The computation time and the capture time played an important role in selecting the strategy. Finally, game-theoretic model predictive control was implemented to handle the differential game of static target defense.

7.6 Summary of contributions

Precisely, the following are the key contributions of the thesis objectives in using online optimization and game-theoretic approaches in solving control problems for nonholonomic mobile robots.

- Contributions from objective I:
 - Developed a novel implementation of NMPC for dynamic obstacles avoidance.
 - Compared the performance of discretization methods for future users of discrete NMPC.
- Contributions from objective II:
 - Developed a novel implementation of NMPC for pursuit-evasion games.
 - Compared the performance of stabilizing methods for future users of NMPC.
- Contributions from objective III:
 - Developed novel game-theoretic methods for solving pursuit-evasion games.
 - Compared the performances of game-theoretic methods based on their advantages, limitations, and implementability.
- Contributions from objective IV:
 - Developed a novel implementation of game-theoretic model predictive control for the differential game of static target defense.

-
- Contributions from objective V:
 - Experimental validation of the developed NMPC algorithm for dynamic obstacles avoidance.
 - Experimental validation of the developed NMPC algorithms for pursuit-evasion problems.
 - Experimental validation of the developed game-theoretic algorithms for solving pursuit-evasion games.
 - Experimental validation of the developed game-theoretic model predictive control

7.6.1 Lists of publications

The following is the list of scientific papers published as a result of the research conducted in this thesis.

- Mukhtar Sani, Bogdan Robu, and Ahmad Hably, "Dynamic Obstacles Avoidance Using Nonlinear Model Predictive Control" in the proceedings of the 47th Annual Conference of the IEEE Industrial Electronics Society (IECON'2021), Toronto, Canada (2021).
- Mukhtar Sani, Bogdan Robu and Ahmad Hably, "Pursuit-evasion games for Nonholonomic Mobile Robots with Obstacles Avoidance Using Nonlinear Model Predictive Control" in the proceedings of the 28th Mediterranean Conference on Control and Automation (MED'2020), St Raphael, France (2020).
- Mukhtar Sani, Bogdan Robu, and Ahmad Hably, "Pursuit-evasion Games Based on Game-theoretic and Model Predictive Control Algorithms" in the proceedings of the 5th International Conference on Control, Automation and Diagnosis (ICCAD'2021), Grenoble, France (2021).
- Mukhtar Sani, Bogdan Robu, and Ahmad Hably, "Limited Information Model Predictive Control for Pursuit-evasion Games" in the proceedings of the 60th IEEE Conference on Decision and Control (CDC'2021), Texas, USA (2021).

7.7 Future research directions

The research work presented in this thesis has some possible future extensions for improving the works:

- Control of a single mobile robot with dynamic obstacles avoidance using NMPC can be extended by:
 1. Considering the dynamic model and include disturbances.

-
2. Considering stochastic NMPC as a possible solution approach.
 3. Considering learning-based NMPC instead of optimization based NMPC
 4. Implementing the algorithms on aerial and underwater systems.
- Pursuit-evasion game problem can be extended by:
 1. Considering cooperative pursuit and cooperative evasion problems.
 2. Considering robust MPC as a possible solution.
 - Differential game static target defense problem can be improved by:
 1. Considering an active target, so that cooperation between the defender and the moving target could defeat the attacker easily.
 2. Considering dynamic trade-off parameter α .
 - Implementation of the controllers on the physical system can be improved by:
 1. Embedding the controllers on the robots so that computations can be done on it.
 2. Using on-board cameras and sensors to provide feedback and detect obstacles.

Bibliography

- Abbas, Muhammad Awais, Ruth Milman, and J. Mikael Eklund (2017). “Obstacle Avoidance in Real Time With Nonlinear Model Predictive Control of Autonomous Vehicles.” In: *Canadian Journal of Electrical and Computer Engineering* 40.1, pp. 12–22 (cit. on p. 10).
- Acevedo, J. J. et al. (2013). “Cooperative perimeter surveillance with a team of mobile robots under communication constraints.” In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5067–5072 (cit. on p. 9).
- Aguiar, A. Pedro, João P. Hespanha, and Petar V. Kokotović (2008). “Performance limitations in reference tracking and path following for nonlinear systems.” In: *Automatica* 44.3, pp. 598–610. URL: <https://www.sciencedirect.com/science/article/pii/S0005109807003482> (cit. on pp. 12, 19).
- Alamir, M. and G. Bornard (1995). “Stability of a truncated infinite constrained receding horizon scheme: the general discrete nonlinear case.” In: *Automatica* 31.9, pp. 1353–1356. URL: <https://www.sciencedirect.com/science/article/pii/000510989500042U> (cit. on p. 40).
- Alamir, Mazen (2006). *Stabilization of Nonlinear Systems Using Receding-horizon Control Schemes: A Parametrized Approach for Fast Systems*. Springer (cit. on p. 18).
- (Jan. 2017). “Contraction-Based Nonlinear Model Predictive Control Formulation Without Stability-Related Terminal Constraints.” In: *Automatica* 75, pp. 288–292. URL: <https://hal.archives-ouvertes.fr/hal-01402122> (cit. on p. 40).
- (Jan. 2018). “Stability proof for nonlinear MPC design using monotonically increasing weighting profiles without terminal constraints.” In: *Automatica* 87, pp. 455–459. URL: <https://hal.archives-ouvertes.fr/hal-01664493> (cit. on p. 40).
- Alexis, Kostas, George Nikolakopoulos, and Anthony Tzes (2011). “Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances.” In: *Control Engineering Practice* 19.10, pp. 1195–1207. URL: <https://www.sciencedirect.com/science/article/pii/S0967066111001262> (cit. on p. 19).
- Alexis, Kostas et al. (2011). “Model predictive quadrotor indoor position control.” In: *2011 19th Mediterranean Conference on Control Automation (MED)*, pp. 1247–1252 (cit. on p. 19).
- Andersson, Joel A. E. et al. (2019). “CasADi: A software framework for nonlinear optimization and optimal control.” In: *Mathematical Programming Computation* 11. URL: <https://doi.org/10.1007/s12532-018-0139-4> (cit. on pp. 22, 41, 60, 73).
- Arai, T., E. Pagello, and L.E. Parker (2002). “Guest editorial advances in multirobot systems.” In: *IEEE Transactions on Robotics and Automation* 18.5, pp. 655–661 (cit. on p. 9).
- Backman, J., T. Oksanen, and A. Visala (2012). “Navigation system for agricultural machines: Nonlinear Model Predictive path tracking.” In: *Computers and Electronics in Agriculture* 82, pp. 32–43. URL: <https://www.sciencedirect.com/science/article/pii/S0168169911003218> (cit. on p. 12).

