PSEUDO-SPECTRAL METHODS BASED REAL-TIME OPTIMAL PATH PLANNING FOR UNMANNED GROUND VEHICLES

by

DENISH KAMLESHKUMAR BAMAN

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2017

To my father Kamleshkumar Baman, my mother Sonal Baman, and my brother Dhaval for always believing in me.

ACKNOWLEDGEMENTS

This thesis would not have been possible without guidance and motivation from many people. First and Foremost, I would like to express my sincere gratitude to my advisor, Dr. Kamesh Subbarao, for providing me an opportunity to pursue graduate research, teaching me the knowledge that helps to take a diverse approach to solve the problem and encouraging me to give my best. His instructions to solve a problem with calm mind and dedication towards the desired goal made me a more capable Engineer.

I would like to thank committee members: Dr. Alan Bowling and Dr. Panayiotis Shiakolas. I am indebted towards Dr. Alan Bowling for teaching me new prospective to solve problems related to Multi-body dynamics when I was his student under "Analytical and Computation Dynamics" course. Dr. Panayiotis Shiakolas gave me some valuable feedback for my research after my thesis defense and I will never forget his encouragement.

My deepest gratitude to all my teacherss in the University of Texas at Arlington and back in India. To activate a hunger for knowledge and wisdom, inspiring me to plan my future and become a better individual.

I have my deepest appreciations to ASL folks Murali, Pengkai, Kelvin, Roopak, Rajnish, Ameya, Paul, Tracie, Abhishek, Karan, Pavan, Alok, Ziad, Diganta, and Abel. I would like to extend my special thanks to Murali for helping me to learn ROS and C++, to Pengkai for starting coding tradition in lab, to Aditya, Pradnya, Snehal, Jainam, and roomies, for being best buddies since last two years, to Jimmy, Salim, Sunny, Deesha, Kaminey, and Masti-makers, for wonderful memories to survive in

difficult time.

I would like to thank my parents and my family members for creating numerous opportunities to pursue my dreams and moral support throughout out my education, especially to my brother Dhaval, for love and support in every moment of my life. This degree is as much his as it is mine.

I would also like to thank my late grandparents for installing Sanskar through wonderful chat. They gave me purity, persistence, and perseverance as a heredity to built and achieve my dreams.

I also place on record, my sense of gratitude to one and all, who directly or indirectly, have lent their hand in this venture.

Finally, I would like to thank Air Force Research Laboratory (AFRL), for providing funding for the research via AFRL Award # FA9453-16-1-0058.

November 20, 2017

ABSTRACT

PSEUDO-SPECTRAL METHODS BASED REAL-TIME OPTIMAL PATH PLANNING FOR UNMANNED GROUND VEHICLES

DENISH KAMLESHKUMAR BAMAN, M.S

The University of Texas at Arlington, 2017

Supervising Professor: Kamesh Subbarao

Real-time optimal trajectory design and tracking for autonomous ground vehicles are maturing technologies with the potential to advance mobility by enhancing time and energy efficiency in application such as indoor surveillance robots or planetary exploration rovers. Pseudo-spectral methods based trajectory generation framework provides the desired trajectory which minimizes a prescribed objective function (i.e. minimum time, acceleration, and energy) while satisfying kinodynamics and various types of constraints (i.e. obstacle avoidance and smooth turning at waypoint transitions). In this thesis cyber-physical system architecture is used for the communication between rover-vehicle and the ground station. By using optimal state and control vector from trajectory generation module and by obtaining the state feedback values from the cyber-physical system architecture, a backstepping based controller provides commanded control values to complete the trajectory.

Combination of novel optimal trajectory framework (Guidance), modified backstepping controller (control) and cyber-physical system architecture makes the complete guidance navigation and control system. This thesis work elaborates, the efficacy

of the overall approach by performing several experimental test runs carried out with the rover vehicle equipped with GPS, compass, and wheel encoders.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

ABSTRACT vi

LIST OF ILLUSTRATIONS xi

LIST OF TABLES xv

Chapter Page

1. INTRODUCTION 1

1.1 Background 1

1.2 Literature Review 1

1.3 Problem Description and Contribution 3

1.4 Thesis Outline 6

2. Mathematical Prelimanary and System Description 7

2.1 General Formulation of Optimal Control Problem 7

2.1.1 Nonlinear Dynamics 7

2.1.2 Performance Index 7

2.1.3 Endpoint Constraints 8

2.1.4 Path Constraints 8

2.2 Numerical methods 10

2.2.1 Indirect Method 12

2.2.2 Direct Methods 12

2.3 Vehicle Kinematics 15

3. Analytical Approch 18

3.1 Minimum Time Optimal Trajector 18

3.2 Minimum Control effort 20
3.3 Simulation Results 21
3.4 Summary 30
4. Collocation Methods based Trajectory Optimization 31
4.1 Nonlinear optimization 31
4.2 Collocation method 32
4.3 Direct collocation method 33
4.3.1 Residual error 35
4.4 Constraints 36
4.4.1 Input constraints 36
4.4.2 Environmental constraints 36
4.4.3 Endpoint constraints 37
4.5 Performance index 37
4.5.1 Goal objective function 37
4.5.2 Final time minimization 38
4.5.3 Heading angle constraint 38
4.5.4 Actuator rate constraints 39
4.5.5 System constraints 39
4.6 Trajectory Optimization problem setup 40
4.7 Offline Simulation Results 43
4.8 Summary 54
5. Experimental Setup and System Architecture 55
5.1 Experimental Steup 55
5.2 CPS architecture 56
5.3 Relationship between Angular velocity and Motor PWM 58
5.4 Communication Delay Characterization 60

6. Trajectory Tracking Results 63
6.1 Trajectory tracking framework 63
6.2 Methode 1 : With GPS and Compass 64
6.3 Method 2: With Encoder and compass 74
6.4 Special Test case: 81
6.5 Summary 88
7. Summary, Conclusions and Future Work 89
7.1 Summary and Conclusions 89
7.2 Future Work 90
Appendix
A. Acronym 91
REFERENCES 93
BIOGRAPHICAL STATEMENT 97

LIST OF ILLUSTRATIONS

Figure		Page
1.1	Optimal trajectory framework	4
2.1	Trajectory optimization methods	10
2.2	Pseudo-spectral collocation points	14
2.3	Vehicle kinematics	16
3.1	Simulated optimal trajectory using analytical approach	22
3.2	Simulated optimal trajectory using numerical approach	22
3.3	Heading angle using analytical approach	23
3.4	Heading angle using numerical approach	24
3.5	Control inputs using analytical approach	25
3.6	Control inputs using numerical approach	25
3.7	Simulated optimal trajectory using analytical approach	26
3.8	Simulated optimal trajectory using numerical approach	27
3.9	Heading angle	28
3.10	Simulated optimal trajectory using analytical method	28
3.11	Control inputs using analytical approach	29
3.12	Control inputs using numerical approach	29
4.1	Collocation methods	33
4.2	Offline trajectory optimization framework	41
4.3	Simulated optimal trajectory for isosceles triangle	44
4.4	Simulated optimal trajectory test for parallelogram	45
4.5	Heading angle of WMR	46

4.6 Simulated optimal control test for isosceles triangle 47
4.7 Simulated optimal control test for parallelogram 48
4.8 Simulated optimal trajectory with obstacles test for isosceles triangle . 48
4.9 Simulated optimal trajectory with obstacles for parallelogram 49
4.10 Heading angle of WMR 50
4.11 Simulated optimal control test for isosceles triangle 50
4.12 Simulated optimal control for parallelogram 51
4.13 Simulated optimal trajectory with obstacles for isosceles triangle . . . 52
4.14 Simulated optimal trajectory with obstacles for parallelogram 52
4.15 Heading angle of WMR 53
4.16 Simulated optimal ccontrol test for isosceles triangle 53
4.17 Simulated optimal control test for parallelogram 54
5.1 WMR built at Aerospace System Laboratory, UTA 55
5.2 CPS architecture for the communication between rover and ground station 57
5.3 Obtained relationship between angular velocity of wheel and motor PWM 59
5.4 communication delay between the rover and ground station using wifi(indoor) 61
5.5 communication delay between the rover and ground station using 4G-LTE(outdoor) 62
6.1 Real-time trajectory tracking framework 63
6.2 Real time trajectory tracking using GPS for isosceles triangle 66
6.3 Real time trajectory tracking using GPS for parallelogram 67
6.4 Error in position between reference and actual trajectory of the WMR with GPS and Compass for state-feedback 68
6.5 Trajectory tracking in X - direction using GPS for isosceles triangle . . 68
6.6 Trajectory tracking in X - direction using GPS for parallelogram . . . 69
6.7 Trajectory tracking in Y - direction using GPS for isosceles triangle . . 70

6.8 Trajectory tracking in Y - direction using GPS for parallelogram . . . 70
6.9 Heading angle using GPS and Compass for isosceles triangle 71
6.10 Heading angle using GPS and Compass for parallelogram 72
6.11 Angular acceleration of right wheel using GPS 73
6.12 Control of the WMR 73
6.13 Real time trajectory tracking using Encoder and Compass for isosceles triangle 74
6.14 Real time trajectory tracking using Encoder and Compass for parallelogram 75
6.15 Error in position between reference and actual trajectory of the WMR with GPS and Compass for state-feedback 76
6.16 Trajectory tracking in X - direction using Encoder for isosceles triangle 76
6.17 Trajectory tracking in Y - direction using Encoder for isosceles triangle 77
6.18 Trajectory tracking in X - direction using Encoder for parallelogram . 78
6.19 Trajectory tracking in Y - direction using Encoder for parallelogram . 78
6.20 Heading angle using Encoder and Compass for isosceles triangle 79
6.21 Heading angle using Encoder and Compass for parallelogram 80
6.22 Angular acceleration of left wheel using Encoder 80
6.23 Angular acceleration of left wheel using Encoder 81
6.24 Real time trajectory tracking using Encoder and Compass for special case 82
6.25 Real time trajectory tracking using Encoder and Compass for special case 83
6.26 Error in position between reference and actual trajectory of the WMR 84
6.27 Trajectory tracking in X - direction using Encoder 84
6.28 Trajectory tracking in X - direction using Encoder for test case 2 . . . 85
6.29 Trajectory tracking in Y - direction using Encoder for test case 2 . . . 85
6.30 Trajectory tracking in Y - direction using Encoder for special case . . . 86

6.31 Heading angle using Encoder and Compass for special case 86
6.32 Heading angle using GPS and Compass 87
6.33 Angular acceleration of left wheel using Encoder for special case 87
6.34 Angular acceleration of left wheel using Encoder for special case 88

LIST OF TABLES

Table Page

4.1 Trajectory Optimization Problem count 43

CHAPTER 1

INTRODUCTION

1.1 Background

The Penatory exploration rover, autonomous car, and Wheeled Mobile Robot (WMR) are vehicles that are capable of sensing the environment and maneuver without any human interference. In an interest to develop robust autonomy; Guidance, Navigation, and Control for wheeled vehicles have become the core of recent research efforts. In last three decades, studies to resolve hurdles involved in the making of autonomous WMR have become the central goal of several research practices. The first major effort towards the development of autonomous vehicle was carried out by Carnegie Mellon University (CMU) with Navlab in 1980s [1, 2]. The extended research in CMU on Navlab series (Navlab 5) was the milestone for first autonomous long distance drive which was achieved in 1995. It is one of the incidents after which many auto companies inclined towards increasing autonomy. The DARPA challenge that was being initiated in 2004 boosted these efforts to achieve fully autonomous ground vehicles that are capable of substantial off-road driving with the constraint of limited time. Meanwhile, trajectory optimization techniques were introduced to achieve a certain level of optimality for autonomous ground vehicle system [3].

1.2 Literature Review

For trajectory optimization problem, a certain level of optimality can be achieved by analytical optimal control approaches. When the problem includes inequality constraints on control variables, the solution can be obtained from the application of

Pontryagin's Minimum Principle [4, 5]. In case of linear system, if only the upper and lower bounds are applied to the controls, then a minimum-time control history can be determined by bang-bang or hysteresis control behavior, in which the control value is saturated with discontinuous jumps between the extremes [6, 7]. Furthermore, in the case of complicated nonlinear higher dimensional dynamical systems with various path constraints, the application of Pontryagin's Minimum Principle (in general analytical approach) can become very difficult. As a result, the numerical method to solve an optimal control problem becomes more reliable. Indirect shooting method converts the optimal control problem into two point boundary value problem (TPBVP).

Based on the solution approach numerical methods are divided into two different categories: Indirect approach and Direct approach. In an indirect method, the calculus of variation is employed to obtain the first-order optimality conditions for the optimal control problem. The indirect approach leads to a TPBVP (for a complex scenario, a multiple-point boundary-value problem). It provides accurate results for the problems with simple control and system constraints scenario, but in case of environmental constraints it is too complicated to solve. On the other hand, a direct method solves the optimal control problem by transcribing an infinite-dimensional optimization problem (Partial Differential Equation - PDE) to a finite-dimensional optimization problem (Ordinary Differential Equation - ODE). It obtains the solution by direct minimization of the performance objective, subject to the constraints of the optimal control problem. Various approaches using parametric optimization and direct collocation can be found in [8, 9]. Real-time trajectory generation by search and interpolation over a large trajectory database trajectory can be found in Atkeson [10]. Pseudo-Spectral (PS) collocation method is a numerical approach that extends the direct collocation method to improve the convergence rates by using an

adaptive method for selection of collocation points and basis functions [11, 12].

On the other hand, well-known Graph search methods (2D A* or D* algorithm) are capable of providing the feasible trajectory that satisfies the environmental constraints. Due to computational advancements, an algorithm can provide the updated optimal trajectory in a fraction of the time. But these consist of straight line segments and are unable to satisfy the system kinematic constraints without significant modification. During real-time trajectory tracking, WMR can exclusively execute either strictly translational or rotational movement (i.e. it can drive straight or turn on the spot) [13, 14, 15, 16]. Incremental search techniques (Rapidly-exploring Random Trees - RRT) is an efficient method for finding feasible trajectories for high-dimensional non-holonomic systems. RRT* is an algorithm that extends RRT to provide steeering functions for non-holonomic systems and RRT^X extends RRT* algorithm to allow for real-time incremental replanning when obstacle region changes [17, 18, 19, 20].

1.3 Problem Description and Contribution

There are many challenges involved with the making of an autonomous ground vehicle and this thesis work will present some theory and experimental results to overcome some of these issues. The main outcome of the research involves the derivation of guidance and control algorithms as well as the development of a reliable system framework to test. The research work addresses the guidance law of a vehicle in the form of optimal trajectory design, which is followed by the derivation of a stable nonlinear control law. Then, the implementation of this framework with navigation sensors will be presented to show that it drives the system along the desired reference trajectory.

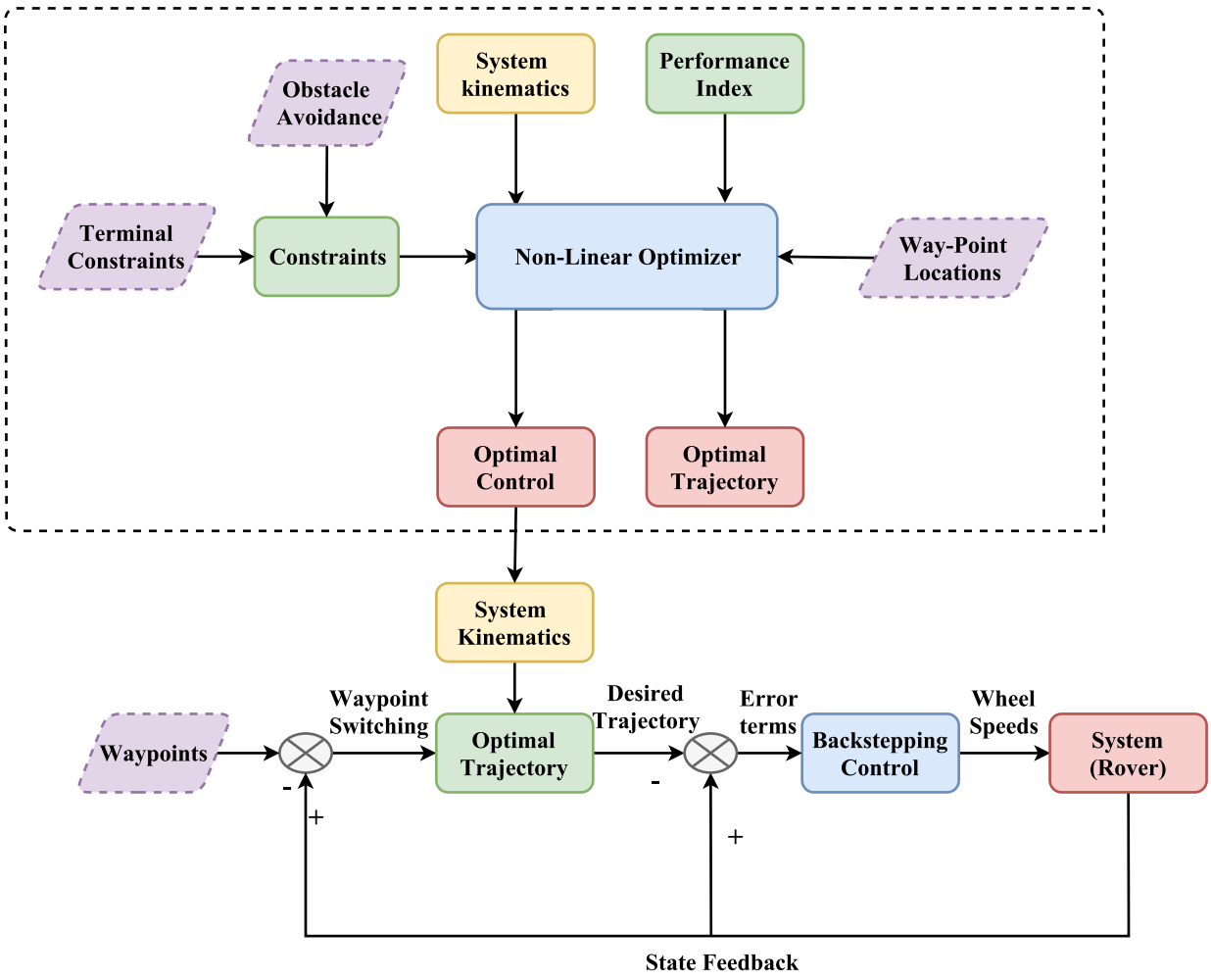

Figure 1.1: Optimal trajectory framework

Systems governed by differential equations are often encountered in multiple fields such as astronautics, aeronautics, and robotics. These equations are generated to describe the dynamics of time-variant systems. From the equations, a designer can simulate open-loop solutions or closed-loop solution by taking into account uncertainties in order to perform specified tasks and also maintain the desired performance requirements. Now the goal becomes finding the necessary control input which minimizes the deviation of the states from the optimal states and controls.

The optimal trajectory and control for any nonlinear dynamical system can be achieved by following optimal trajectory framework illustrated in Fig. 1.1. The trajectory design discussed in this document will show a formulation that is motivated by solving an optimal trajectory design problem for a differential drive vehicle. The trajectory generator designs the reference trajectory from a user-defined initial position to goal position. WMR completes the maneuver in the least possible time while attaining the waypoints (smooth turning at waypoint transition) and avoiding any obstacles along the path with minimum control effort (smooth vehicle velocity) and heading angle change (smooth steering). Initial and final states and control (Pulse Width Modulation - PWM) saturation of WMR can be modified at waypoints based on the requirement. The user can test the robustness of algorithm by increasing the constraints (obstacles) or provide challenging waypoint positions.

The trajectory generator will provide the reference control values for a WMR that can be tracked by the controller. So, the next aspect of this work will address, is the implementation of a nonlinear controller that will ensure that there is bounded tracking of the trajectory. The control law is to provide the angular velocity commands to the wheels of the WMR to ease its implementation on the experimental WMR setup. Therefore, one of the main objectives of this work is to design and implement the optimal reference trajectory along with the backstepping control law on an autonomous WMR to travel outdoor through the number of user-defined waypoints and virtual obstacles.

The experimental setup of a WMR which is constructed to function as a cyber-physical system (CPS) with an embedded system to handle the physical processes on the WMR and a networked ground station computer to process the system's information. Thus, another contribution of this work is the validation of a CPS architecture

deployed with the experimental setup and its ability to use the trajectory and control designs successfully.

1.4 Thesis Outline

The thesis is organized as follows: In Chapter 2, some preliminary concepts are introduced including formulation of Optimal Control Problem (OCP), trajectory optimization techniques, vehicle kinematics and other basic definitions. Chapter 3 introduces analytical approaches for trajectory optimization. Chapter 4 provides detailed information about nonlinear programming (NLP) and different collocation techniques. Aspects about constraint incorporation and performance index are provided. Offline trajectories generated using a PS method is simulated and the results are discussed. Experimental setup and system architecture for real-time implementation is described in chapter 5. Real time trajectory tracking results are discussed in Chapter 6. The effect of communication delay and various feedback methods on real time trajectory tracking are also displayed. Finally, in Chapter 7, concluding remarks are stated.