-
- Bakolas, Efstathios and Panagiotis Tsiotras (2010). “Optimal pursuit of moving targets using dynamic Voronoi diagrams.” In: *49th IEEE Conference on Decision and Control (CDC)*, pp. 7431–7436 (cit. on pp. 35, 69).
- Bartee, James et al. (2009). “Industrial application of nonlinear model predictive control technology for fuel ethanol fermentation process.” In: *2009 American Control Conference*, pp. 2290–2294 (cit. on p. 19).
- Basimanebotlhe, Othusitse and Xiaoping Xue (2014). “Stochastic optimal control to a nonlinear differential game.” In: *Advances in Difference Equations* 2014.1, pp. 1–14 (cit. on p. 52).
- Bera, Ritwik, Venkata Ramana Makkapati, and Mangal Kothari (2017). “A comprehensive differential game theoretic solution to a game of two cars.” In: *Journal of Optimization Theory and Applications* 174.3, pp. 818–836 (cit. on p. 51).
- Bernard, Markus et al. (2011). “Autonomous transportation and deployment with aerial robots for search and rescue missions.” In: *Journal of Field Robotics* 28.6, pp. 914–931. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.20401>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20401> (cit. on p. 9).
- Bhavesh, Vayeda Anshav (2015). “Comparison of Various Obstacle Avoidance Algorithms.” In: *International Journal of Engineering Research and Technology (IJERT)* (cit. on p. 17).
- Bloch, A. M. (2003). “Nonholonomic Mechanics.” In: *Nonholonomic Mechanics and Control*. New York, NY: Springer New York, pp. 207–276. URL: https://doi.org/10.1007/b97376_5 (cit. on pp. 13, 15).
- Borenstein, J. and Y. Koren (1988). “High-speed obstacle avoidance for mobile robots.” In: *Proceedings IEEE International Symposium on Intelligent Control 1988*, pp. 382–384 (cit. on p. 18).
- Bouffard, Patrick, Anil Aswani, and Claire Tomlin (2012). “Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results.” In: *2012 IEEE International Conference on Robotics and Automation*, pp. 279–284 (cit. on p. 19).
- Brockett, R. W. (1983a). “Asymptotic stability and feedback stabilization,” in: *Differential Geometric Control theory, Brockett, R. W., Millman, R. S, and Sussmann, H. J.(eds). Birkhauser*. Pp. 181–191 (cit. on pp. 11, 15).
- (1983b). “Nonlinear Control theory and Differential Geometry,” in: *International Congress of Mathematics*, pp. 1357–1368 (cit. on pp. 3, 13).
- Calise, Anthony J and Xiao-ming Yu (1985). “An analysis of a four state model for pursuit-evasion games.” In: *1985 24th IEEE Conference on Decision and Control*. IEEE, pp. 1119–1121 (cit. on p. 53).
- Castillo, P., A. Dzul, and R. Lozano (2004). “Real-time stabilization and tracking of a four-rotor mini rotorcraft.” In: *IEEE Transactions on Control Systems Technology* 12.4, pp. 510–516 (cit. on p. 10).
- Castillo-Lopez, Manuel et al. (2018). “Model Predictive Control for Aerial Collision Avoidance in Dynamic Environments.” In: *2018 26th Mediterranean Conference on Control and Automation (MED)*, pp. 1–6 (cit. on p. 10).
- Catapang, Angelo Nikko and Manuel Ramos (2016). “Obstacle detection using a 2D LIDAR system for an Autonomous Vehicle.” In: *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, pp. 441–445 (cit. on p. 18).