CHAPTER 2

Mathematical Prelimanary and System Description

Optimal control theory deals with the problem of finding a control law for a given system such that a certain performance objectives is achieved.

2.1 General Formulation of Optimal Control Problem

Optimal control theory deals with the input of the dynamic system that optimizes (i.e. minimize or maximize) a specific objective function while satisfying various constraints on the system. General formulation of optimal control problem has been extensively discussed in [21, 4] that included.

2.1.1 Nonlinear Dynamics

The dynamic constraints or equations of motion (differential equations that describe the system dynamics) are allowed to take the form of general nonlinear differential equations,

$$\dot{\mathbf{X}}(t) = f\left(\mathbf{X}(t), \mathbf{U}(t), t; \mathbf{P}\right), \tag{2.1}$$

where $\mathbf{X}(t) \in \mathbb{R}^n$ is the state vector, $\mathbf{U}(t) \in \mathbb{R}^m$ is the control vector, and $\mathbf{P} \in \mathbb{R}^l$ is the parameters of the system. The time is t, initial time t_0 and final time t_f.

2.1.2 Performance Index

Performance index is the measure of the 'quality' of the path or trajectory [21]. It shall be function of terminal condition, terminal constraints and the integral of

some function of states, controls, system parameters and time evaluated over entire time domain. Consider the Bolza cost function J given as,

$$J = \Phi\left(\mathbf{X}(t_0), t_0, \mathbf{X}(t_f), t_f; \mathbf{P}\right) + \int_{t_0}^{t_f} \mathcal{L}\left(\mathbf{X}(t), U(t), t; \mathbf{P}\right) dt, \tag{2.2}$$

where the terms Φ and $\mathcal{L}$ are terminal objective (endpoint or Mayer cost) and Lagrangian (Lagrange function) respectively. In function minimization problem, lower J gives better results; conversely, in function maximization problem, higher J gives good results.

2.1.3 Endpoint Constraints

The states and time domain may be constrained at initial and final condition by providing upper and lower bounds to general function,

$$\begin{aligned} \phi^0_{min} \leq \phi\left(\mathbf{X}(t_0), t_0\right) \leq \phi^0_{max} \\ \phi^f_{min} \leq \phi\left(\mathbf{X}(t_f), t_f\right) \leq \phi^f_{max} \end{aligned} \tag{2.3}$$

It is also known as boundary condition or terminal condition.

2.1.4 Path Constraints

Based on performance objectives, there may be some constraints that should not be violated along the trajectory. These constraints may be related to states, controls or system parameters on which states and controls depends are called as path constraint. Path constraint may be enforced through upper and lower bounds,

$$\mathbf{g_{min}} \leq \mathbf{g}[\mathbf{X(t)}, \mathbf{U(t)}, \mathbf{t}; \mathbf{P}] \leq \mathbf{g_{max}}, \tag{2.4}$$

Path constraints are in general inequality constraints and thus may not be active (i.e. nearly equal to zero) at the optimal solution. If the problem is initialized with bounds then it is called as Box constraint. Box constraints and Path constraints both are

similar function.

The state vector, control vector and system parameters vector can be written by

$$\mathbf{X}(t) = \begin{bmatrix} x_1(t) \\ . \\ . \\ . \\ x_n(t) \end{bmatrix} \quad \mathbf{U}(t) = \begin{bmatrix} u_1(t) \\ . \\ . \\ . \\ u_m(t) \end{bmatrix} \quad \mathbf{P}(t) = \begin{bmatrix} p_1(t) \\ . \\ . \\ . \\ p_l(t) \end{bmatrix}$$

Before approaching the trajectory optimization problem, we have to discern the difference between the terms trajectory optimization and optimal control as defined in Ref. [21]. These terms are almost the same and more often used interchangeably. The problem where the inputs to the system are static parameters, the task is to determine the values of these parameters and the trajectory that optimizes a given performance index also known as trajectory optimization. It is applicable to higher-dimensional path problems such as satellite, ground vehicle or robotic arms with more degrees of freedom. In contrast, the problem where the inputs to the system are themselves functions and the task is to determine the particular input function and trajectory that optimize a given performance index also known as an optimal control. It is applicable to lower-dimensional problems such as two-dimensional vehicles, where the control can be a function of time or state.

Certain class of trajectory optimization are easily solved by analytical optimal control. When problem consists of inequality constraints or is based on higher dimensional dynamical system then the numerical approaches are the best way to solve the problem. Fig. 2.1 illustrates different approaches to solve the optimal control problem [6, 21]. In this chapter, we will talk about different approaches to solve the particular trajectory optimization problem.

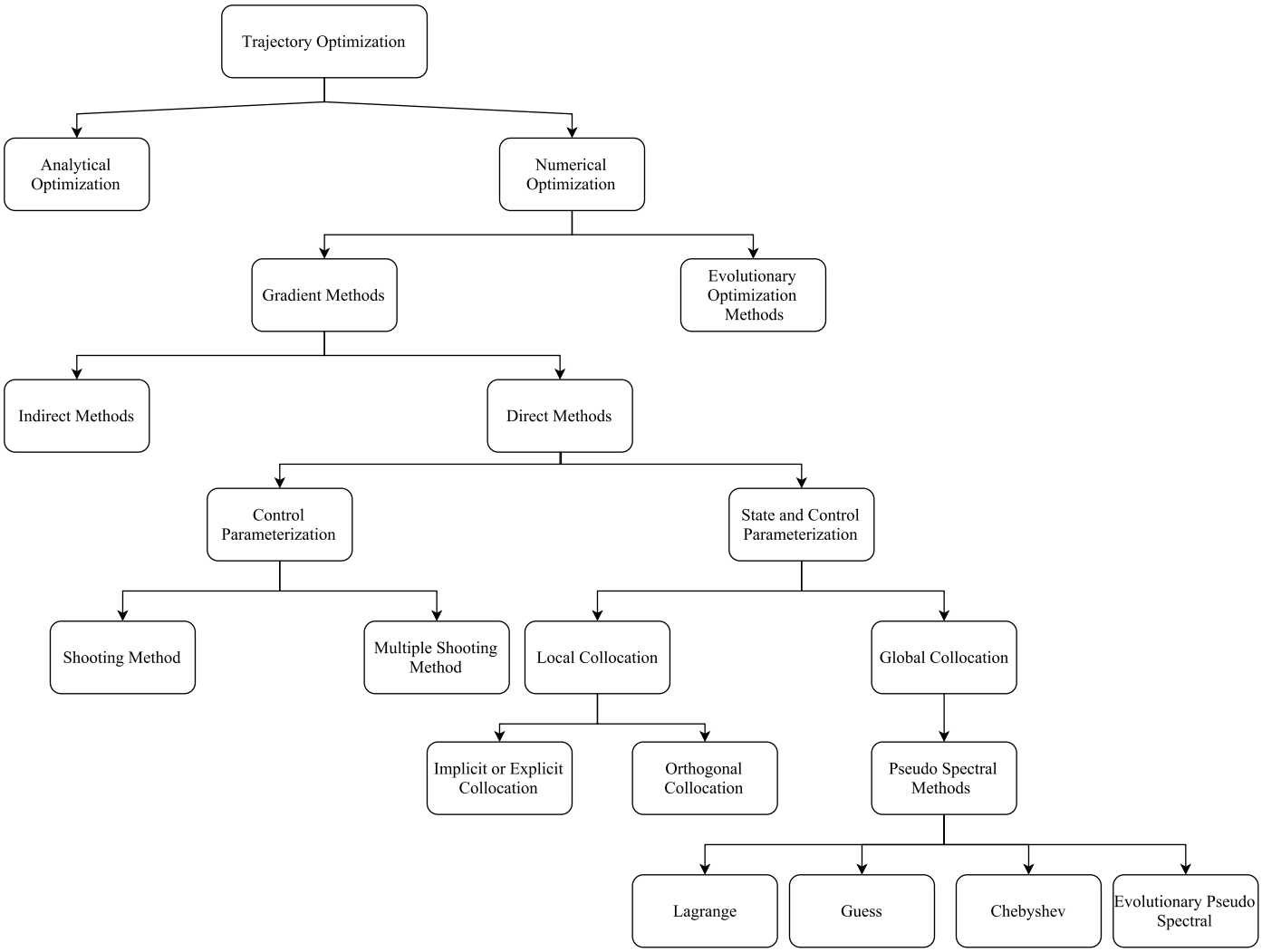

Figure 2.1: Trajectory optimization methods

2.2 Numerical methods

Numerical methods for solving optimal control problems took a leap nearly around the 1950s with the work of mathematician Richard Bellman. A significant contribution of Richard Bellman is a "necessary condition" for optimality associated with the mathematical optimization method known as dynamic programming also known as Bellman equation. Initially, Bellman equation was applied to engineering control theory and applied mathematics, which later became an important tool in

the field of Economic Theory [4, 21, 22, 23]. From that time to the present, the complexity of methods and variety of applications has increased tremendously making optimal control a discipline that is relevant to many branches of engineering and other disciplines.

Various numerical methods are available in the littrature for solving optimal control problems[4, 5, 21]. Numerical methods to solve optimal control problems are divided into two different categories: Indirect method and Direct method.

In an indirect method, the calculus of variation is employed to obtain the first-order optimality conditions of the optimal control problem. The indirect approach leads to a TPBVP (for a complex scenario, a multiple-point boundary-value problem). The boundary-value problem can be solved by taking derivative of a Hamiltonian. Thus, the resulting dynamic system is of the form

$$\begin{aligned} \dot{\mathbf{X}} &= \frac{\partial H}{\partial \boldsymbol{\lambda}} \\ \dot{\boldsymbol{\lambda}} &= \frac{\partial H}{\partial \boldsymbol{\lambda}} \end{aligned} \tag{2.5}$$

where,

$$H = \boldsymbol{\mathcal{L}} + \boldsymbol{\lambda}^T \mathbf{f} - \boldsymbol{\mu}^T \mathbf{g} \tag{2.6}$$

is augmented Hamiltonian. $\mathcal{L}$ is Lagrangian. $\boldsymbol{\mu} \in \mathbb{R}^{\mathbf{g}}$ is Lagrange multipliers associated with path constraint. Indirect methods use costate or adjoint $\boldsymbol{\lambda} \in \mathbb{R}^{\mathbf{f}}$ to eliminate the control variable $\mathbf{U}$. The co-states will introduce additional boundary value problems and algorithms such as "shooting" methods are available to solve the problem. This method works better for problems with simple control and no path constraints.

In a direct method, the state vector $\mathbf{X}$ and control vector $\mathbf{U}$ of the optimal control problem is discretized using some technique (i.e. state or control parameterization

technique). The approach is known as a nonlinear optimization problem or NPL. The NLP is then solved using well known optimization techniques (i.e. SNOPT, NPSOL, KNITRO, fmincon) [21, 24, 25, 26]. The direct method is mainly used by researchers, who are interested in trajectory optimization or optimal control techniques.

Based on Indirect and Direct philosophy, several methods are derived and used to solve optimal control problems are as follows:

2.2.1 Indirect Method

There are many type of indirect methods, but we are only concerned with the shooting method. In shooting method, an initial guess for endpoint boundary condition is used with some initial condition to integrate the trajectory (forward integration t_0 to t_f or backward integration t_f to t_0). The integration loop tries to achieve desired terminal condition. If the integrated terminal conditions differ from the known terminal conditions by a specified tolerance, then unknown initial conditions are adjusted and the process is repeated until the difference becomes lesser than a specified threshold.

2.2.2 Direct Methods

Direct Shooting Method

Shooting method is a basic method to solve the optimal control problem using the direct approach. Direct shooting method also known as control parametrization technique,where the control vector is parametrized using a specified functional form,

$$\mathbf{U}(t) \approx \sum_{i=1}^{m} a_i \zeta_i(t) \tag{2.7}$$

where $\zeta(t)$ is known function and a_i are the parameters to be determined from the optimization. The dynamics are satisfied by integrating the differential equations us-

ing a time marching algorithm. The cost function is determined using a quadrature approximation that is consistent with the numerical integrator used to solve the differential equations. It can minimize the cost subject to any path and interior-point constraints.

Differential Inclusion

Differential inclusion is a strictly direct method. The method of differential inclusions enforces the equation of motion at each discrete node by applying inequality constraints on the state derivatives. These inequality constraints are obtained by substituting the upper and lower bound on the control vector. When the inequality constraints are satisfied, the state vector at each node is said to lie in an attainable set.

Direct Collocation Method

Direct collocation methods are the most powerful methods for solving the general optimal control problem. It transcribes the dynamic optimization problem into the nonlinear parameter optimization problem. It is a state and control parameterization method where the state and control are approximated using a special function form. We will discuss direct collocation in detail in later chapters.

Pseudo Spectral Method

PS methods are a class of numerical methods used in applied mathematics and scientific computing for the solution of partial differential equations, also known as discrete variable representation (DVR) methods. Initially, it was developed to solve problems in Computational Fluid Dynamics (CFD) [21, 27]. Over last three decades, due the improvement in the computational power and improved convergence rates of PS methods compared to basic approaches, it has acquired high prestige among the numerical solution techniques applied to the optimal control problem.

A PS method is a global form of orthogonal collocation [11, 21, 28, 29], which is

similar to direct collocation method (a local solution), that transcribes optimal control problem to a NLP and searches for a global solution. It parametrizes the state and control using global polynomial and collocates the differential-algebraic equations at nodal points.

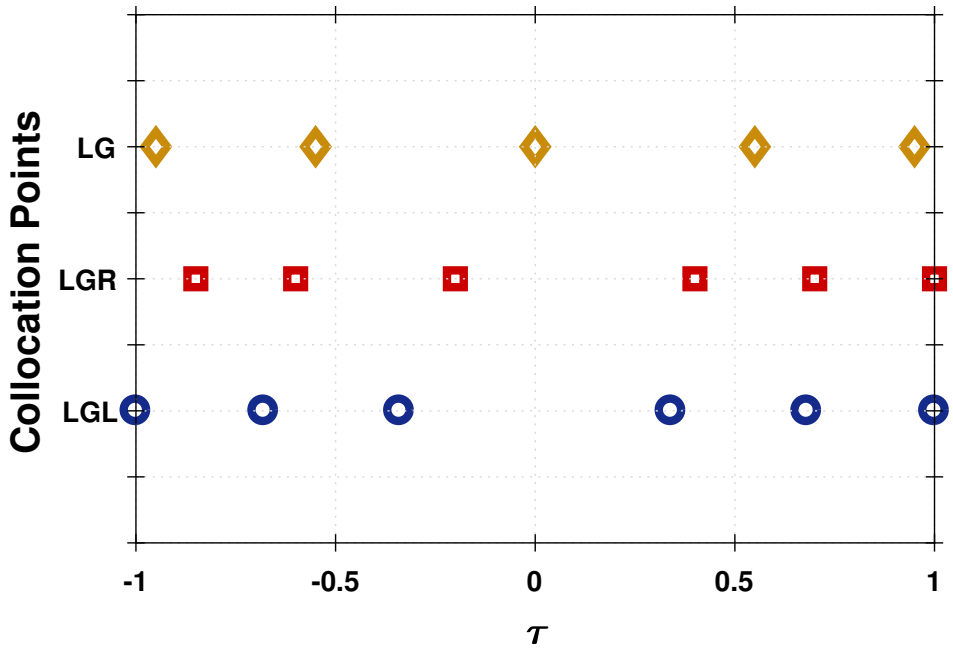

Figure 2.2: Pseudo-spectral collocation points

During the 1990s, PS methods were introduced for OCP with the basis functions of Chebyshev polynomial (Chebyshev Points) which is given in the Ref. [30, 31]. However, Chebyshev polynomial is not able to satisfy the isolation property, which leads to more complicated collocation conditions. As a result, use of Lagrange polynomials as basis functions for PS trajectory become popular as given in Ref [21]. Based on Lagrange polynomial and extended Gauss quadratures, most popular PS methods are Lagrange Gauss (LG), Lagrange Gauss Radau (LGR), and Lagrange

Gauss Lobatto (LGL) methods. LG quadrature points are distributed on the interior of the interval but do not include the endpoints. On the other hand, LGR quadrature points include one endpoint and LGL quadrature points include both endpoints of the interval [11, 12].

Based on the type of direct method used, the length of the optimization problem can be small (i.e. as in a direct shooting or Quasi-linearization method), medium (i.e. PS method [32]) or may be large (i.e. a direct collocation method) [26].
Several previous work that provide the basic understanding of trajectory optimization, optimal control and PS method had been published in open literature [26, 32].

2.3 Vehicle Kinematics

The kinematic model of differential drive vehicle is used for the optimal trajectory generation problem. The motion of systems considered without the external forces affected to the system. The system kinematics is illustrated in Fig. 2.3.

- Inertial reference frame and point: N is inertial reference point and ($x_{inertial}$, $y_{inertial}$, $z_{inertial}$) is inertial reference frame. $x_{inertial}$, $y_{inertial}$, $z_{inertial}$ are directed towards east, north and upward respectively.
- Body frame: (x_{body}, y_{body}, z_{body}) is body frame. x_{body}, y_{body}, z_{body} are directed towards front, left side and upward to the vehicle respectively.
- The heading angle, ψ is positive, anti-clockwise direction about $z_{inertial}$ axis.
- The transformation from the inertial frame to the body frame is given as,

$$R_B^N = \begin{bmatrix} cos\psi & -sin\psi & 0 \\ sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

where R_B^N is rotational matrix from inertial frame to body frame.

- The nonlinear differential equations that governs the WMR:

$$\begin{aligned} \dot{x} &= \frac{r}{2}(\omega_r + \omega_l)\cos(\psi) \\ \dot{y} &= \frac{r}{2}(\omega_r + \omega_l)\sin(\psi) \\ \dot{\psi} &= \frac{r}{b}(\omega_r - \omega_l) \end{aligned} \tag{2.9}$$

  where r is the wheel radius and b is the base length of the vehicle.

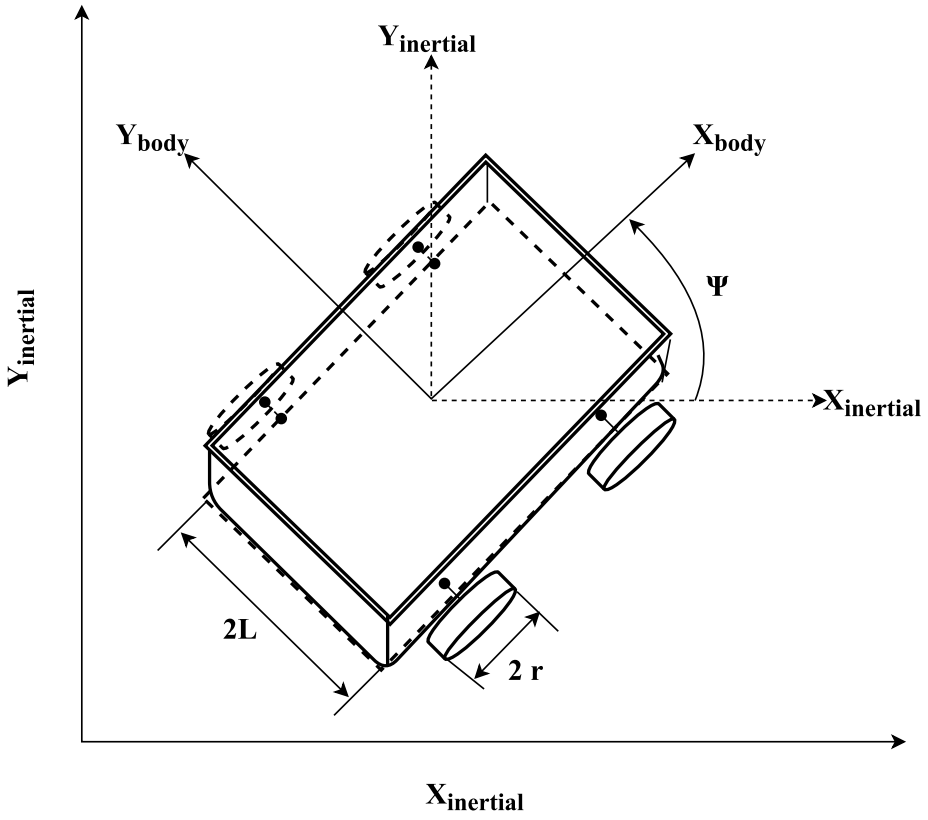

Figure 2.3: Vehicle kinematics

- The state vector of the rover model is defined as the Cartesian position in x and y coordinates and its heading angle ψ, i.e. $\mathbf{X} = [x\ y\ \psi]^T$.
- The control vector for the system is the PWM of the right and left wheels, i.e. $\mathbf{U} = [U_r\ U_l]^T$. ω_L and ω_R are the wheel speeds for the left and right

wheels, which are calculated from angular velocity and motor PWM relationship described in Sec. 5.3.

CHAPTER 3

Analytical Approch

The trajectory optimization problem can be solved by using an analytical approach to achieve a certain level of optimality. Different methods to find optimal trajectory are summarized in following sections:

3.1 Minimum Time Optimal Trajector

One objective is to design a path for WMR to maneuver from an arbitrary initial position to a final position in least possible time. The cost function for this problem is

$$J = \int_{t_0}^{t_f} dt, \tag{3.1}$$

The differential equation that governs the dynamical system is given in Eq. 2.9. As the system needs to reach the goal position, the deviation between the state at final time and the goal position has to be minimized. So, the final performance objective after adding the terminal objective function is,

$$J = \Phi\left(\mathbf{X}(t_f), t_f, \mathbf{P}\right) + \int_{t_0}^{t_f} \left(1 \; + \; \boldsymbol{\lambda}^T \mathbf{f}\left(\mathbf{X}(t), \mathbf{U}(t), t, \mathbf{P}\right)\right) dt, \tag{3.2}$$

As the path needs to respect the system kinematics, the Lagrange multipliers is introduced. In some cases, the Lagrange multipliers may eliminate the control vector to simplify the problem. The Hamiltonian H for this case is defined as,

$$H = 1 \; + \; \boldsymbol{\lambda}^T \mathbf{f}\left(\mathbf{X}(t), \mathbf{U}(t), t, \mathbf{P}\right),$$

As per the fundamental theorem of calculus of variation, for a trajectory to be considered an extremized solution, the first order variation of the cost (δJ) for

all admissible variation in states ($\delta\mathbf{X}$) , control ($\delta\mathbf{U}$) and Lagrange multiplier ($\delta\boldsymbol{\lambda}$) around the trajectory has to be zero. This results in,

$$\mathbf{H}_X^T = -\dot{\boldsymbol{\lambda}},$$

called the adjoint equation,

$$\mathbf{H}_U^T = 0,$$

called the control equations, and

$$\mathbf{H}_\lambda^T = \dot{\mathbf{X}},$$

called the traversability condition. Where, $\mathbf{H_X} \in \mathbb{R}^\mathbf{n}$, $\mathbf{H_U} \in \mathbb{R}^\mathbf{m}$ and $\mathbf{H}_\lambda \in \mathbb{R}^\mathbf{n}$ are a vectors of partial derivatives with respect to $\mathbf{X}, \mathbf{U}$ and $\boldsymbol{\lambda}$. The necessary condition provides six differential equations,

$$\begin{aligned}
\dot{x} &= \upsilon \cos(\psi) \\
\dot{y} &= \upsilon \sin(\psi) \\
\dot{\psi} &= \omega \\
\dot{\lambda}_x &= 0 \\
\dot{\lambda}_y &= 0 \\
\dot{\lambda}_\psi &= \upsilon(\lambda_x \sin\psi - \lambda_y \cos\psi)
\end{aligned} \tag{3.3}$$

and the six boundary conditions to solve this equations are

$$\begin{aligned}
x(t_0) &= x_0 \\
y(t_0) &= y_0 \\
\psi(t_0) &= \theta_0 \\
\lambda_x(t_f) &= 2(x_V(t_f) - x_G) \\
\lambda_y(t_f) &= 2(y_V(t_f) - y_G) \\
\lambda_\psi(t_f) &= \zeta
\end{aligned} \tag{3.4}$$

where, ζ is the heading angle at time t_f, $x_V(t_f)$ and $y_V(t_f)$ are the position of the vehicle at final time, x_G and y_G denote the desired goal position . The Lagrange multiplier for the final time is derived from the following terminal constraint equation,

$$\Phi_f = (x_V(t_f) - x_G)^2 + (y_V(t_f) - x_G)^2 + \zeta\ \psi_V(t_f), \tag{3.5}$$

From the differential equations given in Eq. (3.3), boundary conditions given in Eq. (3.4) and terminal constraint given in Eq. (3.5), we can solve for the minimum time optimal trajectory for a differential drive robot using Pontryagin's minimum principle. If in contrast the control values are bounded then the resulting solution will be a bang-bang type control law.

Similarly, we can derive the trajectory optimization problem for a minimum control effort objective function.

3.2 Minimum Control effort

The objective here is to design a path for WMR to maneuver from an arbitrary initial position to a final position by using minimum control effort. By enforcing cost on the actuator rate, a minimum control effort can be achieved. The cost function is,

$$J = \frac{1}{2}\int_{t_0}^{t_f} \mathbf{U}^T\mathbf{U}dt, \tag{3.6}$$

As the system need to reach the goal position, the deviation between the state at final time and the goal position has to be minimized. By enforcing the cost on the control input, the objective to minimum energy trajectory can be achieve. The final performance objective after adding the terminal objective function is,

$$J = \Phi\left(\mathbf{X}(t_f), t_f, \mathbf{P}\right) + \int_{t_0}^{t_f} \frac{1}{2}\mathbf{U}^T\mathbf{U}\ dt\ +\ \boldsymbol{\lambda}^T\mathbf{f}\left(\mathbf{X}(t), \mathbf{U}(t), t, \mathbf{P}\right)\ dt, \tag{3.7}$$

The Hamiltonian H for this case is defined as,

$$H = \frac{1}{2}\mathbf{U}^T\mathbf{U}\ +\ \boldsymbol{\lambda}^T\mathbf{f}\left(\mathbf{X}(t), \mathbf{U}(t), t; \mathbf{P}\right),$$

By following same approach used in Sec. 3.1, the boundary condition and differential equation can be found. In this case, as control values are directly involved in the objective function. We can replace the control vector with Lagrange multipliers in differential equations by solving for necessary conditions. After solving for the necessary condition, control values can eliminate by Lagrange multipliers using Eq. (3.8).

$$\begin{aligned} \upsilon = & -(\lambda_x \cos\psi + \lambda_y \sin\psi) \\ \omega = & -\lambda_\psi \end{aligned} \tag{3.8}$$

By using differential equations Eq. (3.3) and boundary conditions (initial and final value of states) Eq. (3.4), the solution states and Lagrange multipliers can be obtained by posing the problem as TPBVP.

3.3 Simulation Results

In this section, simulation results for a reference trajectory (with minimum control effort objective function) generated using analytical approach derived in Sec. 3.2 and reference trajectory generated using numerical approach described in ch. 4 are shown. To analyse the performance of the algorithms, trajectory generation with various boundary conditions are described in following section.

Note that by providing a good initial guess of the state vector and Lagrange multipliers convergence rate for finding the solution can be increased.

For the first test case, simple trajectory from an initial state ($x_0 = 0,\ y_0 = 0,\ \psi_0 = 90^\circ$) to a final state ($x_f = 10,\ y_f = 0,\ \psi_f = 0^\circ$) using the analytical and numerical approach is generated. A safe zone of a meter diameter is considered around the waypoint.

The reference trajectory from initial position to the goal position using ana-

lytical and numerical approach are depicted in Figs. 3.1 and 3.2 respectively. Both trajectories have smooth maneuver from the initial position to the goal position.

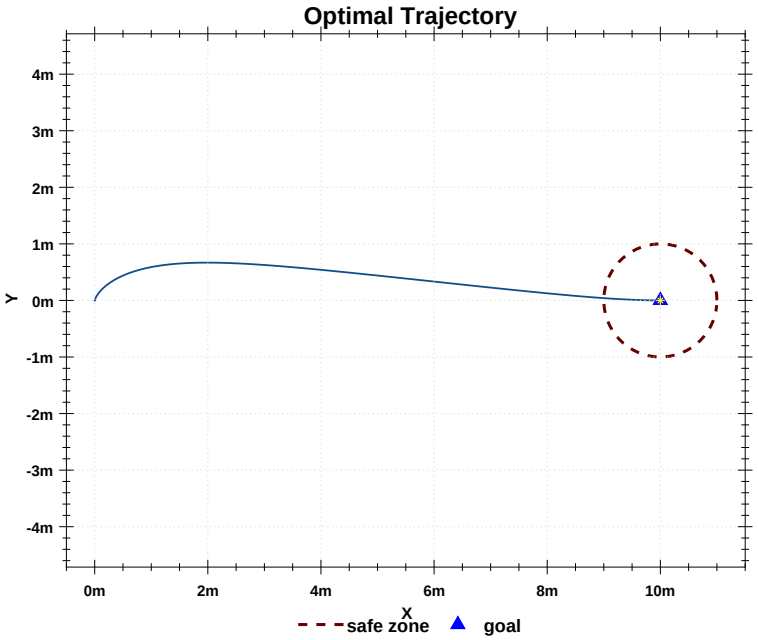

Figure 3.1: Simulated optimal trajectory using analytical approach

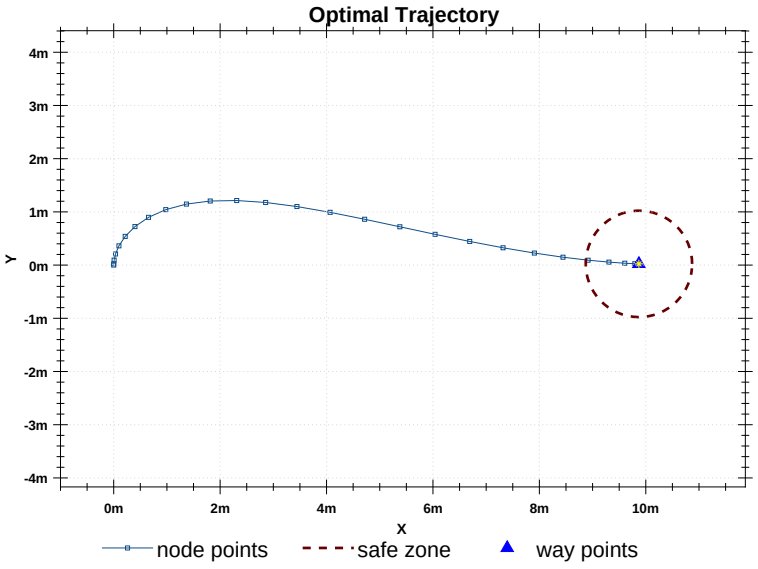

Figure 3.2: Simulated optimal trajectory using numerical approach

Node points shown in Fig. 3.2 refers to the point at which the optimizer tries to satisfies the objective functions and constraints in order to generate the optimal trajectory. Further details about the trajectory generated using the numerical approach are described in ch. 4.

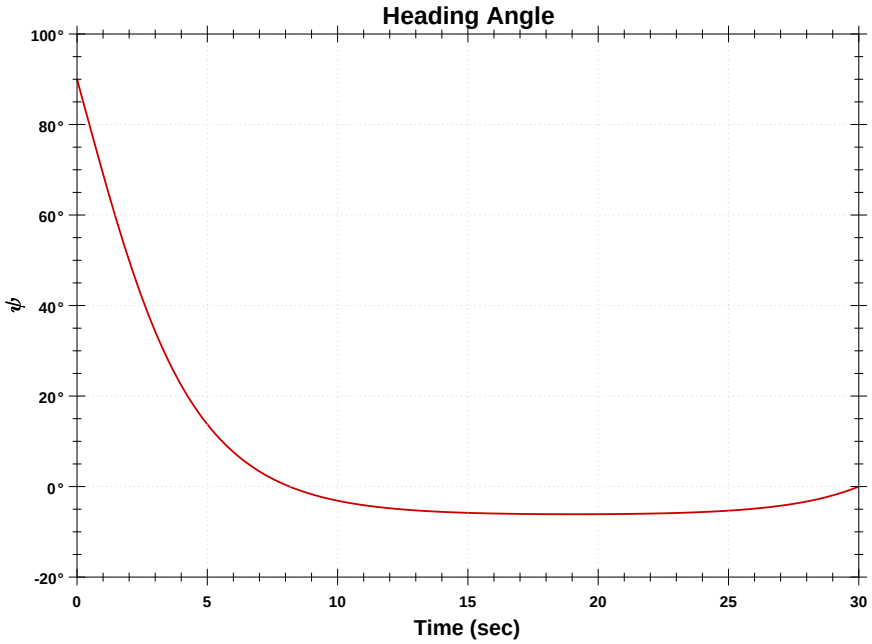

Figure 3.3: Heading angle using analytical approach

The change in heading angle during the entire trajectory is shown in Figs. 3.3 and 3.4 for respective analytical and numerical approaches. In the case of the analytical approach, heading angle of WMR quickly changed to -5° from 90° in the first 7 seconds and stays constant at -5° uptill last 3 seconds and changed to 0° at the end to satisfy the boundary condition. In contrast,with the numerical approach, the heading angle of WMR is gradually changing during the entire trajectory which help to reduce jerk on the system.

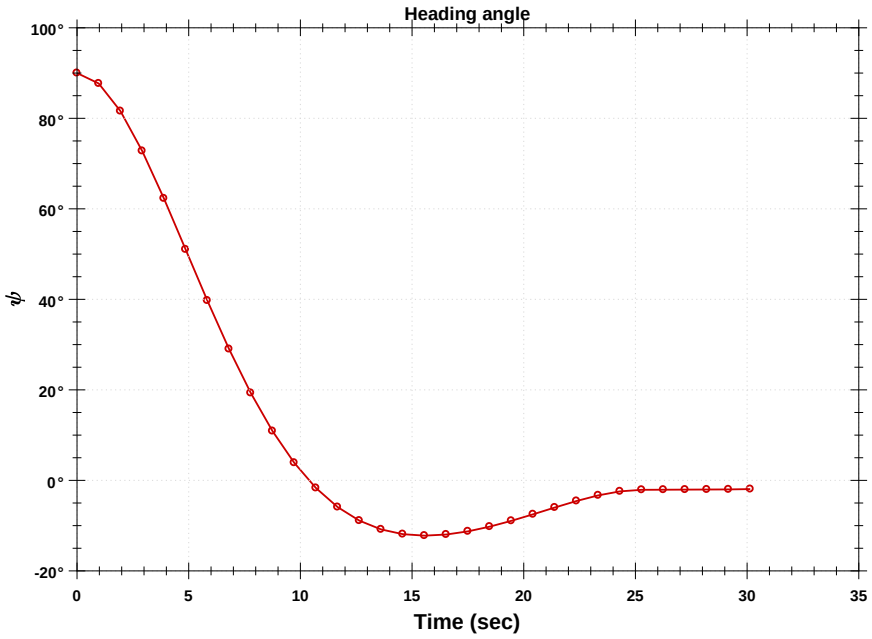

Figure 3.4: Heading angle using numerical approach

The required control input to achieve the desired trajectory is shown in Fig. 3.5 for the analytical approach and in Fig. 3.6 for the numerical approach. With the analytical approach, the control vectors are eliminated by using Lagrange multipliers from the systems differential equation in order to simplify the problem. Once the optimal trajectory is created, the time history for the control is obtained by back-substituting using Eq. (3.8). As control vectors are not part of trajectory generation, still we can reduce the change in the control effort by reducing change in states but we cannot put constraints (i.e terminal constraint) on it. As a result, linear and angular velocity of WMR starts and ends at any values which is difficult to incorporate while implementing the trajectory on real-hardware. In contrast, with the numerical approach the controls bounds along with the initial and final control values can be specified while solving for optimal control. As control inputs have constraints at initial and final time, the control history starts and ends at the specified values.

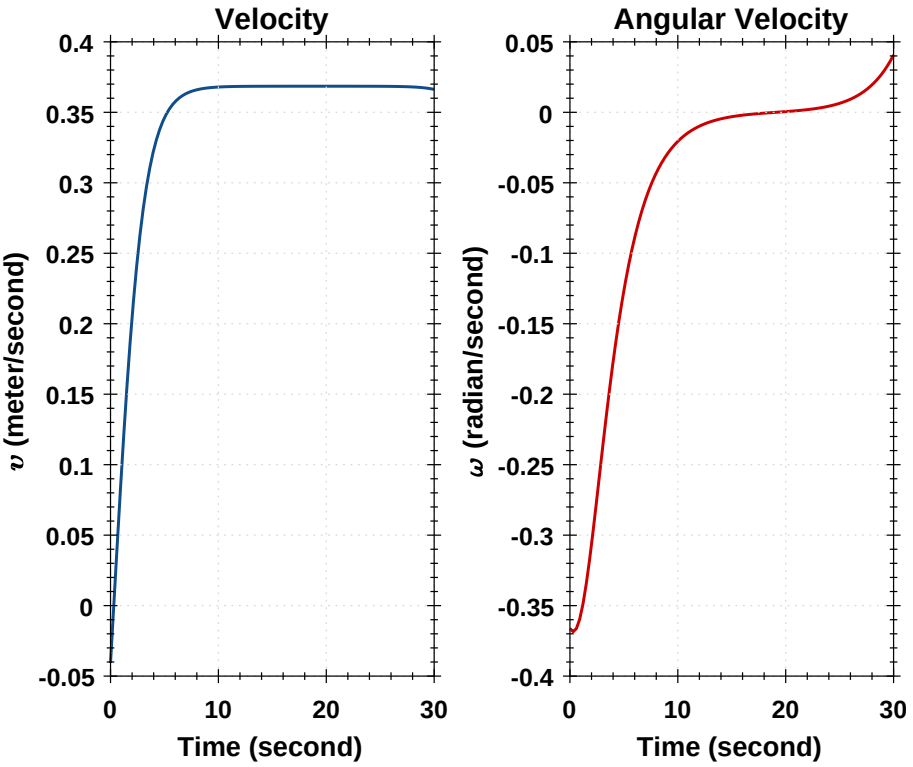

Figure 3.5: Control inputs using analytical approach

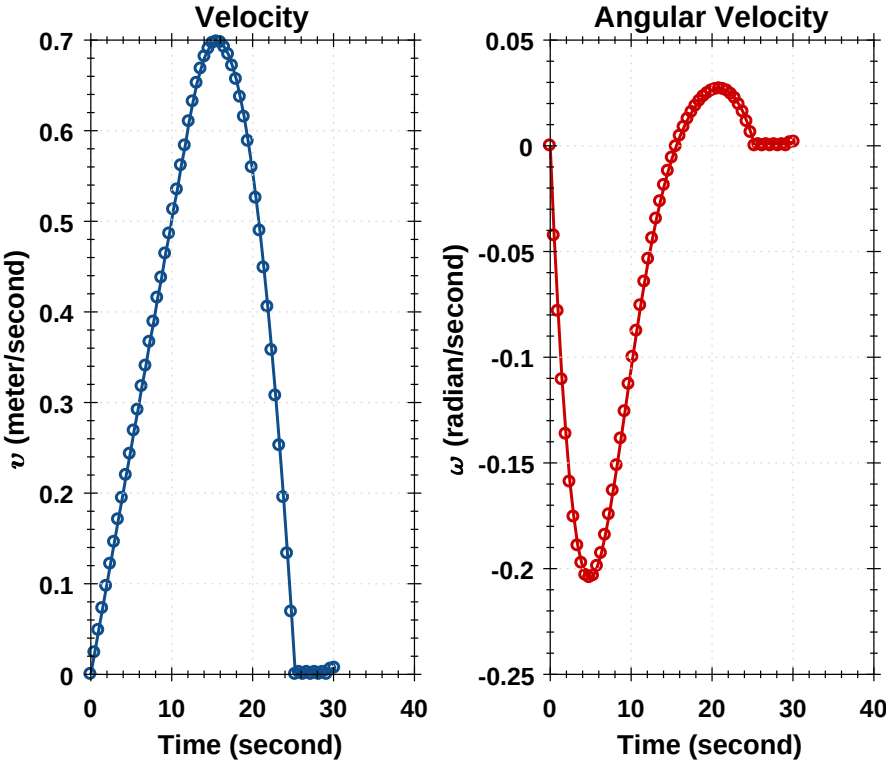

Figure 3.6: Control inputs using numerical approach

To evaluate the capability of the analytical method, another trajectory with initial state ($x_0 = 0,\ y_0 = 0,\ \psi_0 = 0°$) to final state ($x_f = 10,\ y_f = 10,\ \psi_f = 45°$) is generated.