-
- Chemori, Ahmed and Nicolas Marchand (2008). “A prediction-based nonlinear controller for stabilization of a non-minimum phase PVTOL aircraft.” In: *International Journal of Robust and Nonlinear Control* 18.8, pp. 876–889. URL: <https://hal.archives-ouvertes.fr/hal-00176071> (cit. on p. 19).
- Chernousko, FL and AA Melikyan (1975). “Some differential games with incomplete information.” In: *Optimization Techniques IFIP Technical Conference*. Springer, pp. 445–450 (cit. on p. 52).
- Chwa, Dongkyoung (2004). “Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates.” In: *IEEE Transactions on Control Systems Technology* 12.4, pp. 637–644 (cit. on p. 12).
- Cortes, Patricio et al. (2008). “Predictive Control in Power Electronics and Drives.” In: *IEEE Transactions on Industrial Electronics* 55.12, pp. 4312–4324 (cit. on p. 19).
- Dalamagkidis, Konstantinos, Kimon P. Valavanis, and Les A. Piegl (2011). “Nonlinear Model Predictive Control With Neural Network Optimization for Autonomous Autorotation of Small Unmanned Helicopters.” In: *IEEE Transactions on Control Systems Technology* 19.4, pp. 818–831 (cit. on p. 19).
- Dixon, W.E. et al. (2000). “Global exponential tracking control of a mobile robot system via a PE condition.” In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 30.1, pp. 129–142 (cit. on pp. 12, 19).
- Dong, J., X. Zhang, and X. Jia (2012). “Strategies of Pursuit-Evasion Game Based on Improved Potential Field and Differential Game Theory for Mobile Robots.” In: *2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control*, pp. 1452–1456 (cit. on pp. 34, 51, 70).
- Eklund, J. M., J. Sprinkle, and S. S. Sastry (2012). “Switched and Symmetric Pursuit/Evasion Games Using Online Model Predictive Control With Application to Autonomous Aircraft.” In: *IEEE Transactions on Control Systems Technology* 20.3, pp. 604–620 (cit. on pp. 19, 35).
- Fahimi, F. (2007). “Non-linear model predictive formation control for groups of autonomous surface vessels.” In: *International Journal of Control* 80.8, pp. 1248–1259. eprint: <https://doi.org/10.1080/00207170701280911>. URL: <https://doi.org/10.1080/00207170701280911> (cit. on p. 19).
- Faulwasser, T., B. Kern, and R. Findeisen (2009). “Model predictive path-following for constrained nonlinear systems.” In: *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pp. 8642–8647 (cit. on pp. 10, 12).
- Faulwasser, Timm (2012). “Optimization-based Solutions to Constrained Trajectory-tracking and Path-following Problems.” PhD thesis. Otto-von-Guericke-Universität Magdeburg (cit. on pp. 11, 12).
- Feng, Xie (2004). “Model Predictive Control of Non-holonomic Mobile Robots.” PhD thesis. Oklahoma State university, USA (cit. on pp. 11–13, 39).
- Fisac, Jaime F and S Shankar Sastry (2015). “The pursuit-evasion-defense differential game in dynamic constrained environments.” In: *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, pp. 4549–4556 (cit. on p. 51).

-
- Fuchs, Zachariah E and Pramod P Khargonekar (2016). “Generalized engage or retreat differential game with escort regions.” In: *IEEE Transactions on Automatic Control* 62.2, pp. 668–681 (cit. on p. 51).
- Fuchs, Zachariah E., Pramod P. Khargonekar, and Johnny Evers (2010). “Cooperative defense within a single-pursuer, two-evader pursuit evasion differential game.” In: *49th IEEE Conference on Decision and Control (CDC)*, pp. 3091–3097 (cit. on p. 35).
- Fuseiller, G. et al. (2018). “Reactive path planning for collaborative robot using configuration space skeletonization.” In: *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, pp. 29–34 (cit. on p. 17).
- Garcia, Eloy, David W. Casbeer, and Meir Pachter (2019). “Design and Analysis of State-Feedback Optimal Strategies for the Differential Game of Active Defense.” In: *IEEE Transactions on Automatic Control* 64.2, pp. 553–568 (cit. on p. 70).
- Garcia, Eloy et al. (2017). “A Geometric Approach for the Cooperative Two-Pursuer One-Evader Differential Game.” In: *IFAC-PapersOnLine* 50.1. 20th IFAC World Congress, pp. 15209–15214. URL: <https://www.sciencedirect.com/science/article/pii/S240589631733183X> (cit. on pp. 35, 58, 69).
- Garcia, Eloy et al. (2019). “Cooperative Two-Pursuer One-Evader Blocking Differential Game.” In: *2019 American Control Conference (ACC)*, pp. 2702–2709 (cit. on pp. 35, 69).
- Garimella, Gowtham et al. (2018). “Robust Obstacle Avoidance using Tube NMPC.” In: *Robotics: Science and Systems* (cit. on p. 10).
- Getz, W. M. and M. Pachter (1981a). “Capturability in a two-target ‘game of two cars’.” In: *Journal of Guidance and Control* 4.1, pp. 15–21. eprint: <https://doi.org/10.2514/3.19715>. URL: <https://doi.org/10.2514/3.19715> (cit. on p. 70).
- Getz, Wayne M and Meir Pachter (1981b). “Capturability in a two-target ‘game of two cars’.” In: *Journal of Guidance and Control* 4.1, pp. 15–21 (cit. on p. 51).
- Getz, W.M and G Leitmann (1979). “Qualitative differential games with two targets.” In: *Journal of Mathematical Analysis and Applications* 68.2, pp. 421–430. URL: <https://www.sciencedirect.com/science/article/pii/0022247X79901264> (cit. on p. 70).
- Getz, WM and M Pachter (1981c). “Two-target pursuit-evasion differential games in the plane.” In: *Journal of Optimization Theory and Applications* 34.3, pp. 383–403 (cit. on p. 51).
- Giovannangeli, C et al. (2010). “Pursuit-Evasion Games in Presence of Obstacles in Unknown Environments: towards an optimal pursuit strategy.” In: *Cutting edge robotics* 2010, pp. 47–80 (cit. on p. 52).
- Greenfeld, I (1987). “A differential game of surveillance evasion of two identical cars.” In: *Journal of optimization theory and applications* 52.1, pp. 53–79 (cit. on p. 51).
- Greenwood, Nigel (1992). “A differential game in three dimensions: The aerial dogfight scenario.” In: *Dynamics and Control* 2.2, pp. 161–200 (cit. on p. 52).
- Gross, Matthew and Mark Costello (2014). “Impact point model predictive control of a spin-stabilized projectile with instability protection.” In: *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 228.12, pp. 2215–2225. eprint: <https://doi.org/10.1177/0954410013514743>. URL: <https://doi.org/10.1177/0954410013514743> (cit. on p. 19).