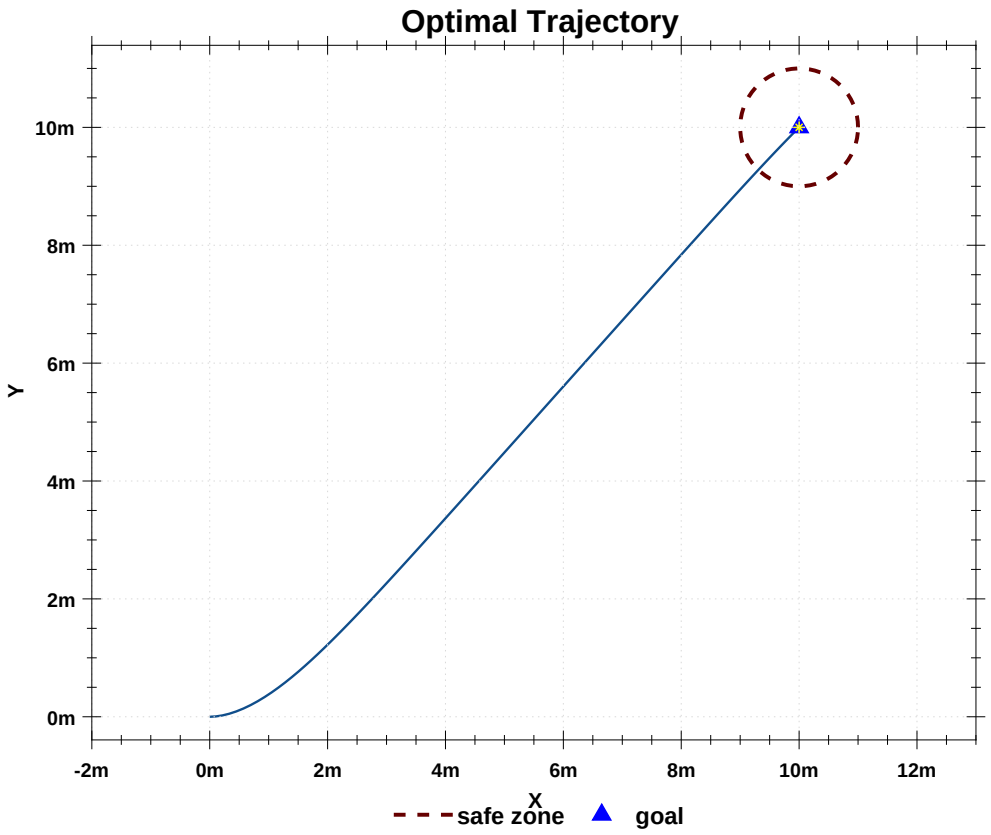

Figure 3.7: Simulated optimal trajectory using analytical approach

The reference trajectory for the second test case is depicted in Figs. 3.7 and 3.8. The results reveals that both the approaches are capable of generating feasible trajectories for various boundary conditions.

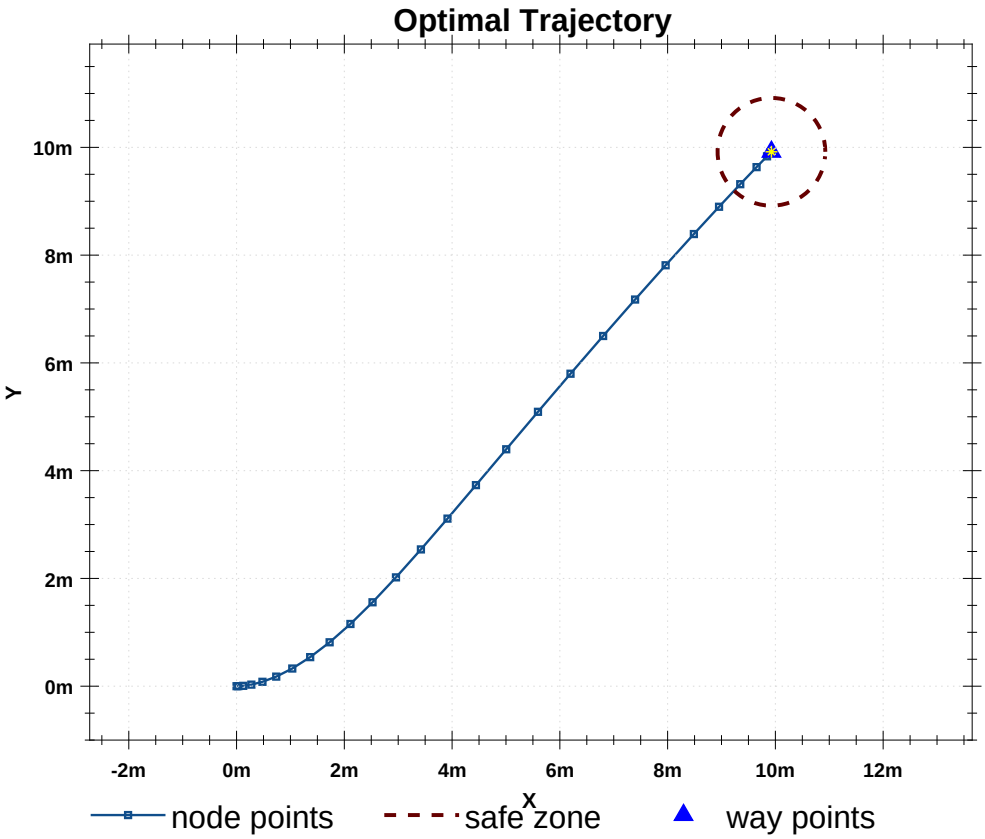

Figure 3.8: Simulated optimal trajectory using numerical approach

As seen in the first test case, the heading angle generated using analytical approach has turns quickly as shown Fig. 3.7. While with the numerical approach, as the trajectory changes heading angle and time history to achieve the trajectory are also varies in order to maintain smooth steering of WMR as shown in Fig. 3.8.

Similarly, the control input history to achieve desired trajectory is shown in Fig. 3.11 and 3.12 for the respective analytical and numerical approaches.

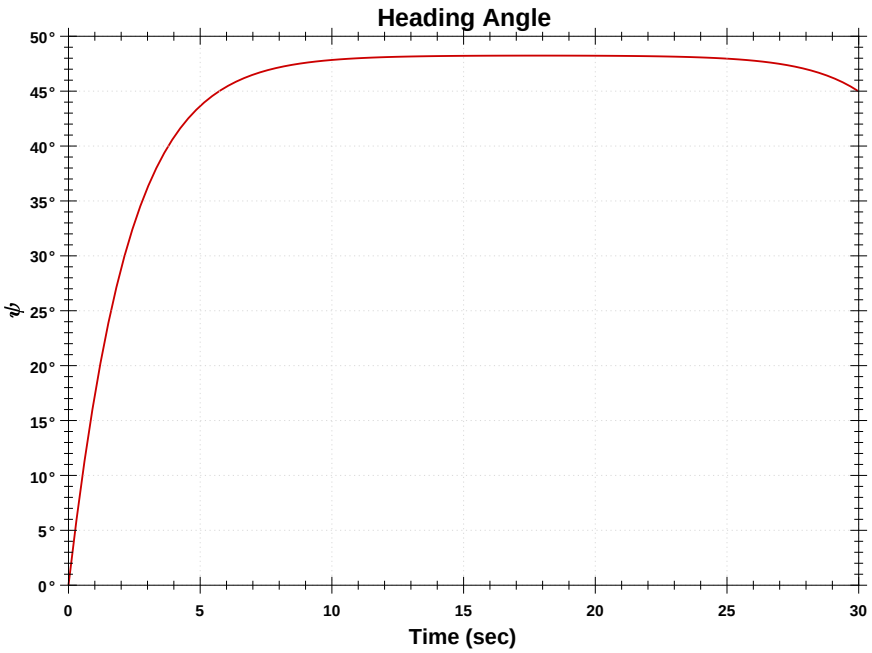

Figure 3.9: Heading angle

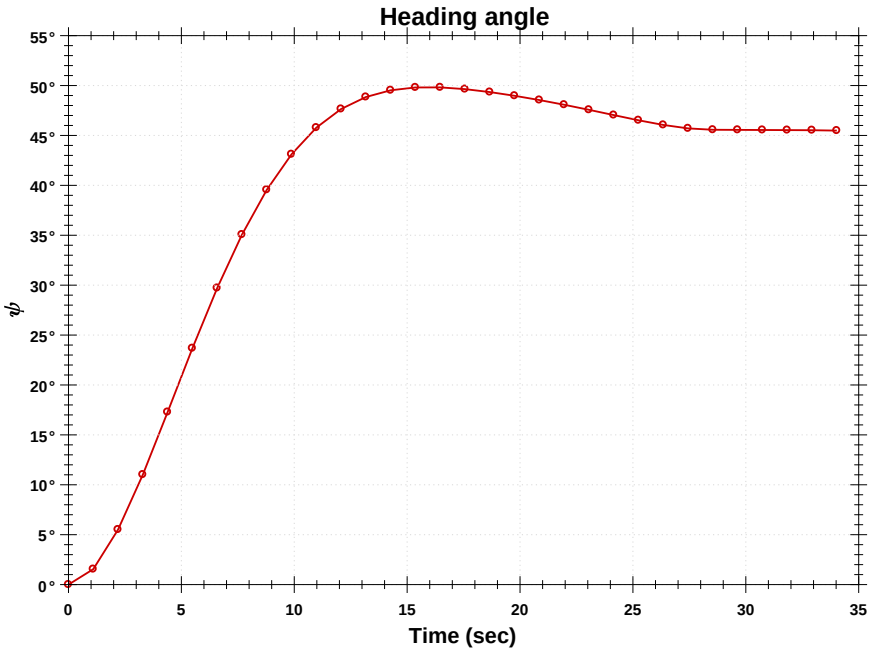

Figure 3.10: Simulated optimal trajectory using analytical method

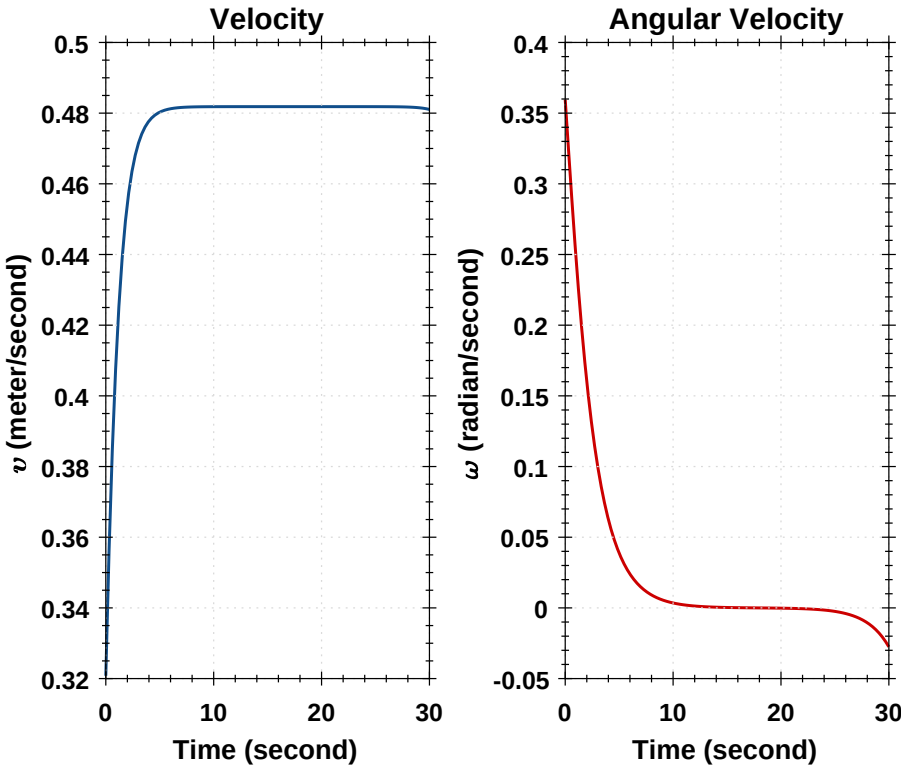

Figure 3.11: Control inputs using analytical approach

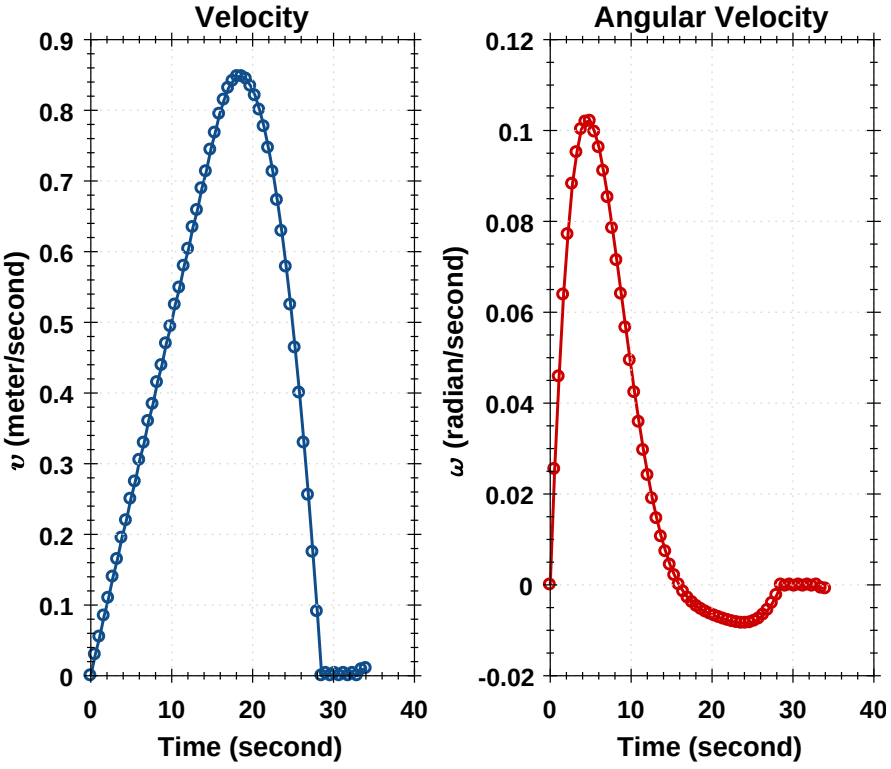

Figure 3.12: Control inputs using numerical approach

3.4 Summary

This chapter shows the capability of the analytical method to achieve a feasible trajectory given a particular objective function. For the case optimal trajectory with bounds on the applied control input, numerical approach is best option. Moreover, it is difficult to find a good initial condition for different trajectories using the analytical approach and when using inequality constraints it becomes difficult to solve the problem using the analytical approach. All this problem can be resolved by using the numerical approach which is described in the next chapter.

CHAPTER 4

Collocation Methods based Trajectory Optimization

This chapter presents an different collocation techniques and how NLP plays a key role to solve the trajectory optimization problem. This following sections will presents how to pose the NLP problem using different collocation methods.

Initially, general formulation of the NLP is presented. Then the classification of the different collocation methods and improved integration technique to increases the computation time and convergence rate to solve the trajectory optimization problem is described. At the end, the offline trajectory optimization framework to design the reference trajectory for WMR is proposed.

4.1 Nonlinear optimization

The main challenge in solving for OCP is to solve the NLP. NLP is the process of solving an optimization problem defined by a system of equalities and inequalities, collectively termed as constraints, over a set of unknown real variables, along with an objective function to be maximized or minimized, where some of the constraints or the objective function are nonlinear [33] . General formulation of the NLP is following: the decision vector $\mathbf{X}(t) \in \mathbb{R}^n$ that minimizes the cost function $J(\mathbf{X})$ subjected to the algebraic constraints,

$$\begin{aligned} \mathbf{g}(\mathbf{X}) &= 0 \\ \mathbf{h}(\mathbf{X}) &\leq 0 \end{aligned} \tag{4.1}$$

where $\mathbf{g}(\mathbf{X}) \in \mathbb{R}^m$ and $\mathbf{h}(\mathbf{X}) \in \mathbb{R}^p$ and the cost function is,

$$J(\mathbf{X}_{i+1}) \leq J(\mathbf{X}_i) + K\ \alpha_i\ \nabla Js^T(\mathbf{X}_i)\ \mathbf{p}_i \tag{4.2}$$

where K is a parameter between 0 and 1, i is index, $\mathbf{p}_i$ is minimization or maximization search direction and α_i is the step size. From search direction and step size, $\mathbf{X}_{i+1}$ can be determined.

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \alpha_i \mathbf{p}_i \tag{4.3}$$

The most popular solution methods for NLP problem are the interior-point(IP) and the sequential quadratic programming (SQP).

4.2 Collocation method

The collocation method uses a polynomial approximation technique. It transcribes the dynamic optimization problem into the nonlinear parameter optimization problem. This conversion process is also known as transcription, and hence a collocation method is also called a transcription method [9, 12, 21]. In collocation, the entire trajectory is divided into the nodes called interval or nodal points, where the state and control are approximated using piecewise continuous functions. The control is approximated using a piecewise continuous linear function and the states use piecewise continuous cubic polynomials.

Collocation points are the point at which system kinematic equations are enforced through interpolation or quadrature rule. Depending on the integration method, the collocation points are either set at the mid points of the nodal points (Direct Collocation method) or they are exactly at nodal points (PS method). Fig. 4.1 shows the nodal point and collocation points where state and control values are approximated.

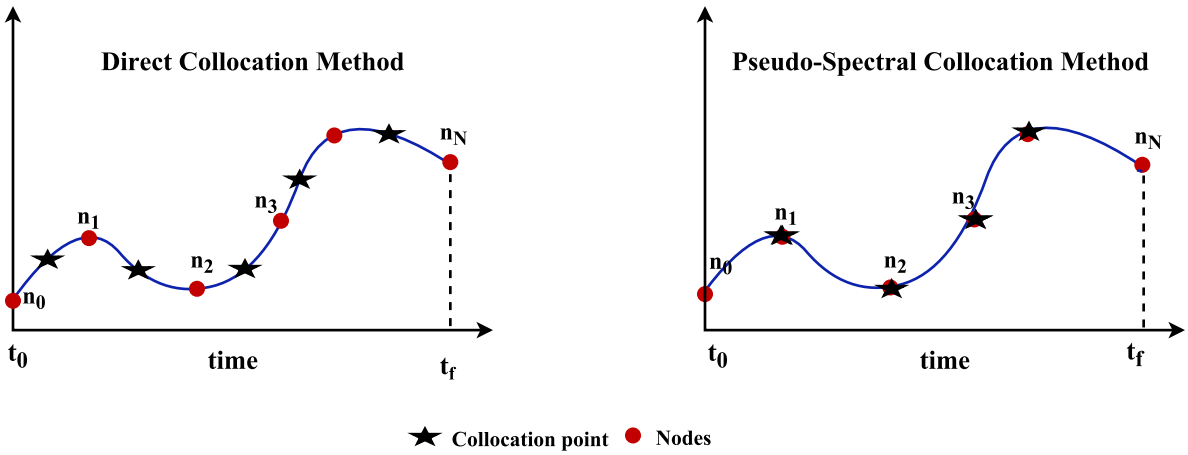

Figure 4.1: Collocation methods

Collocation methods are summarised into two group of techniques: Direct collocation and PS collocation. In next section we will discuss general formulation of the direct collocation method and how it transcribe the dynamic optimization problem into NLP.

4.3 Direct collocation method

Direct collocation is one of the simplest and effective collocation technique. It discretizes the entire trajectory into equally spaced nodes in time. Now, the entire problem becomes a coefficient search problem for piecewise continuous functions which satisfy the boundary conditions between each successive node.

Let us descritize the entire problem into N nodes,

$$0 = t_i < t_{i+1} < t_{i+2} < \ldots < t_N = t_f$$

where $t_i, t_{i+1}, \ldots, t_f$ are the times at nodes $i, i+1, \ldots, N$ respectively and t_f is the final time. The state vector and the control vector are also discretized into N parameters at

N grid points along with a final time t_N, which will be the parameters for a non-linear program,

$$\mathbf{NLP} = (\mathbf{X}(t_i), \mathbf{X}(t_{i+1}), \ldots, \mathbf{X}(t_N), \mathbf{U}(t_i), \mathbf{U}(t_{i+1}), \ldots, \mathbf{U}(t_N), t_N) \tag{4.4}$$

For each interval from i to $i+1$, the control value is chosen as the piecewise linear interpolating function,

$$\mathbf{U}(t) = \mathbf{U}(t_i) + \frac{t - t_i}{t_{i+1} - t_i}(\mathbf{U}(t_{i+1}) - \mathbf{U}(t_i)) \tag{4.5}$$

The state alue is chosen as the piecewise continuous cubic polynomial between $\mathbf{X}_i$ and $\mathbf{X}_{i+1}$,

$$\mathbf{X}(t) = c_o + c_1 t + c_2 t^2 + c_3 t^3 \tag{4.6}$$

To calculate the polynomial coefficients between i and $i+1$,

$$\begin{aligned} \mathbf{X}(t_i) &= c_o + c_1 t_i + c_2 t_i^2 + c_3 t_i^3 \\ \dot{\mathbf{X}}(t_i) &= c_1 + 2c_2 t_i + 3c_3 t_i^2 \\ \mathbf{X}(t_{i+1}) &= c_o + c_1 t_{i+1} + c_2 t_{i+1}^2 + c_3 t_{i+1}^3 \\ \dot{\mathbf{X}}(t_{i+1}) &= c_1 + 2c_2 t_{i+1} + 3c_3 t_{i+1}^2 \end{aligned} \tag{4.7}$$

where, $\mathbf{X}(t_i)$ and $\mathbf{X}(t_{i+1})$ are obtained from the node points and $\dot{\mathbf{X}}(t_i),\ \dot{\mathbf{X}}(t_{i+1})$ are obtained from the differential equation given in Eq. (2.9) by substituting $\mathbf{X}(t_i),\ \mathbf{X}(t_{i+1})$ respectively. Then the coefficients can be obtained by solving

$$\begin{bmatrix} c_o \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 1 & t_i & t_i^2 & t_i^3 \\ 0 & 1 & 2t_i & 3t_i^2 \\ 1 & t_{i+1} & t_{i+1}^2 & t_{i+1}^3 \\ 0 & 1 & 2t_{i+1} & 3t_{i+1}^2 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{X}(t_i) \\ \dot{\mathbf{X}}(t_i) \\ \mathbf{X}(t_{i+1}) \\ \dot{\mathbf{X}}(t_{i+1}) \end{bmatrix} \tag{4.8}$$

with the boundary conditions $\mathbf{X}_i$ at t_i and $\mathbf{X}_{i+1}$ at t_{i+1}.