-
- Gu, Dongbing and Huosheng Hu (2005). “A stabilizing receding horizon regulator for non-holonomic mobile robots.” In: *IEEE Transactions on Robotics* 21.5, pp. 1022–1028 (cit. on pp. 10, 12, 19).
- (2006). “Receding horizon tracking control of wheeled mobile robots.” In: *IEEE Transactions on Control Systems Technology* 14.4, pp. 743–749 (cit. on pp. 10, 12).
- Hexner, G (1979). “A differential game of incomplete information.” In: *Journal of Optimization Theory and Applications* 28.2, pp. 213–232 (cit. on p. 52).
- Ho, Y., A. Bryson, and S. Baron (1965). “Differential games and optimal pursuit-evasion strategies.” In: *IEEE Transactions on Automatic Control* 10.4, pp. 385–389 (cit. on p. 34).
- Huang, Haomiao et al. (Dec. 2011). “Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers.” English (US). In: *2011 50th IEEE Conference on Decision and Control and European Control Conference, CDC-ECC 2011*. Proceedings of the IEEE Conference on Decision and Control. 2011 50th IEEE Conference on Decision and Control and European Control Conference, CDC-ECC 2011 ; Conference date: 12-12-2011 Through 15-12-2011, pp. 4835–4840 (cit. on p. 35).
- Imado, Fumiaki and Takeshi Kuroda (2005). “A method to solve missile-aircraft pursuit-evasion differential games.” In: *IFAC Proceedings Volumes* 38.1, pp. 176–181 (cit. on p. 52).
- Indiveri, G. (1999). “Kinematic time-invariant control of a 2D nonholonomic vehicle.” In: *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No.99CH36304)*. Vol. 3, 2112–2117 vol.3 (cit. on pp. 10, 12).
- Široký, Jan et al. (2011). “Experimental analysis of model predictive control for an energy efficient building heating system.” In: *Applied Energy* 88.9, pp. 3079–3087. URL: <https://www.sciencedirect.com/science/article/pii/S0306261911001668> (cit. on p. 19).
- Isaacs, R. (1956). *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*. Dover Books on Mathematics. Dover Publications. URL: <https://books.google.fr/books?id=GPjBAGAAQBAJ> (cit. on pp. 4, 33, 34, 50, 52, 70).
- Jiang-Dagger, Zhong-Ping and Henk Nijmeijer (1997). “Tracking Control of Mobile Robots: A Case Study in Backstepping**This paper was not presented at any IFAC meeting.” In: *Automatica* 33.7, pp. 1393–1399. URL: <https://www.sciencedirect.com/science/article/pii/S0005109897000551> (cit. on p. 12).
- Jualin, L. (2015). *Mobile Robotics*. Dover Books on Mathematics. Elsevier Publications, pp. 17–17 (cit. on pp. 2, 13).
- Kang, Yeonsik and J. Karl Hedrick (2009). “Linear Tracking for a Fixed-Wing UAV Using Nonlinear Model Predictive Control.” In: *IEEE Transactions on Control Systems Technology* 17.5, pp. 1202–1210 (cit. on p. 19).
- Kanjanawanishkul, Kiattisin and Andreas Zell (2009). “Path following for an omnidirectional mobile robot based on model predictive control.” In: *2009 IEEE International Conference on Robotics and Automation*, pp. 3341–3346 (cit. on p. 19).
- Kothari, Mangal, Joel G. Manathara, and Ian Postlethwaite (2014). “A Cooperative Pursuit-Evasion Game for Non-holonomic Systems.” In: *IFAC Proceedings Volumes* 47.3. 19th IFAC World Congress, pp. 1977–1984. URL: <https://www.sciencedirect.com/science/article/pii/S1474667016419038> (cit. on pp. 35, 58).

-
- Kothari, Mangal, Joel G. Manathara, and Ian Postlethwaite (2017). “Cooperative Multiple Pursuers against a Single Evader.” In: *Journal of Intelligent and Robotic Systems* 86.3, pp. 551–567. URL: <https://doi.org/10.1007/s10846-016-0423-3> (cit. on pp. 35, 69).
- Kouro, Samir et al. (2009). “Model Predictive Control—A Simple and Powerful Method to Control Power Converters.” In: *IEEE Transactions on Industrial Electronics* 56.6, pp. 1826–1838 (cit. on p. 19).
- Kuhne, F., W.F. Lages, and J.M.G. da Silva (2005). “Point stabilization of mobile robots with nonlinear model predictive control.” In: *IEEE International Conference Mechatronics and Automation, 2005*. Vol. 3, 1163–1168 Vol. 3 (cit. on pp. 10, 12, 19).
- Kumar, Amit and Aparajita Ojha (2019). “Experimental Evaluation of Certain Pursuit and Evasion Schemes for Wheeled Mobile Robots.” In: *International Journal of Automation and Computing* 16, pp. 491–510. URL: <https://doi.org/10.1007/s11633-018-1151-x> (cit. on p. 35).
- Lee, Ti-Chung et al. (2001). “Tracking control of unicycle-modeled mobile robots using a saturation feedback controller.” In: *IEEE Transactions on Control Systems Technology* 9.2, pp. 305–318 (cit. on pp. 12, 19).
- Leitmann, George (1968). “A simple differential game.” In: *Journal of Optimization Theory and Applications* 2.4, pp. 220–225 (cit. on p. 53).
- Lewin, Joseph and JV Breakwell (1975). “The surveillance-evasion game of degree.” In: *Journal of Optimization Theory and Applications* 16.3-4, pp. 339–353 (cit. on p. 51).
- Li, Shaosong et al. (2019). “Dynamic Trajectory Planning and Tracking for Autonomous Vehicle With Obstacle Avoidance Based on Model Predictive Control.” In: *IEEE Access* 7, pp. 132074–132086 (cit. on p. 10).
- Liang, Li et al. (2017). “Two coupled pursuit-evasion games in target-attacker-defender problem.” In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 5596–5601 (cit. on p. 70).
- Lim, Heonyoung et al. (2008a). “Nonlinear Model Predictive Controller Design with Obstacle Avoidance for a Mobile Robot.” In: *2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, pp. 494–499 (cit. on pp. 10, 12).
- (2008b). “Nonlinear Model Predictive Controller Design with Obstacle Avoidance for a Mobile Robot.” In: *2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, pp. 494–499 (cit. on pp. 10, 15).
- Lin, Ruitao et al. (2010). “A new numerical method of nonlinear equations by four order Runge-Kutta method.” In: *2010 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 1295–1299 (cit. on p. 16).
- Lio, Wai Hou, J.A. Rossiter, and Bryn L. Jones (2014). “A review on applications of model predictive control to wind turbines.” In: *2014 UKACC International Conference on Control (CONTROL)*, pp. 673–678 (cit. on p. 19).
- Lumelsky, V.J. and T. Skewis (1990). “Incorporating range sensing in the robot navigation function.” In: *IEEE Transactions on Systems, Man, and Cybernetics* 20.5, pp. 1058–1069 (cit. on p. 17).
- Lynch, K.M. and F.C. Park (2017). *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press. URL: <https://books.google.fr/books?id=8uS3QAACAAJ> (cit. on pp. 2, 3, 11, 13).