4.3.1 Residual error

At collocation points, the states approximated using the differential equations and the change in the states approximated using the cubic polynomial have to be equal. The process to approximate the state at collocation point is known as the residual error or state error approximation.

The approximating functions of the states have to satisfy the differential equations at the grid point t_i, where $i = 1, \ldots, N$. The assumed approximation of $\mathbf{X}(t_i)$ must satisfy the differential equations at the grid point t_i. To fulfill residual error condition at each iteration, it can be placed as a constraint (equality constraint) in nonlinear programming problem setup. The residual error at the collocation point should be driven to zero.

$$\mathbf{f}(\mathbf{X}, \mathbf{U}, t_c) - \dot{\mathbf{X}}_c = 0 \tag{4.9}$$

where, t_c is mid point of t_i and t_{i+1},

$$t_c = \frac{(t_{i+1} - t_i)}{2}, \tag{4.10}$$

state at collocation point,

$$\mathbf{X}_c = \frac{\mathbf{X}_{i+1} + \mathbf{X}_i}{2}, \tag{4.11}$$

and the polynomial approximated at collocation point

$$\dot{\mathbf{X}}_c = c_1 + 2c_2 t_c + 3c_3 t_c^2, \tag{4.12}$$

Based on the interpolation of polynomials and modification in the integration rules, direction collocation technique can also called PS method.

4.4 Constraints

In trajectory optimization problem there are two types of constraints to deal with, input constraints and state constraints. In many control system problems, input variables have such limitation that they cannot be arbitrarily large, such as magnitude limits. Also because of the system kinematics and for the safety reasons states must be bounded.

4.4.1 Input constraints

Input constraint follows a general form,

$$\mathbf{U}_{lb} \leq \mathbf{U}_i \leq \mathbf{U}_{ub}, \quad i = 1, ..., N-1 \tag{4.13}$$

where $\mathbf{U}_{lb}$ is lower bound and $\mathbf{U}_{ub}$ is upper bound. As per the motor calibration test, the lower and higher control bounds are depicted in Fig. 5.3. The trajectories are designed to go forward and WMR have capability to turn, so anti clock rotation of motor are also bounded using input constraints.

4.4.2 Environmental constraints

Environmental constraints are the inequality constraints on state vector or control vector during the entire trajectory. The generalize form to pose the environmental constraint is,

$$\mathbf{g}(\mathbf{X}(t_i), \mathbf{U}(t_i), t_i) \geq 0, \quad i = 1, \ldots, N \tag{4.14}$$

where, $\mathbf{g}$ is function of state and control at nodal points. By providing proximity distance (r_p), which is a minimum distance vehicle has to maintain from the obstacle position. The obstacle constraint can be set as,

$$r_p^2 - ((x_V(i) - x_O(k))^2 + (y_V(i) - y_O(k))^2) \leq 0 \tag{4.15}$$

where, (x_V, y_V) and (x_O, y_O) are denoted as vehicle and obstacle position in cartesian coordinates respectively , $i = 1, \ldots, N$ is number of nodes and $k = 1, \ldots, M$ is number of obstacles.

4.4.3 Endpoint constraints

For stable maneuver, system has to be in equilibrium at the last nodal point. The state vector and control vector of the vehicle can be set at specific values which makes the system stable at the end. However, to achieve a smooth stop at the end, the system has to be decelerate near the goal position. Hence, by implementing an endpoint constraint, we can achieve a smooth stop at the goal position.

$$\upsilon((t_f - 5) \rightarrow t_f) = 0 \tag{4.16}$$

In case of multiple waypoint trajectory, vehicle has to maintain same state and control values during the waypoint switching to achieve continuous trajectory. By adding the modified endpoint condition for the waypoint switching, continues turning trajectory can be achieve. The endpoint condition for waypoint switching is,

$$\upsilon((t_f - 5) \rightarrow t_f) - 0.5 = 0 \tag{4.17}$$

where, υ is

$$\upsilon = \sqrt{\dot{x}_V^2 + \dot{y}_V^2} \tag{4.18}$$

the speed of vehicle.

4.5 Performance index

4.5.1 Goal objective function

Achieving the goal position is a primary objective for most of the trajectory optimization problem. Performance objective can be satisfied if we enforce magnitude

of last nodal point of vehicle to be a goal position in objective function. The objective function to achieve goal becomes,

$$J_{goal} = \beta_{goal}((x_V(N) - x_G)^2 + (y_V(N) - y_G)^2) \tag{4.19}$$

where, (x_G, y_G) is goal position in Cartesian coordinates and β_{goal} is the weighting factor to achieve the goal position.

4.5.2 Final time minimization

Minimum time optimal trajectory can be achieved by reaching the goal position as quickly as possible. This objective can be satisfy by enforcing final time as an additional objective function to the overall performance objective. The objective function for minimum time trajectory,

$$J_{t_f} = \beta_{time} t_f \tag{4.20}$$

where, β_{time} is the weighting factor to complete the trajectory in a minimum time.

4.5.3 Heading angle constraint

To achieve a smooth desired trajectory, deviation in the heading angle has to be minimum. As shown in Eq. (2.9) change in the heading angle of vehicle is based on the vehicle angular velocity. A sudden change in the heading angle will cause a jerk in the system. Performance objective can be satisfied by enforcing summation of change in the heading angle as an objective function. The cost which reduces the change in the heading angle during entire trajectory,

$$J_{\psi} = \beta_{\psi} \sum_{i=1}^{N-1} (\psi_{i+1} - \psi_i)^2 \tag{4.21}$$

where β_{ψ} is the weighting factor to reduce a change in the heading angle.

4.5.4 Actuator rate constraints

The WMR motor often functions similar to a first order system and sudden changes can cause damage to WMR or onboard equipment. Hence, constraints on a rate of change of the actuator inputs are necessary. The change in angular velocity of wheels is constrained at nodal points by doing a summation of change in the angular velocity. The cost which reduces change in the angular velocity of wheel during the entire trajectory,

$$J_{control} = \beta_{control} \sum_{i=1}^{N-1} \left(\mathbf{U}(i+1) - \mathbf{U}(i)\right)^2 \tag{4.22}$$

where, $\beta_{control}$ is the weighting factor to reduce the change in the angular velocity. Using above constraints and performance indices, main performance objective of trajectory optimization problem is,

$$\begin{aligned}
\min_{(\mathbf{U},\ t_f)} \quad &: \quad J(\mathbf{X},\ \mathbf{U},\ t_f) \\
Subject\ to \quad &: \quad r_p^2 - ((x_V(i) - x_O(k))^2 + (y_V(i) - y_O(k))^2) \leq 0 \\
&\quad\ \ \upsilon((t_f - 5) \rightarrow t_f) = 0 \\
Performance\ Index \quad &: \quad J = J_{t_f} + J_G + J_\psi + J_{control}
\end{aligned}$$

4.5.5 System constraints

By incorporating actuator rate and heading angle constraints, we can reduce the sudden change in states and controls. Still to achieve desire trajectory, we have to tune the weights on objective functions. In order to reduce that hassle, we can indirectly enforce the constraint on control through the linear and angular acceleration of the system. The linear and angular acceleration constraint can be set as,

$$a_V - a_{max} \leq 0 \tag{4.23}$$

$$\alpha_V - \alpha_{max} \leq 0 \tag{4.24}$$

Relationship between states, linear acceleration and angular acceleration is shown in Eq. (4.25) can derived by differentiating the vehicle kinematics Eq. (2.9), which shows that the constraint on linear and angular acceleration are directly effective to reduce the sudden change in vehicle position and heading angle.

$$\begin{aligned} \ddot{x} &= -\upsilon\ sin(\psi)\ \omega\ +\ \dot{\upsilon}\ \cos(\psi) \\ \ddot{y} &= \upsilon\ cos(\psi)\ \omega\ +\ \dot{\upsilon}\ \sin(\psi) \\ \ddot{\psi} &= \dot{\omega} \end{aligned} \tag{4.25}$$

where, a_V is linear acceleration of the vehicle

$$\dot{\upsilon}\ =\ a_V\ =\ \frac{\upsilon_{i+1} - \upsilon_i}{t_{i+1} - t_i}, \tag{4.26}$$

and α_V is angular acceleration of the vehicle

$$\dot{\omega}\ =\ \alpha_V\ =\ \frac{\omega_{i+1} - \omega_i}{t_{i+1} - t_i} \tag{4.27}$$

4.6 Trajectory Optimization problem setup

By combining the different components (system kinematics, NLP, constrints, objective function) to solve the trajectory optimization problem, the generalize offline trajectory optimization framework is shown in Fig. 4.2. As shown in the framework, providing system kinematics, constraints, and performance index with some initial guess to the NLP, we can solve for optimal trajectory and optimal control input. Moreover, continuous multiple waypoint trajectories can be achieved by providing current and next waypoint position and current control input to the interior point solver.

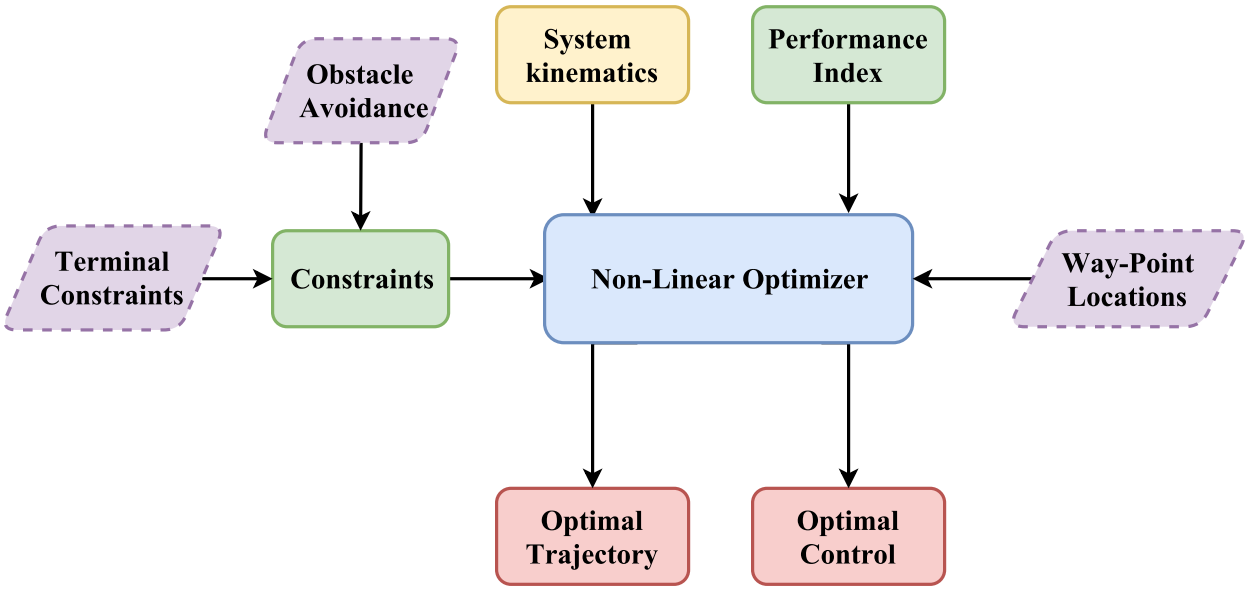

Figure 4.2: Offline trajectory optimization framework

As optimizer is solving trajectory optimization problem using collocation method with different constraints and objective functions, the generalize form to calculate the total number of problems solved at each iteration are as follows :

- As the state vector state vector $\mathbf{X} = [x\ y\ \psi]^T$ can be calculated using the control vector $\mathbf{U} = [\omega_r\ \omega_l]^T$ from system kinematic Eq. (2.9), so the number of states are 3 denoted by 'States', the number of control are 2 denoted by 'Control'
- The trajectory is divided into n number of nodes denoted as,

$$Node = n,$$

so that the number collocation points becomes,

$$c = \frac{n+1}{2}$$

- Control vector is approximated at the nodes as well as at the collocation point, so the number of control points becomes,

$$m = 2n + 1$$

- Let the number of obstacle be k
- C_1 be the constraint for the states and controls at initial and at the goal position given as,

$$C_1 = State_{initial} + Control_{initial} + State_{goal} + Control_{goal}$$

- State vector approximated at collocation points is called as residual error, which is posed as equality constraint. So the number of state errors approximated at single iteration becomes,

$$C_2 = c\ State$$

- Based on system configuration, bounds on the control vector is enforced as an input constraint at all control points. The number of input constraints becomes,

$$C_3 = m\ Control$$

- As per user defined number of obstacles, obstacle constraints are incorporated at each nodal point in order to achieve smooth collision free trajectory. The number of obstacle constraint becomes,

$$C_4 = n\ k$$

- To achieve smooth stop at goal position, endpoint constraint is enforced for the last 'i' number of nodes. The number of endpoint constraint for the velocity of vehicle becomes,

$$C_5 = i\ v$$

- To achieve smooth trajectory, constraints on linear and angular acceleration are posed at each nodal points. The number of system constraints become,

$$C_6 = n\ a_V + n\ \alpha_V$$

- Based on performance objective, the cost function to be satisfied is given by,

$$J = J_{t_f} + J_G + J_\psi + J_{control}$$

Total number of problems solved at each iteration for particular test case is shown in table. Based on number of nodes and number of obstacles, the problem solved at each iteration varies.

Case No.	Test Case	C_1	C_2	C_3	C_4	C_5	C_6	Performance Objective	Total
1	n = 31 k = 0	3+2+3+2 = 10	16 × 3 = 48	63 × 2 = 126	31 × 0 = 0	5 × 1 = 5	31 × 2 = 62	1+1+1+1 = 4	255
2	n = 31 k = 2	3+2+3+2 = 10	16 × 3 = 48	63 × 2 = 126	31 × 2 = 62	5 × 1 = 5	31 × 2 = 62	1+1+1+1 = 4	317
3	n = 33 k = 3	3+2+3+2 = 10	17 × 3 = 51	67 × 2 = 134	31 × 3 = 93	5×1= 5	33 × 2 = 66	1+1+1+1 = 4	363

Table 4.1: Trajectory Optimization Problem count

4.7 Offline Simulation Results

Results presented in this chapter uses the offline trajectory optimization framework proposed in Fig. 4.2 to generate the reference trajectory to arrive at a desired goal position while avoiding obstacles. To analyze the performance of the optimization algorithm, various trajectories with different sets of waypoints and obstacle position are described in the following section.

Note that providing a good initial guess of the state vector, control vector, and final time to reach at goal will increase the convergence rate of the optimizer to achieve

desired results. The weight on the a particular objective function(like Minimum time, actuator rate, etc) can be tuned to emphasize a specific objective function.[1]

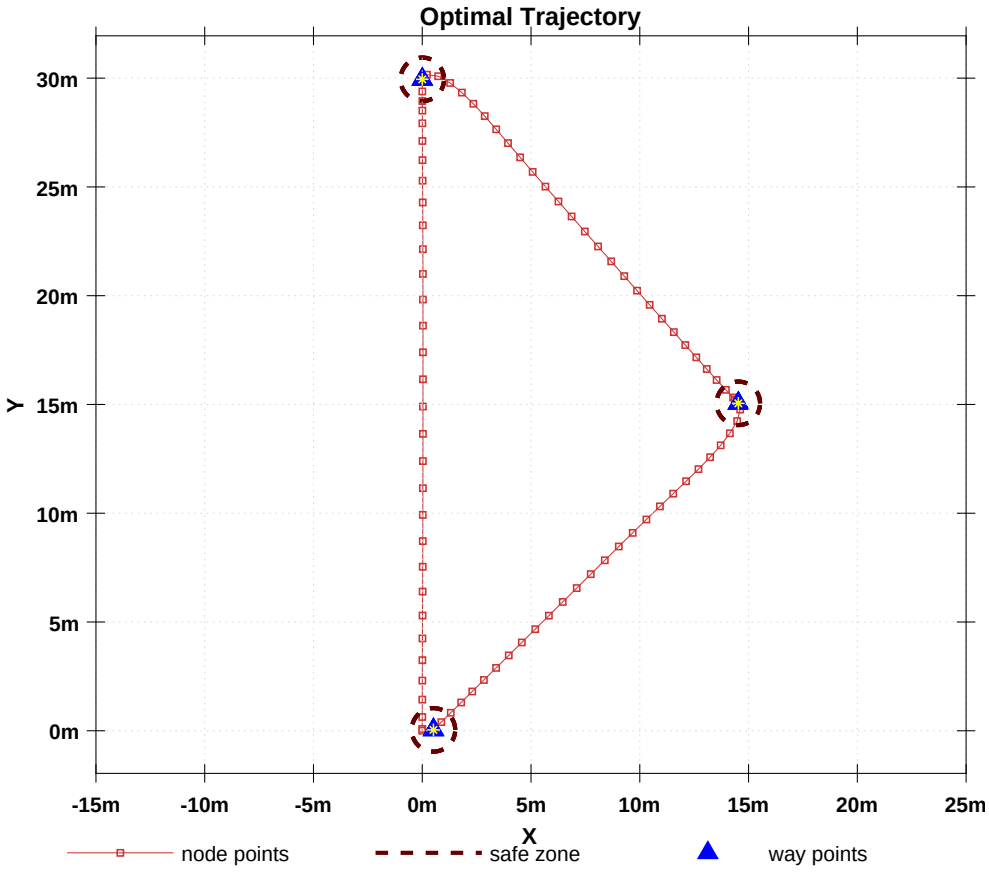

Figure 4.3: Simulated optimal trajectory for isosceles triangle

For the first test case three waypoints which form an isosceles triangle of respective side length 30-15-15 meters and for the second test case 4 waypoints which form a parallelogram of a respective length of 15 meters provided to an optimizer. The safe zone of a 1-m diameter is considered around the waypoint.

The reference trajectory which traverse through all waypoints obtained from op-

[1]The effect of weight on a particular cost function can be analyzed by performing optimization with different weights. Then the respective weight can be selected based on desired performance.

timizer are depicted in Figs. 4.3 and 4.4, where minimum control effort and minimum change in heading angle are shown in Figs. 4.5, 4.6, and 4.7 respectively.

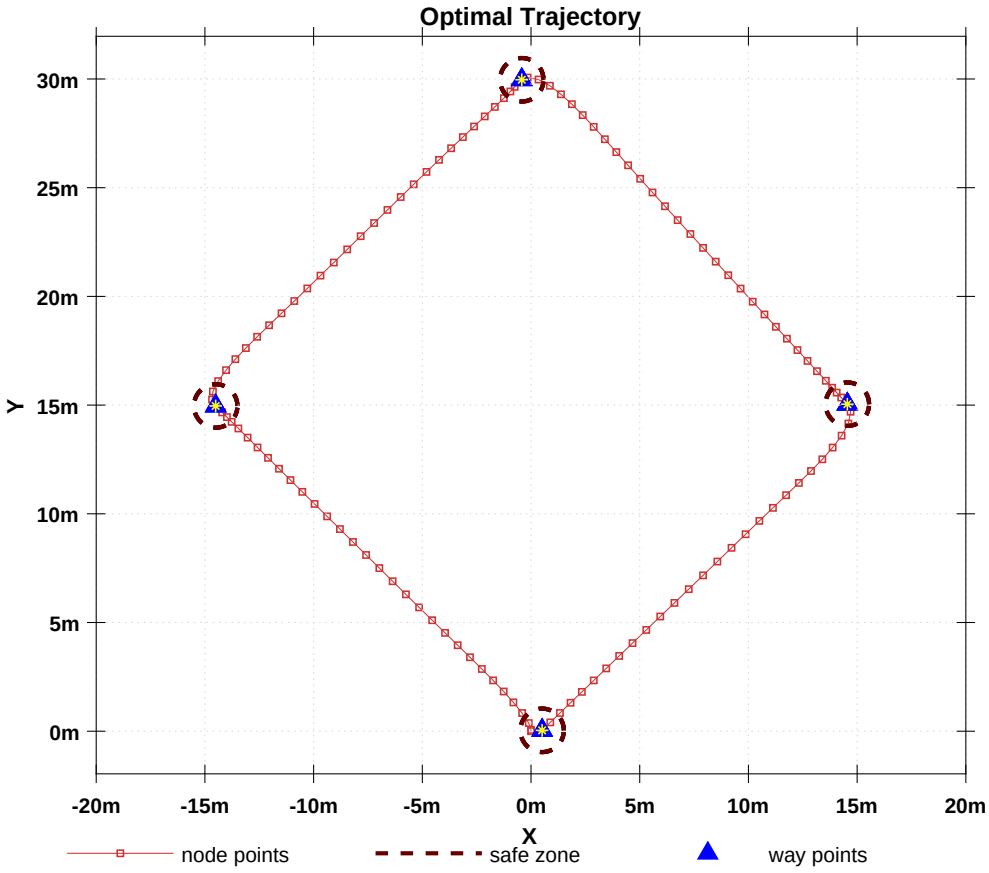

Figure 4.4: Simulated optimal trajectory test for parallelogram

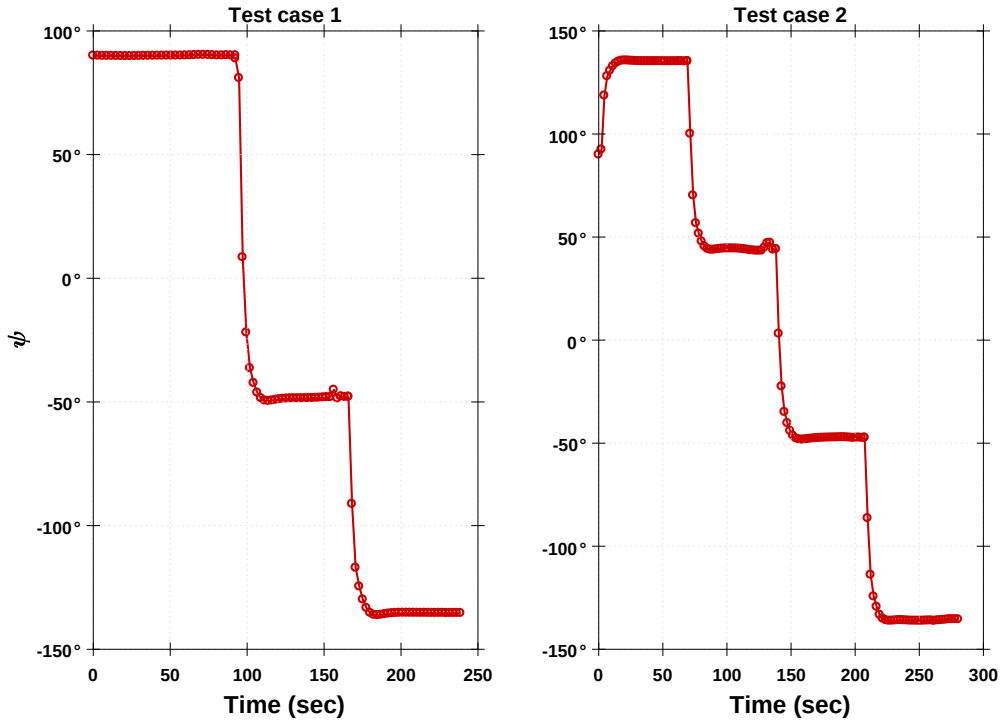

Figure 4.5: Heading angle of WMR

As shown in Fig. 4.5, heading angle of WMR changes smoothly varied during the entire maneuver. As we assumed that the vehicle has differential drive steering which can change heading angle at its current position, there is a sudden change in heading angle at waypoint transition point. Even though there is a sudden change in heading angle at waypoint transition, there is little fluctuation in control as WMR has to maintain a nominal speed of 0.1 m/sec at waypoint transition. We can see that in Figs. 4.6 and 4.7 for respective cases.

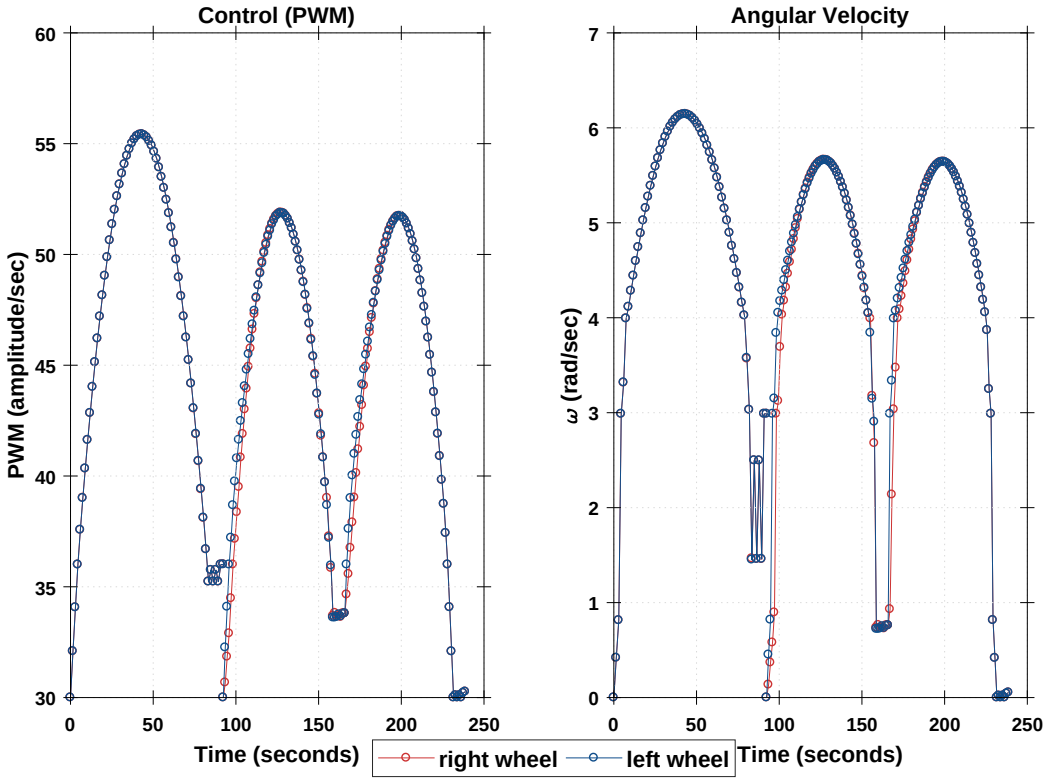

Figure 4.6: Simulated optimal control test for isosceles triangle

Figs. 4.3 and 4.4 illustrates the efficiency of the optimal trajectory generator to design desired reference trajectories under terminal constraints.

Moreover, Figs. 4.8 and 4.9 reveals that the optimizer is also proficient in generating the desired reference trajectory under the presence of an additional obstacle constraint. For both cases, obstacles have a 1.5 m diameter along with the additional proximity of the 0.5 m diameter.

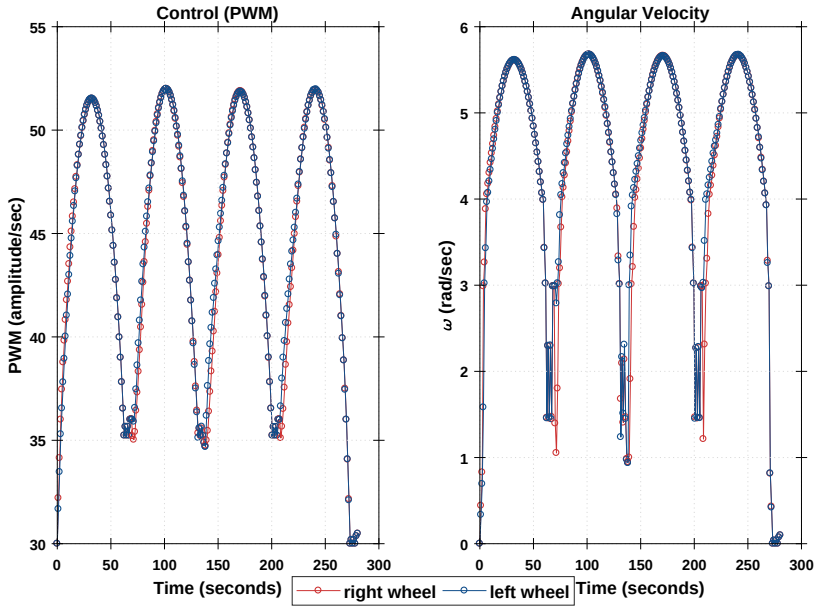

Figure 4.7: Simulated optimal control test for parallelogram

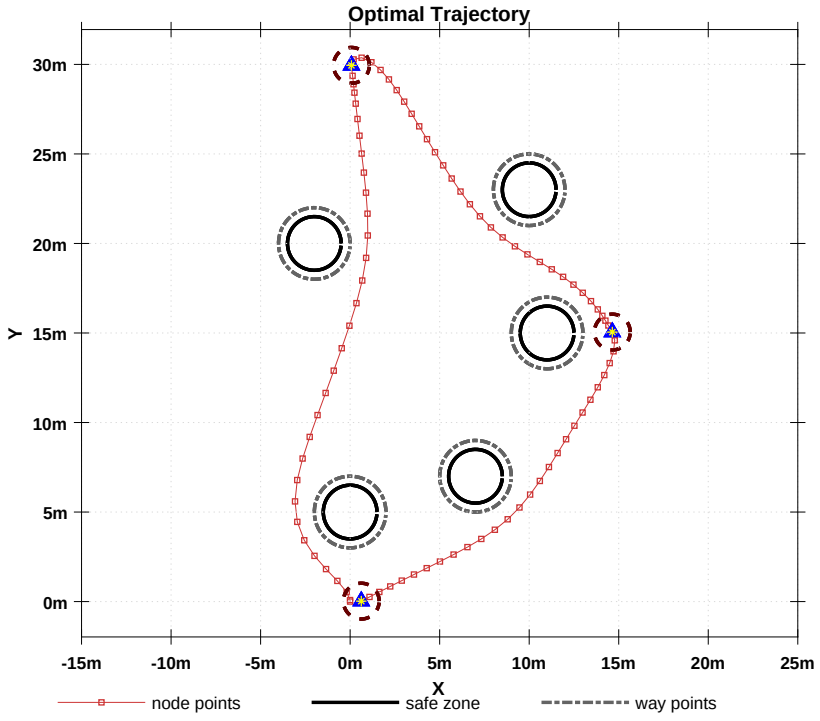

Figure 4.8: Simulated optimal trajectory with obstacles test for isosceles triangle

As shown in Figs. 4.8 and 4.9, the optimal trajectory generation algorithm starts to incorporate each obstacle starting from the initial node, while in the case of no obstacles the trajectory moves directly towards the waypoint. As the distance between vehicle and obstacle decreases the optimizer starts generating a reference trajectory which can avoid the obstacle with considering the proximity region.

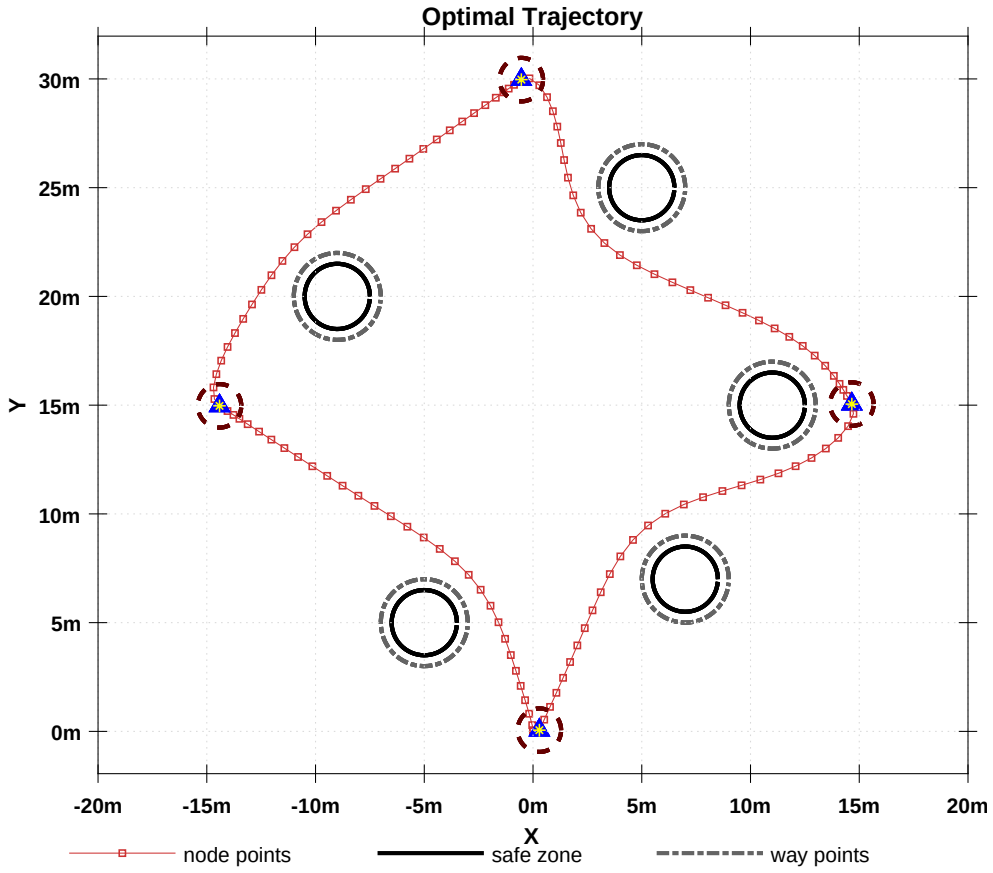

Figure 4.9: Simulated optimal trajectory with obstacles for parallelogram

The WMR can make a sudden change in heading angle at the waypoint transition. However, due to the objective function, the trajectory generator enforce the WMR to turn smoothly.

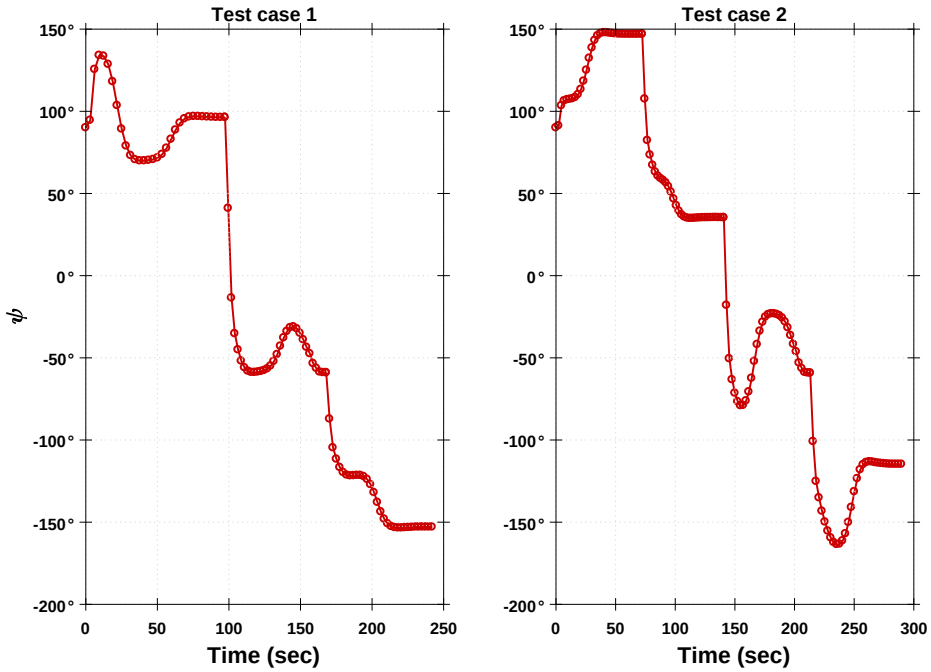

Figure 4.10: Heading angle of WMR

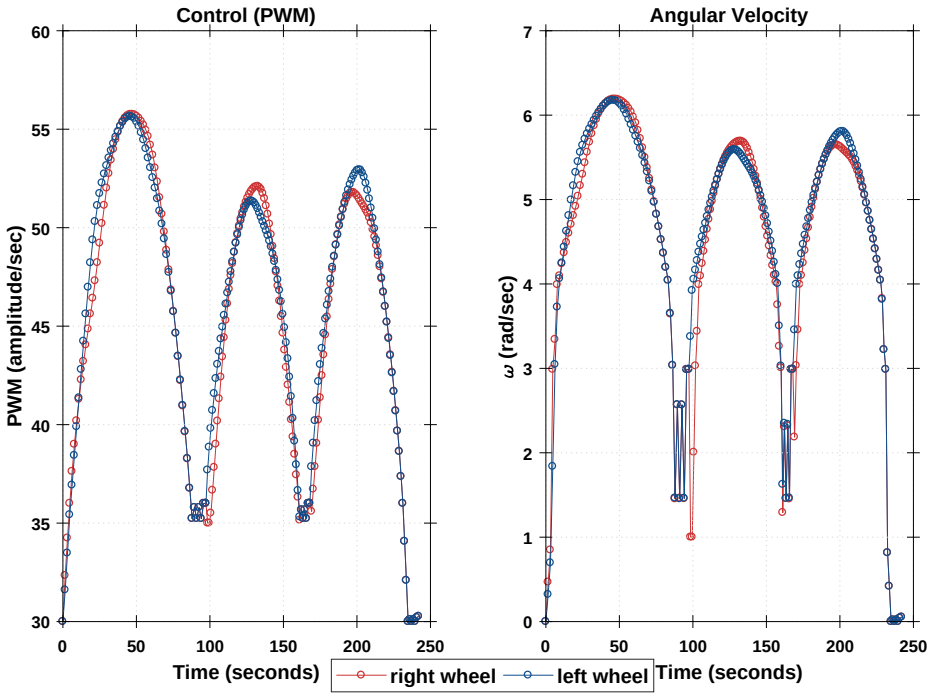

Figure 4.11: Simulated optimal control test for isosceles triangle

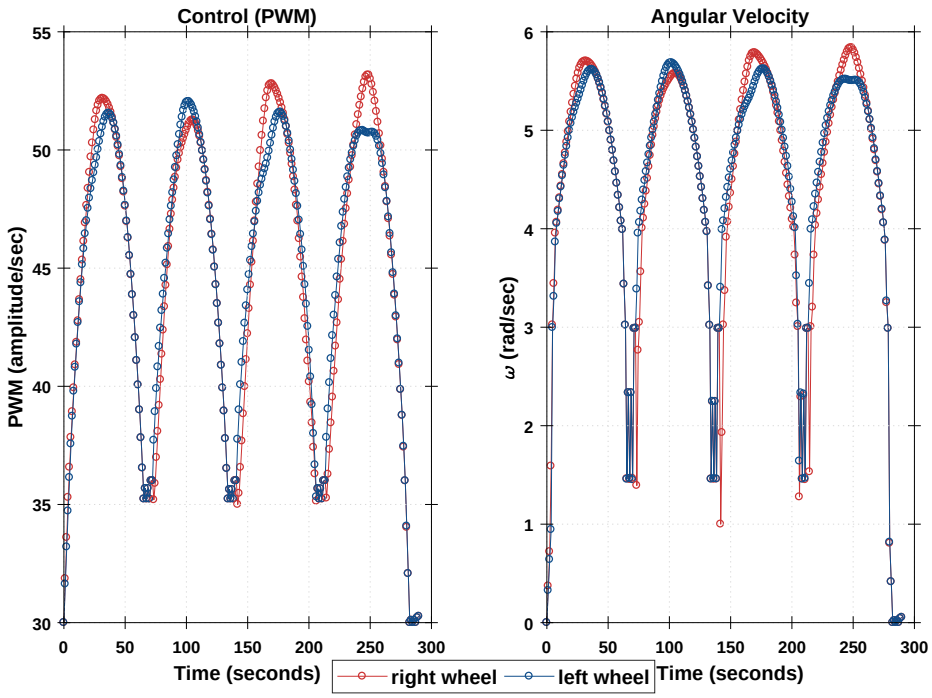

Figure 4.12: Simulated optimal control for parallelogram

As per the comparison of test cases with and without obstacle from Figs. 4.6, 4.7, 4.11, and 4.12, the trajectory with obstacles has little fluctuations in angular velocity and it also takes extra time for trajectory completion compared to trajectory without obstacles, which can be expected for additional obstacle avoidance constraint.

As illustrated in the Figs. 4.13 and 4.14, the desired reference trajectory designed for a case with more obstacles condition proves the robustness of the algorithm. The optimizer is proficient enough to design the desired reference trajectory between the wall of obstacles without collision while satisfying all constraints. Analyzing the reference trajectories, the optimizer has created different reference trajectories for all the cases. As the number of obstacles varies, we find that the optimizer varies the trajectories by considering all obstacles during the whole maneuver.