-
- Maestre, J.M. (Oct. 2010). “Distributed Model Predictive Control Based on Game Theory.” PhD thesis (cit. on pp. 48, 50).
- Marchal, Christian (1975). “Analytical study of a case of the homicidal chauffeur game problem.” In: *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974*. Ed. by G. I. Marchuk. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 472–481 (cit. on p. 51).
- Marden, Jason R. and Jeff S. Shamma (2018). “Game Theory and Control.” In: *Annual Review of Control, Robotics, and Autonomous Systems* 1.1, pp. 105–134. eprint: <https://doi.org/10.1146/annurev-control-060117-105102>. URL: <https://doi.org/10.1146/annurev-control-060117-105102> (cit. on pp. 33, 53).
- Margraff, Julien, Joanny Stéphant, and Ouiddad Labbani-Igbida (2020). “UAV 3D path and motion planning in unknown dynamic environments.” In: *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 77–84 (cit. on p. 17).
- Mayne, D. Q. et al. (June 2000). “Constrained model predictive control: Stability and optimality.” English (US). In: *Automatica* 36.6, pp. 789–814 (cit. on p. 39).
- M’Closkey, R.T. and R.M. Murray (1997). “Exponential stabilization of driftless nonlinear control systems using homogeneous feedback.” In: *IEEE Transactions on Automatic Control* 42.5, pp. 614–628 (cit. on pp. 10, 12).
- Mehrez, M. W. (2017). “Optimization Based Solutions for Control and State Estimation in Non-holonomic Mobile Robots: Stability, Distributed Control, and Relative Localization.” PhD thesis. Memorial University of Newfoundland, Canada (cit. on pp. 3, 11, 12, 20).
- Mehrez, M. W., G. K. I. Mann, and R. G. Gosine (2013). “Stabilizing NMPC of wheeled mobile robots using open-source real-time software.” In: *2013 16th International Conference on Advanced Robotics (ICAR)*, pp. 1–6 (cit. on pp. 10, 12, 19).
- Mehrez, Mohamed W., George K. I. Mann, and Raymond G. Gosine (2015). “Comparison of stabilizing NMPC designs for wheeled mobile robots: An experimental study.” In: *2015 Moratuwa Engineering Research Conference (MERCon)*, pp. 130–135 (cit. on pp. 10, 12, 19).
- Meier, L. (1969). “A new technique for solving pursuit-evasion differential games.” In: *IEEE Transactions on Automatic Control* 14.4, pp. 352–359 (cit. on p. 51).
- Merz, A. W. (1971). “The Homicidal Chauffeur - A Differential Game.” In: Guidance and Control Laboratory, Wright-Patterson AFB, OH, Tech. URL: <http://www.dtic.mil/dtic> (cit. on p. 50).
- (1974). “The Homicidal Chauffeur.” In: *AIAA Journal* 12.3, pp. 259–260. eprint: <https://doi.org/10.2514/3.49215>. URL: <https://doi.org/10.2514/3.49215> (cit. on p. 51).
- Merz, AW (1985). “To pursue or to evade—that is the question.” In: *Journal of guidance, control, and dynamics* 8.2, pp. 161–166 (cit. on p. 52).
- Meyers, Robert A (2012). *Computational complexity: theory, techniques, and applications*. Springer (cit. on p. 53).
- Mhaskar, Prashant (2006). “Robust Model Predictive Control Design for Fault-Tolerant Control of Process Systems.” In: *Industrial and Engineering Chemistry Research* 45.25, pp. 8565–8574. eprint: <https://doi.org/10.1021/ie060237p>. URL: <https://doi.org/10.1021/ie060237p> (cit. on p. 19).

-
- Michalek, Maciej and Krzysztof Kozłowski (2010). “Vector-Field-Orientation Feedback Control Method for a Differentially Driven Vehicle.” In: *IEEE Transactions on Control Systems Technology* 18.1, pp. 45–65 (cit. on pp. 12, 19).
- Mikumo, Ryuhei and Hiroyuki Ichihara (2017). “Dynamic collision avoidance among multiple mobile robots: A model predictive control approach.” In: *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pp. 1136–1137 (cit. on p. 10).
- Mitchell, I. M., A. M. Bayen, and C. J. Tomlin (2005). “A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games.” In: *IEEE Transactions on Automatic Control* 50.7, pp. 947–957 (cit. on p. 34).
- Morin, P. and Claude Samson (Jan. 2008). “Motion Control of Wheeled Mobile Robots.” In: pp. 799–826 (cit. on p. 3).
- Mosca, Edoardo (1991). “Adaptive optimal control: The thinking man’s gpc, R. R. Bitmead, M. Gevers and V. Wertz, Prentice-Hall, Englewood Cliffs, NJ, 1990, ISBN 0-13-013277-2, E32.95, xi + 244 pp.” In: *International Journal of Adaptive Control and Signal Processing* 5, pp. 349–350 (cit. on p. 39).
- Myerson, Roger B. (1977). “Graphs and Cooperation in Games.” In: *Mathematics of Operations Research* 2.3, pp. 225–229. eprint: <https://doi.org/10.1287/moor.2.3.225>. URL: <https://doi.org/10.1287/moor.2.3.225> (cit. on p. 50).
- Nash, John F. (1950). “Equilibrium Points in N-Person Games.” In: *Proceedings of the National Academy of Sciences of the United States of America* 36 1, pp. 48–9 (cit. on p. 53).
- Nielsen, Christopher, Cameron Fulford, and Manfredi Maggiore (2010). “Path following using transverse feedback linearization: Application to a maglev positioning system.” In: *Automatica* 46.3, pp. 585–590. URL: <https://www.sciencedirect.com/science/article/pii/S0005109810000294> (cit. on pp. 12, 19).
- Oh, So-Ryeok and Jing Sun (2010). “Path following of underactuated marine surface vessels using line-of-sight based model predictive control.” In: *Ocean Engineering* 37.2, pp. 289–295. URL: <https://www.sciencedirect.com/science/article/pii/S0029801809002431> (cit. on p. 19).
- Oliveira Kothare, S.L. de and M. Morari (2000). “Contractive model predictive control for constrained nonlinear systems.” In: *IEEE Transactions on Automatic Control* 45.6, pp. 1053–1071 (cit. on p. 40).
- Oriolo, G., A. De Luca, and M. Vendittelli (2002). “WMR control via dynamic feedback linearization: design, implementation, and experimental validation.” In: *IEEE Transactions on Control Systems Technology* 10.6, pp. 835–852 (cit. on pp. 10, 12).
- Oyler, Dave (Jan. 2016). “Contributions To Pursuit-Evasion Game Theory.” PhD thesis (cit. on p. 51).
- Oza, Harshal B. and Radhakant Padhi (2012). “Impact-Angle-Constrained Suboptimal Model Predictive Static Programming Guidance of Air-to-Ground Missiles.” In: *Journal of Guidance, Control, and Dynamics* 35.1, pp. 153–164. eprint: <https://doi.org/10.2514/1.53647>. URL: <https://doi.org/10.2514/1.53647> (cit. on p. 19).
- Pachter, M (1987). “The ‘Homicidal Chauffeur’ model in naval pursuit-evasion.” In: *Guidance, Navigation and Control Conference*, p. 2320 (cit. on p. 52).