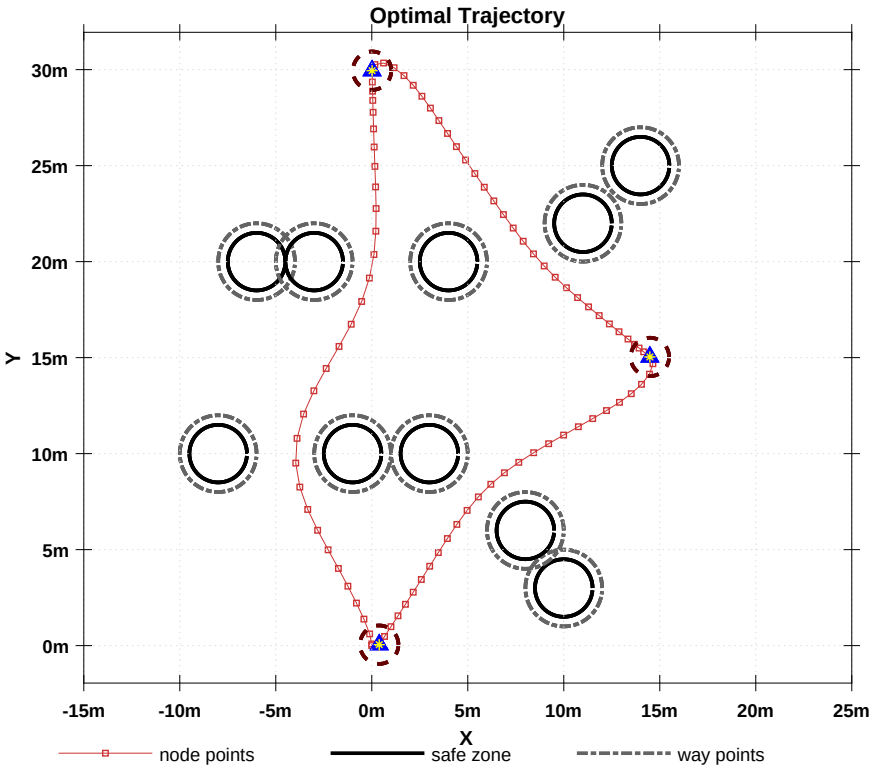

Figure 4.13: Simulated optimal trajectory with obstacles for isosceles triangle

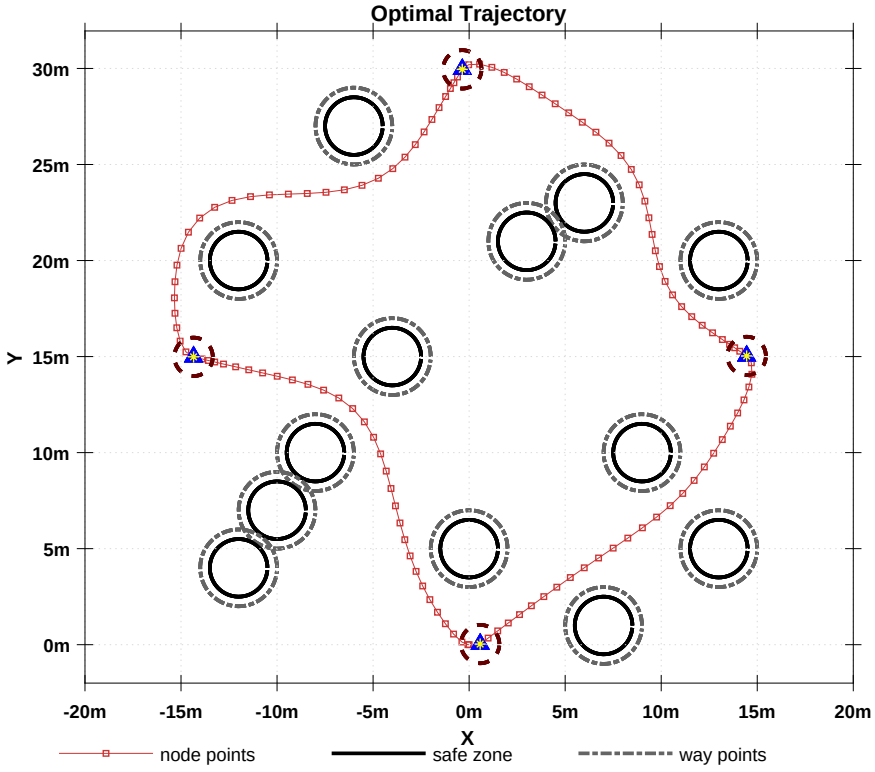

Figure 4.14: Simulated optimal trajectory with obstacles for parallelogram

In a challenging condition of 10 and 13 obstacles for respective test cases, the heading angle and trajectory control consistently meets the performance criteria.

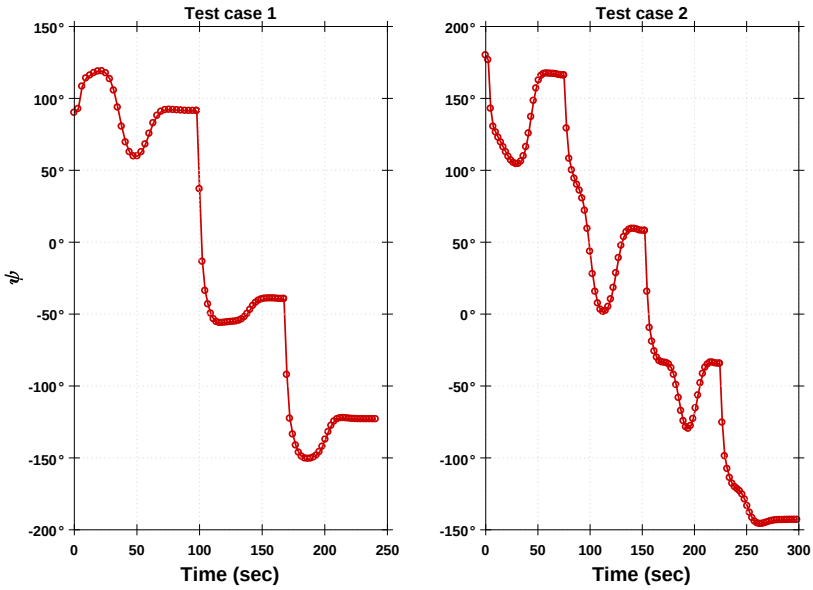

Figure 4.15: Heading angle of WMR

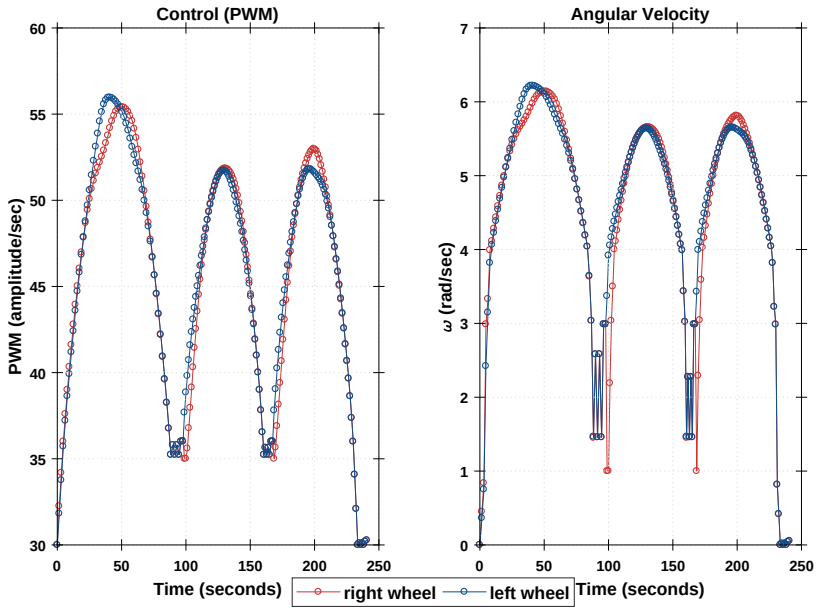

Figure 4.16: Simulated optimal ccontrol test for isosceles triangle

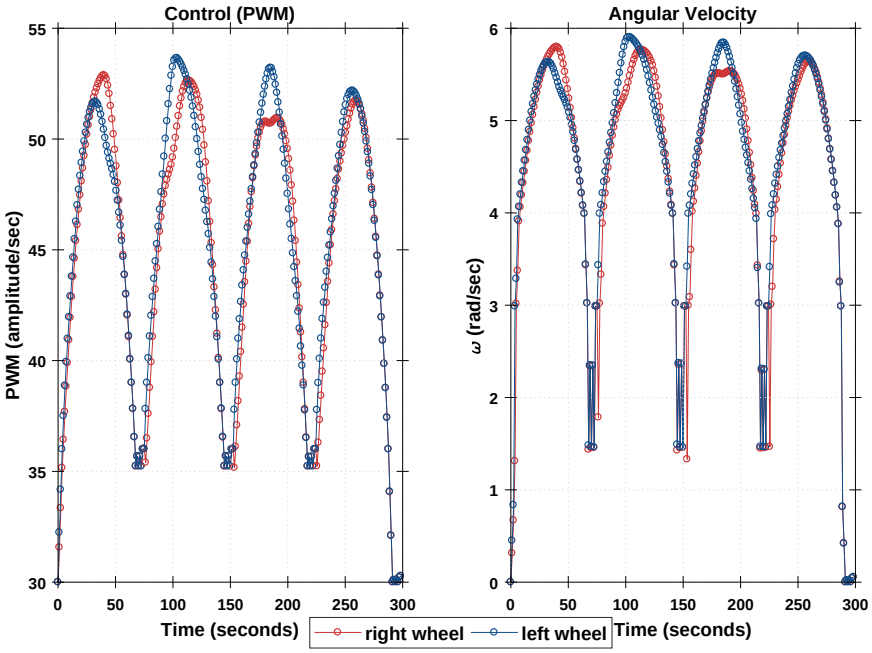

Figure 4.17: Simulated optimal control test for parallelogram

All optimization trajectories shown in this section has minimal control effort and change in heading angle, which can be implementable on a real system.

4.8 Summary

The main objective of offline trajectory planning algorithm is to develop optimal trajectory and control input that can be used in real time onboard implementation. While we run different trajectory with various environmental conditions such as different waypoint position and obstacle position, we see how wheel speed changes with environmental conditions to satisfy the performance objective and help to understand the need for an online solution.

CHAPTER 5

Experimental Setup and System Architecture

5.1 Experimental Steup

Figure 5.1: WMR built at Aerospace System Laboratory, UTA

The vehicle used to conduct the experiments presented within this research was designed as a modified version of the Mantis^{TM} off road rover chassis. The design has a motor actuator attached to each wheel making it four wheel drive and is made with aluminum shock absorbing suspension making it able to travel through different types of terrain. An illustration of the finished vehicle is given in Fig. 5.1.

The main components on-board the rover consists of a single-board computer

(Odroid-XU4), a Pixhawk autopilot and an Arduino Mega micro-controller board. The sensors used by the system are an inertial measurement unit (IMU), Global positioning system (GPS) receiver and compass which are connected through the Pixhawk autopilot as well as Quadrature encoders mounted on the wheels that are connected through the Arduino Mega. There are also four actuated motors that are controlled through the Arduino interface. The information on-board the rover itself like pixhawk sensor data, encoder data is passed through the Odroid computer which is then transmitted to a ground station computer where it is processed. The ground station computer is connected to the rover system through a wireless network and is used to handle the computations needed for localization and control of the rover. The information flow chart for the rover system is displayed in Fig. 5.2.

5.2 CPS architecture

CPS shown in Fig. 5.2 depicts how the system on the rover manages the physical processes involved and communicates with a network connected ground station to process the data. The communication between the devices included in the system architecture is handled through the Robot Operating System (ROS)[1]. This open source framework facilitates the communication of information between different components through network connections. Therefore, it renders the CPS architecture incorporated in this design possible.

There will be a few pitfalls to overcome with the CPS based architecture employed on the rover. Chief among them is the presence of network communication delays shown in Sec. 5.4. These kinds of difficulties arise due to the strength of the

[1] `http://www.ros.org/about-ros/`

network being used and the capabilities of the hardware involved. The results will demonstrate the effects of these delays and any other issues that can arise.

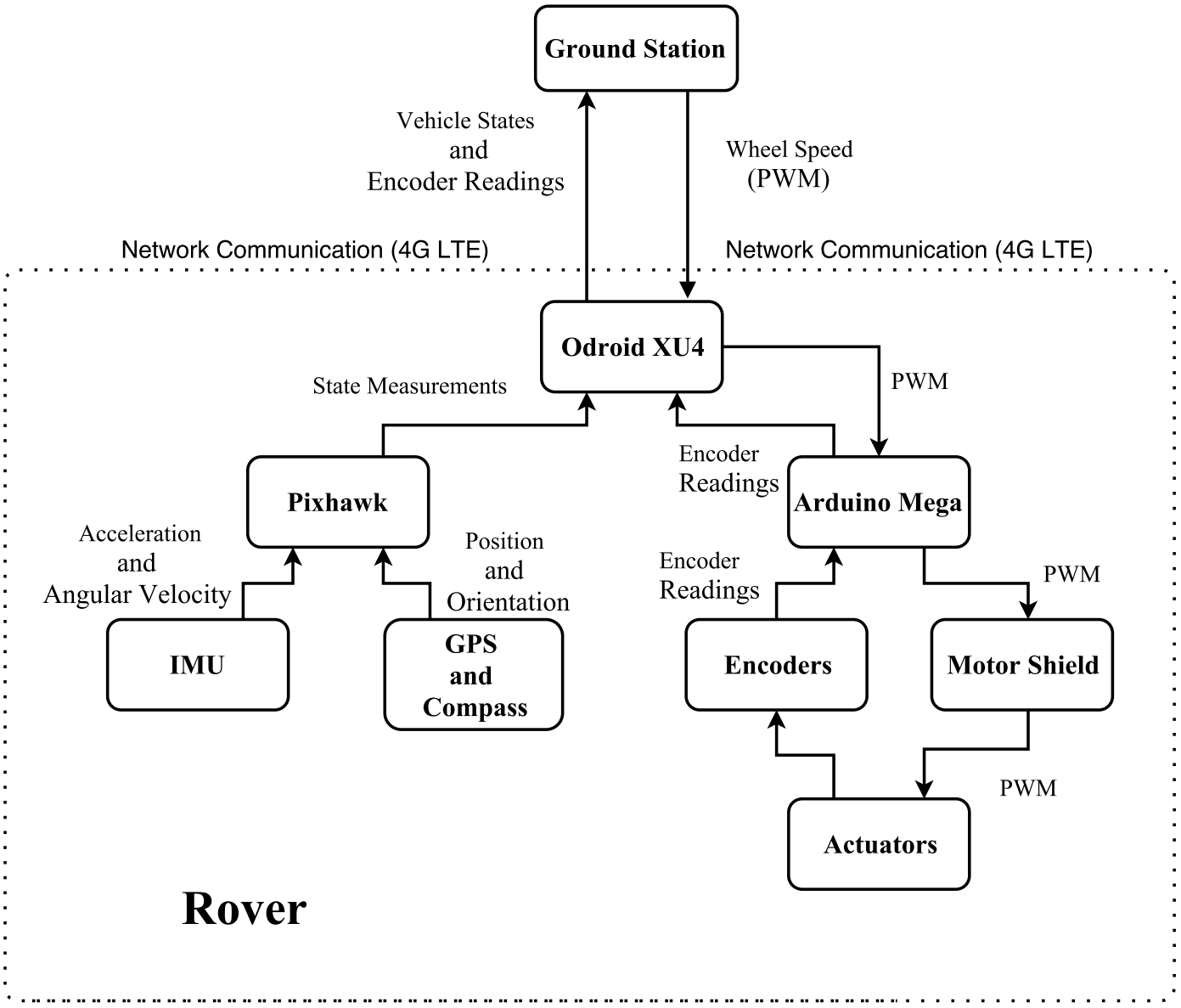

Figure 5.2: CPS architecture for the communication between rover and ground station

The experiments are conducted in an outdoor environment with user-defined waypoints. The network used for communication will be a 4G mobile hotspot with

the system architecture depicted in Fig. 5.2. Additionally, the localization of the rover will be handled through the encoder readings and GPS measurements. And the orientation of the rover will be handled with the compass. The overall objective and contribution of this work will be to validate the CPS architecture as described on the rover vehicle and its application to handling some basic tasks.

Ground station shown in Fig. 5.2 uses MATLAB/SIMULINK along with 'Robotics System Toolbox' for processing the incoming data. Trajectory generation, calculating control signals and processing encoder readings are done on ground station. Processing GPS measurements, calculating network communication delays, etc. are handled on rover vehicle and sent to ground station through network.

The IMU inside pixhawk has an update rate of 25 Hz, GPS and compass connected to pixhawk has an update rate of 5 Hz, and encoders attached to motors connected through Arduino has an update rate based on the motor speed as hardware interrupts on micro-controller is used to get the encoder pulses. So the ground station is maintained at an update of 10 Hz if the encoders are used for rover localization and 5 Hz if the GPS readings are used for rover localization. The results presented in the subsequent sections uses Earth-North-Up (ENU) based local navigation frame for rover localization with rover starting position as ENU frame origin (0,0).

5.3 Relationship between Angular velocity and Motor PWM

To design an optimal trajectory that satisfies given objective functions, optimal control input is required. For the equations of motion proposed in Eq. 2.9, angular velocity is a control input for the desired rotation of wheels. In-order to supply the control signal to physical systems such as the motor which requires PWM, an appropriate relationship between the angular velocity and motor PWM voltage signals

are required. The PWM is considered as the control input for the trajectory design instead of the wheel speeds in order to apply realistic actuator constraints.

To get the relationship, selected PWM signals are supplied to motors with a rover on the ground for a particular timespan of δt and the distance travelled by rover is measured. By using PWM signals and distance travelled (ΔS) measurements, an angular velocity of the wheel (ω) is calculated using Eq. 5.1,

$$\omega = \frac{\Delta S}{r\ \delta t} \tag{5.1}$$

Relationship between the angular velocity of the wheels and motor PWM signal is depicted in Fig. 5.3.

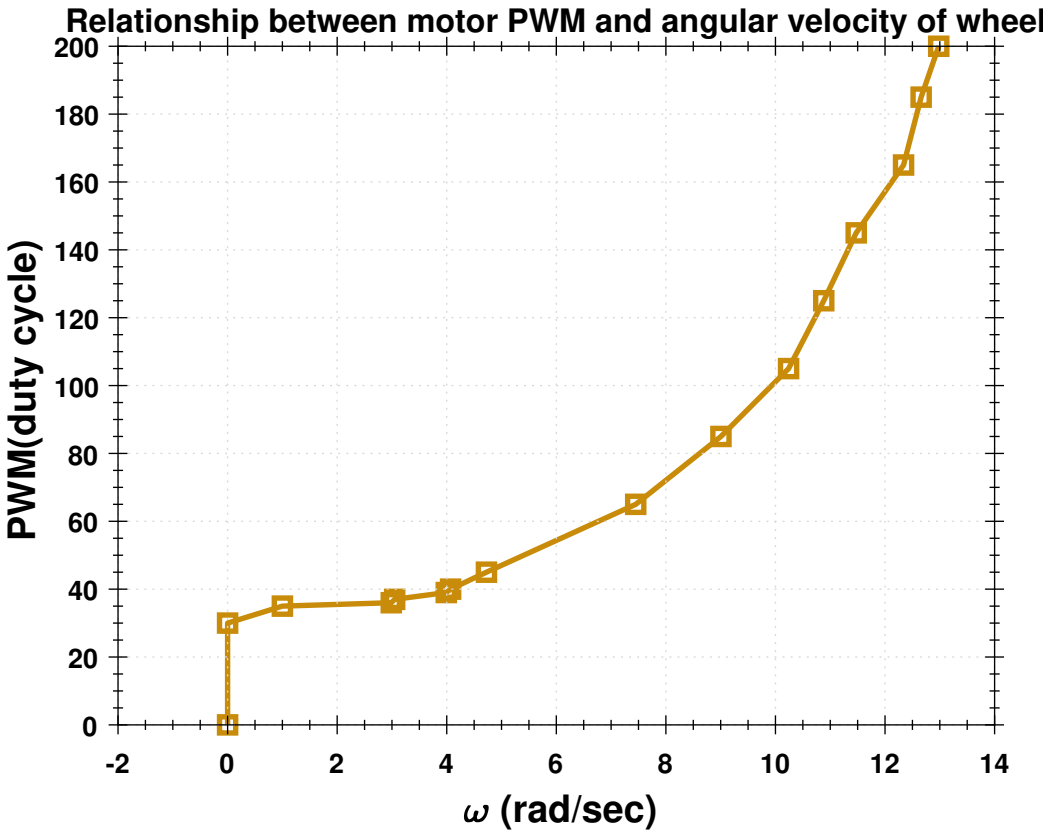

Figure 5.3: Obtained relationship between angular velocity of wheel and motor PWM

5.4 Communication Delay Characterization

In-order to measure the network communication delay between the rover and the ground station, GPS pseudo-range based technique is used. All the data before being sent to ground station is time-stamped which says the time-of-sent of a signal. And when a signal reaches the ground station, a time-of-arrival of signal is recorded. So the delay between the rover and the ground station is the difference between the time-of-sent and time-of-arrival of a particular signal. This method requires that both the clocks (on ground station and on rover) need to be in synchronization. As this is not possible because of the time-drift in systems, at a fixed time-interval (5 *sec*) the local clock offset between the ground station and the rover is measured. By adding or subtracting the local clock offset to the time-of-arrival of signal, approximate communication delay is calculated. The communication delay between rover and ground station in indoor and outdoor environment is shown in Figs. 5.4 and 5.5.