-
- Pachter, Meir and Yaakov Yavin (1981). “A stochastic homicidal chauffeur pursuit-evasion differential game.” In: *Journal of Optimization Theory and Applications* 34.3, pp. 405–424 (cit. on p. 52).
- Pierson, Alyssa, Zijian Wang, and Mac Schwager (2017). “Intercepting Rogue Robots: An Algorithm for Capturing Multiple Evaders With Multiple Pursuers.” In: *IEEE Robotics and Automation Letters* 2.2, pp. 530–537 (cit. on pp. 35, 69).
- Portugal, David, Lino Marques, and Manuel Armada (Sept. 2014). “DEPLOYING FIELD ROBOTS FOR HUMANITARIAN DEMINING: CHALLENGES, REQUIREMENTS AND RESEARCH TRENDS.” In: pp. 649–656 (cit. on p. 9).
- Portugal, David and Rui Rocha (2011). “A Survey on Multi-robot Patrolling Algorithms.” In: *Technological Innovation for Sustainability*. Ed. by Luis M. Camarinha-Matos. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 139–146 (cit. on p. 9).
- Poundstone, William (1992). “Prisoner’s Dilemma: John Von Neumann, Game Theory and the Puzzle of the Bomb.” In: *Physics Today* 45, pp. 73–74 (cit. on p. 48).
- Prodan, Ionela et al. (Mar. 2013a). “Predictive Control for Trajectory Tracking and Decentralized Navigation of Multi-Agent Formations.” In: *International Journal of Applied Mathematics and Computer Science* 23.1, pp. 91–102. URL: <https://hal-supelec.archives-ouvertes.fr/hal-00826632> (cit. on p. 12).
- Prodan, Ionela et al. (2013b). “Receding horizon flight control for trajectory tracking of autonomous aerial vehicles.” In: *Control Engineering Practice* 21.10, pp. 1334–1349. URL: <https://www.sciencedirect.com/science/article/pii/S0967066113001020> (cit. on p. 12).
- Quinlan, S. and O. Khatib (1993). “Elastic bands: connecting path planning and control.” In: *[1993] Proceedings IEEE International Conference on Robotics and Automation*, 802–807 vol.2 (cit. on p. 18).
- Raffo, Guilherme V. et al. (2009). “A Predictive Controller for Autonomous Vehicle Path Tracking.” In: *IEEE Transactions on Intelligent Transportation Systems* 10.1, pp. 92–102 (cit. on pp. 10, 12).
- Rawlings, J. B., D. Q. Mayne, and M.M. Diehl (2019). *Model Predictive Control: Theory, Computation and Design*. 2nd Edition. Nob Hill Publishing, LLC (cit. on p. 18).
- Rosolia, Ugo, Stijn De Bruyne, and Andrew G. Alleyne (2017). “Autonomous Vehicle Control: A Nonconvex Approach for Obstacle Avoidance.” In: *IEEE Transactions on Control Systems Technology* 25.2, pp. 469–484 (cit. on p. 10).
- Roxin, Emilio and Chris P Tsokos (1970). “On the definition of a stochastic differential game.” In: *Mathematical systems theory* 4.1, pp. 60–64 (cit. on p. 52).
- Sani, M., B. Robu, and A. Hably (2020). “Pursuit-evasion Game for Nonholonomic Mobile Robots With Obstacle Avoidance using NMPC.” In: *2020 28th Mediterranean Conference on Control and Automation (MED)*, pp. 978–983 (cit. on pp. 6, 10, 15, 70).
- Sani, Mukhtar, Bogdan Robu, and Ahmad Hably (2021a). “Dynamic Obstacles Avoidance Using Nonlinear Model Predictive Control.” In: *47th Annual Conference of the IEEE Industrial Electronics Society (IES)* (cit. on p. 6).
- (2021b). “Limited Information Model Predictive Control for Pursuit-evasion Games.” In: *60th Conference on Decision and Control (CDC2021)* (cit. on p. 6).

-
- Sani, Mukhtar, Bogdan Robu, and Ahmad Hably (2021c). “Pursuit-evasion Games Based on Game-theoretic and Model Predictive Control Algorithms.” In: *ICCAD 2021: 5th IEEE International Conference on Control, Automation and Diagnosis* (cit. on p. 6).
- Shinar, J and S Gutman (1979). “Recent advances in optimal pursuit and evasion.” In: *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*. IEEE, pp. 960–965 (cit. on p. 52).
- Shinar, Josef (1980). *Solution techniques for realistic pursuit-evasion games*. Tech. rep. TECHNION-ISRAEL INST OF TECH HAIFA DEPT OF AERONAUTICAL ENGINEERING (cit. on p. 52).
- Shinar, Josef, Valery Y Glizer, and Vladimir Turetsky (2009a). “A pursuit-evasion game with hybrid evader dynamics.” In: *2009 European Control Conference (ECC)*. IEEE, pp. 121–126 (cit. on p. 52).
- (2009b). “A pursuit-evasion game with hybrid pursuer dynamics.” In: *European Journal of Control* 15.6, pp. 665–684 (cit. on p. 52).
- Souza, Cristino de et al. (2021). “Decentralized Multi-Agent Pursuit Using Deep Reinforcement Learning.” In: *IEEE Robotics and Automation Letters* 6.3, pp. 4552–4559 (cit. on p. 35).
- Sun, W. et al. (2017). “Pursuit-evasion games in dynamic flow fields via reachability set analysis.” In: *2017 American Control Conference (ACC)*, pp. 4595–4600 (cit. on p. 34).
- Taktak-Meziou, Manel et al. (June 2014). “Linear and nonlinear MPC for track following in the design of HDD servo systems.” In: *International Journal of Systems, Control and Communications* 6.1, pp. 20–47. URL: <https://hal-lirmm.ccsd.cnrs.fr/lirmm-01718130> (cit. on p. 10).
- Thrun, Sebastian, Wolfram Burgard, and Dieter Fox (Jan. 2005). “Probabilistic Robotics (Intelligent Robotics and Autonomous Agents).” In: (cit. on p. 15).
- Tolwinski, B. (1989). “Numerical solutions to differential games based on approximations by Markov games.” In: *Proceedings of the 28th IEEE Conference on Decision and Control*, 174–179 vol.1 (cit. on p. 34).
- Tzannetos, G., P. Marantos, and K. J. Kyriakopoulos (2016). “A competitive differential game between an unmanned aerial and a ground vehicle using model predictive control.” In: *2016 24th Mediterranean Conference on Control and Automation (MED)*, pp. 1053–1058 (cit. on pp. 35, 70).
- Vivas, Andrés and Philippe Poignet (2005). “Predictive functional control of a parallel robot.” In: *Control Engineering Practice* 13.7. Control Applications of Optimisation, pp. 863–874. URL: <https://www.sciencedirect.com/science/article/pii/S0967066104002084> (cit. on p. 19).
- Von Moll, Alexander et al. (2018). “Pursuit-evasion of an Evader by Multiple Pursuers.” In: *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 133–142 (cit. on pp. 35, 69).
- Von Neumann, John and Oskar Morgenstern (1947). *Theory of Games and Economic Behavior*. Second. New Jersey: Princeton University Press (cit. on p. 48).
- Wang, Liuping (2009). *Model Predictive Control Systems Design and Implementation Using MATLAB*. Springer (cit. on p. 18).

-
- Weintraub, I. E., M. Pachter, and E. Garcia (2020). “An Introduction to Pursuit-evasion Differential Games.” In: *2020 American Control Conference (ACC)*, pp. 1049–1066 (cit. on p. 33).
- Wit, C. Canudas de and O.J. Sordalen (1991). “Exponential stabilization of mobile robots with nonholonomic constraints.” In: *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*, 692–697 vol.1 (cit. on pp. 10, 12).
- Worthmann, Karl et al. (2015). “Regulation of Differential Drive Robots using Continuous Time MPC without Stabilizing Constraints or Costs.” In: *IFAC-PapersOnLine* 48.23. 5th IFAC Conference on Nonlinear Model Predictive Control NMPC 2015, pp. 129–135. URL: <https://www.sciencedirect.com/science/article/pii/S2405896315025562> (cit. on pp. 12, 19).
- Worthmann, Karl et al. (2016). “Model Predictive Control of Nonholonomic Mobile Robots Without Stabilizing Constraints and Costs.” In: *IEEE Transactions on Control Systems Technology* 24.4, pp. 1394–1406 (cit. on pp. 10, 12, 19).
- Xie, Feng and Rafael Fierro (2008). “First-state contractive model predictive control of non-holonomic mobile robots.” In: *2008 American Control Conference*, pp. 3494–3499 (cit. on pp. 10, 12, 19).
- Yavin, Yakoov (1986). “A pursuit-evasion differential game with noisy measurements of the evader’s bearing from the pursuer.” In: *Journal of optimization theory and applications* 51.1, pp. 161–177 (cit. on p. 52).
- Zhang, Jingjun et al. (2020). “A trajectory tracking and obstacle avoidance approach for nonholonomic mobile robots based on model predictive control.” In: *2020 IEEE 16th International Conference on Control Automation (ICCA)*, pp. 1038–1043 (cit. on p. 10).
- Zhou, Zhengyuan et al. (2016). “Cooperative pursuit with Voronoi partitions.” In: *Automatica* 72, pp. 64–72. URL: <https://www.sciencedirect.com/science/article/pii/S0005109816301911> (cit. on pp. 35, 69).