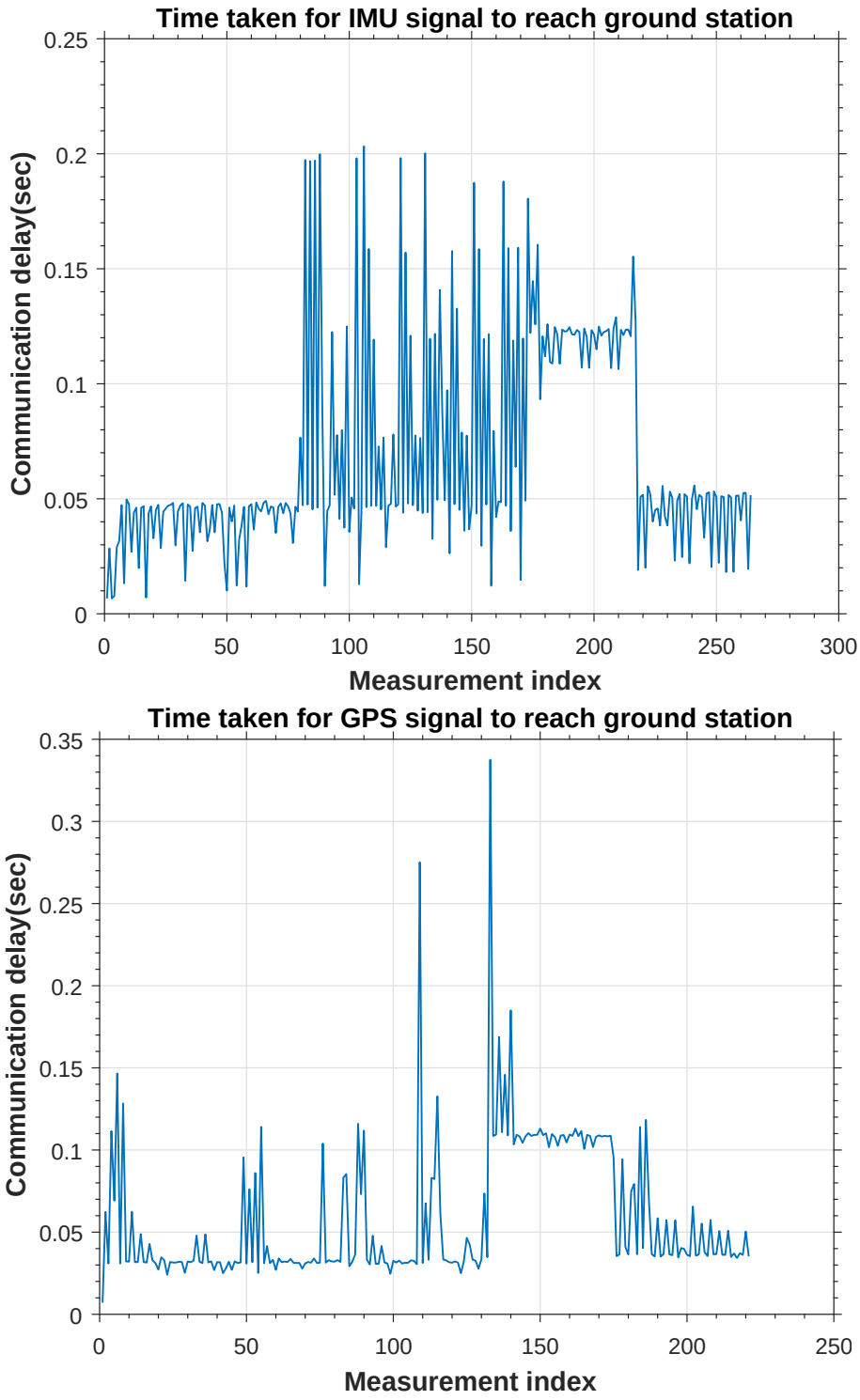

Figure 5.4: communication delay between the rover and ground station using wifi(indoor)

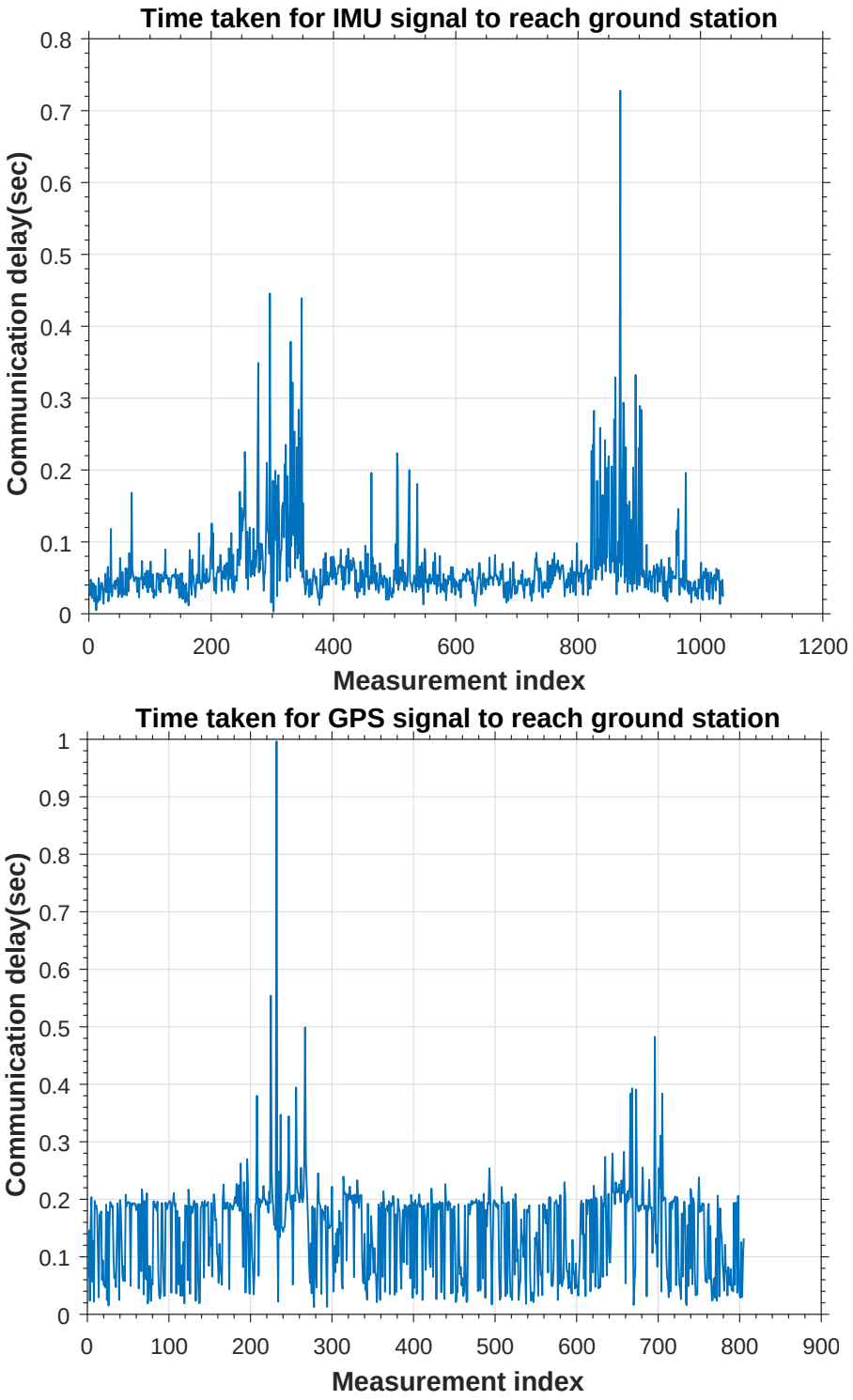

Figure 5.5: communication delay between the rover and ground station using 4G-LTE(outdoor)

CHAPTER 6

Trajectory Tracking Results

6.1 Trajectory tracking framework

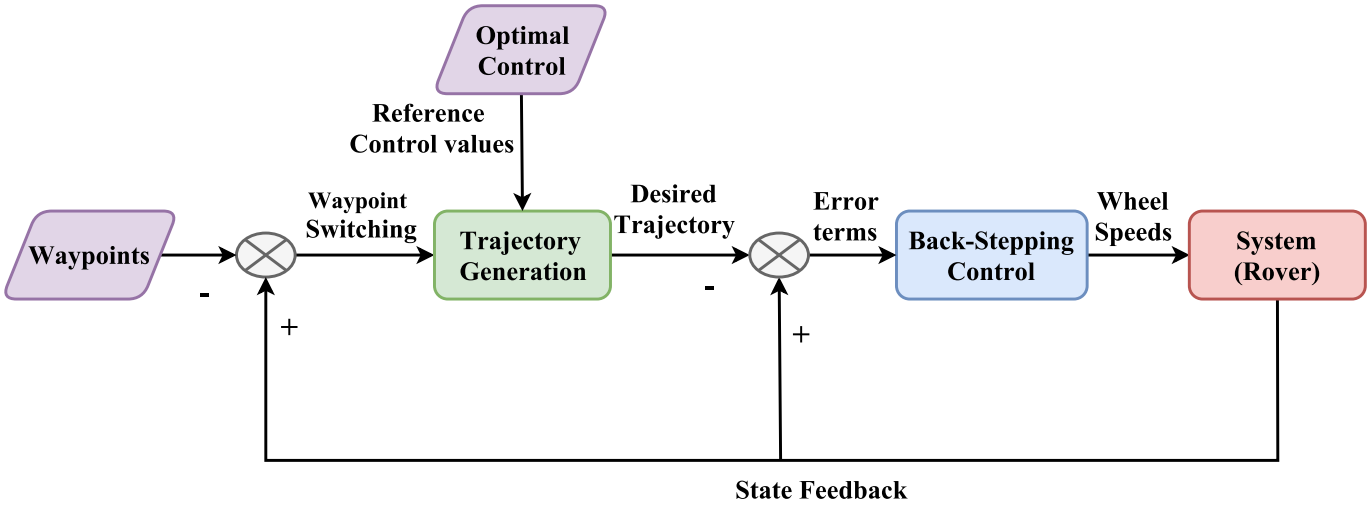

Figure 6.1: Real-time trajectory tracking framework

Real-time trajectoruy tracking framework is shown in Fig. 6.1.The optimal trajectory generator proposed in chapter. 4 and the back-stepping controller proposed in [34, 35] are respectively used to design and track a desired reference trajectory with waypoints and obstacles. To get the position feedback from the rover vehicle, different feedback methods were used i.e. GPS-Compass and Encoder-Compass. The GPS readings obtained from the rover has a covariance of about 0.5 m^2 under clear skies and about 2 *to* 3 m^2 under cloudy skies. On the other hand, encoders have good accuracy for short distances but are expected to drift with time for long distances especially when the rover turns. Moreover, compass gives accurate results during

the smooth trajectory, but sometimes due to sudden change in heading angle near wapoint switching it we get jerky behaviour.

In this section the experimental results of WMR which tracks the desired reference trajectory are shown. Obstacles, proximity region, waypoints, and safe zone are same as described in Sec. 4.7. During trajectory tracking, when the rover is within this safe-zone of a particular waypoint, then it starts to generate the reference trajectory form current position to next waypoint.

For the test cases, reference trajectories generated in Sec. 4.7 portrayed in Fig. 4.13 and 4.14 are selected.

6.2 Methode 1 : With GPS and Compass

As GPS gives the position information of the rover in global co-ordinate frame, these measurements need to be transformed to local navigation frame(ENU) in-order to localize the rover. To transform these GPS measurements to ENU frame, the GPS measurements are first transformed to Earth-Centered-Earth-Fixed (ECEF) frame by using WGS-84 ellipsoidal model given by Eq. 6.1. Then by taking the first ECEF co-ordinate as a starting position, these ECEF co-ordinates are tranformed to ENU frame using Eq's. 6.2 and 6.3.

$$\begin{aligned} N &= \frac{a}{\sqrt{1 - e^2 \sin^2 \phi}} \\ X_{ECEF} &= (N + h) \cos\phi \cos\lambda \\ Y_{ECEF} &= (N + h) \cos\phi \sin\lambda \\ Z_{ECEF} &= (N(1 - e^2) + h) \sin\phi \end{aligned} \tag{6.1}$$

$$\mathbf{r}_{NED} = \begin{pmatrix} -\sin\phi\cos\lambda & -\sin\phi\sin\lambda & \cos\phi \\ -\sin\lambda & \cos\lambda & 0 \\ -\cos\phi\cos\lambda & -\cos\phi\sin\lambda & -\sin\phi \end{pmatrix} [ECEF_{current} - ECEF_{ini}] \quad (6.2)$$

$$\mathbf{r}_{ENU} = \mathbf{R}_{NED}^{ENU}\mathbf{r}_{NED} \quad (6.3)$$

where N, is the length of the normal to the ellipsoid.

X_{ECEF}, is the X co-ordinate of the rover in ECEF frame.

Y_{ECEF}, is the Y co-ordinate of the rover in ECEF frame.

Z_{ECEF}, is the Z co-ordinate of the rover in ECEF frame.

ϕ, λ, h are the latitude, longitude and altitude at the current rover position.

semi-major axis of earth(a) = $6,378,137.0\,m$, semi-minor axis of earth (b) = $6,356,752.3142\,m$.

e, is the eccentricity of the earth.

$\mathbf{R}_{NED}^{ENU}$, is the rotation matrix from NED to ENU frame.

$$\mathbf{E}(t) = \mathbf{r}^{act}(t - \delta d) - \mathbf{r}^{ref}(t) \quad (6.4)$$

where, $\mathbf{E}$ is error between actual position ($\mathbf{r}^{act}$) and reference position ($\mathbf{r}$), and δd is the time delay of the signal to reach the ground station from the rover.

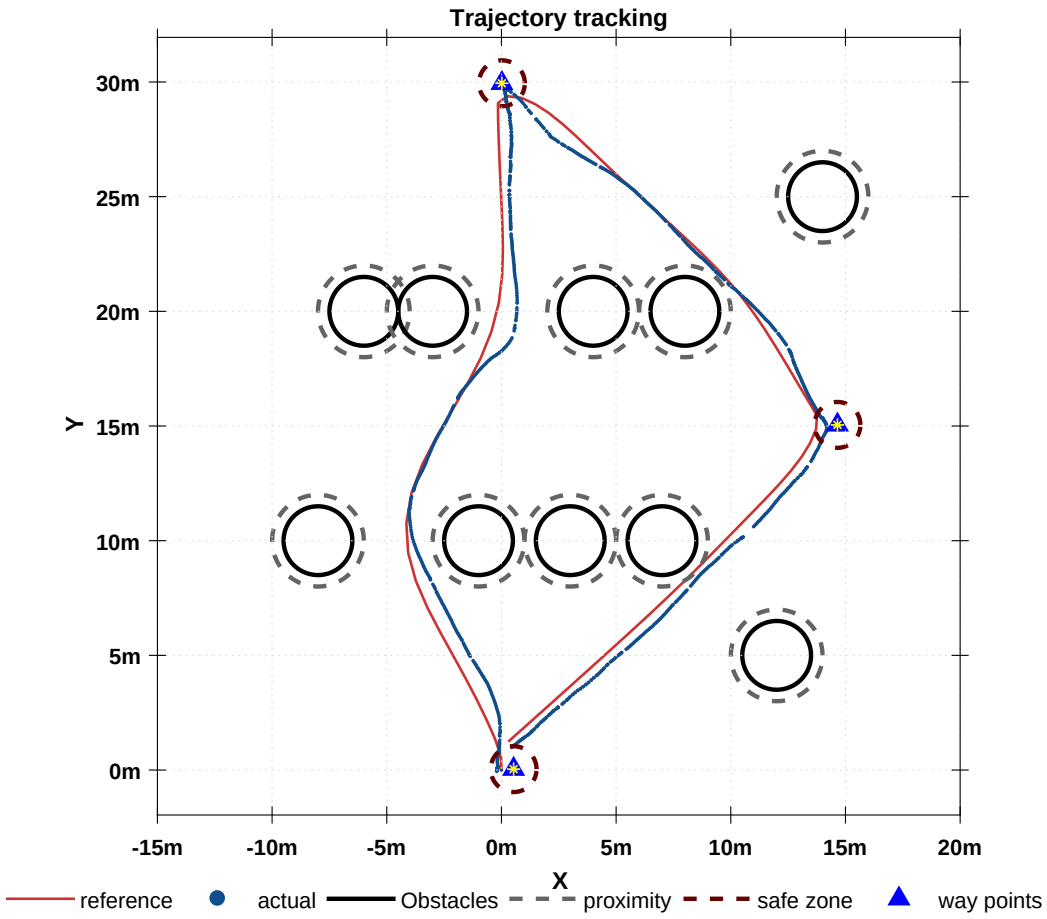

Figure 6.2: Real time trajectory tracking using GPS for isosceles triangle

The reference trajectory with multiple waypoints and obstacles designed in Sec. 4.7 tracked by WMR for different test cases using backstepping controller proposed in [34, 35] are illustrated in Fig. 6.2 and 6.3. It shows that the WMR tracked the reference trajectory well although a presence of the network communication delay was between the ground station and the WMR. In test case 1, real time trajectory deviates from the reference trajectory, around 0.9 m. The reference trajectory does pass through the final waypoint, this minor deviation is acceptable in presence of communication delay and other uncertainties.

Similarly in test case 2, the actual trajectory deviates from the path due to dif-

ferent initial heading angle and/or communication delay, apart from that it tracked the reference trajectory very well. In both test cases, trajectory tracking has nominal error is around 0.9 m as shown in Fig. 6.4.

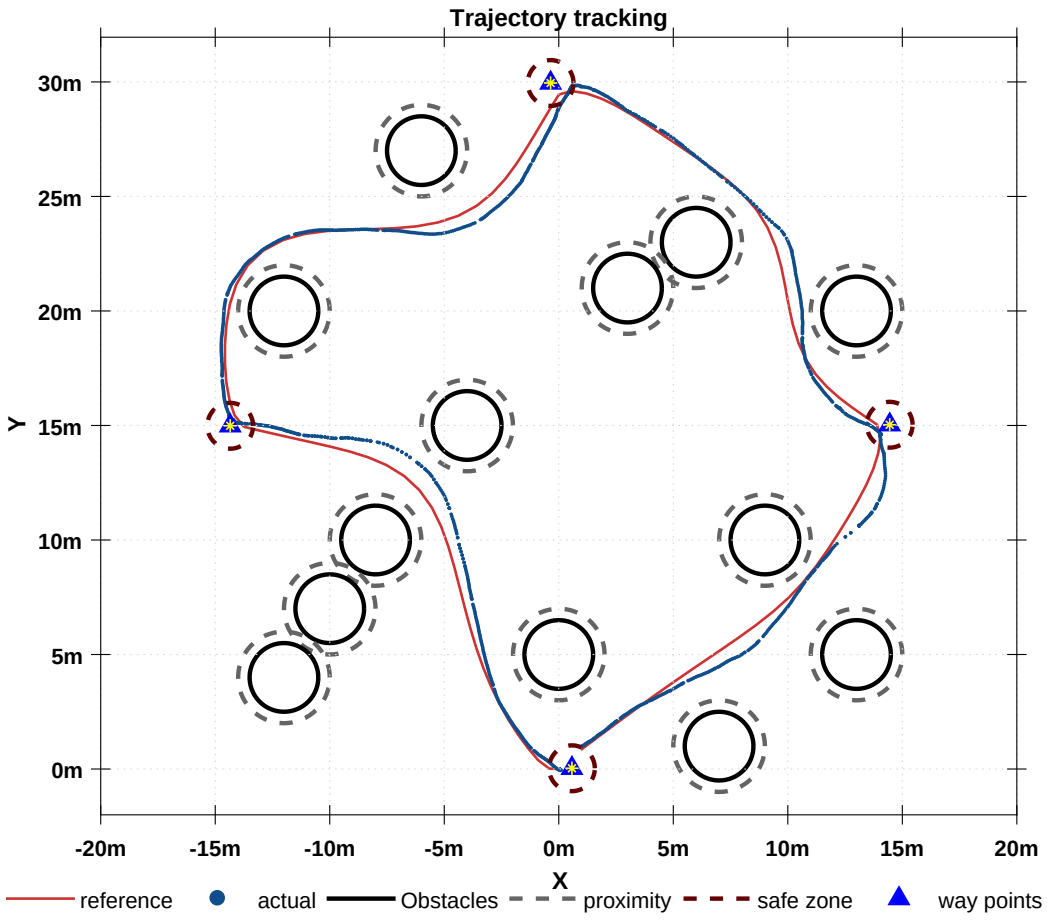

Figure 6.3: Real time trajectory tracking using GPS for parallelogram

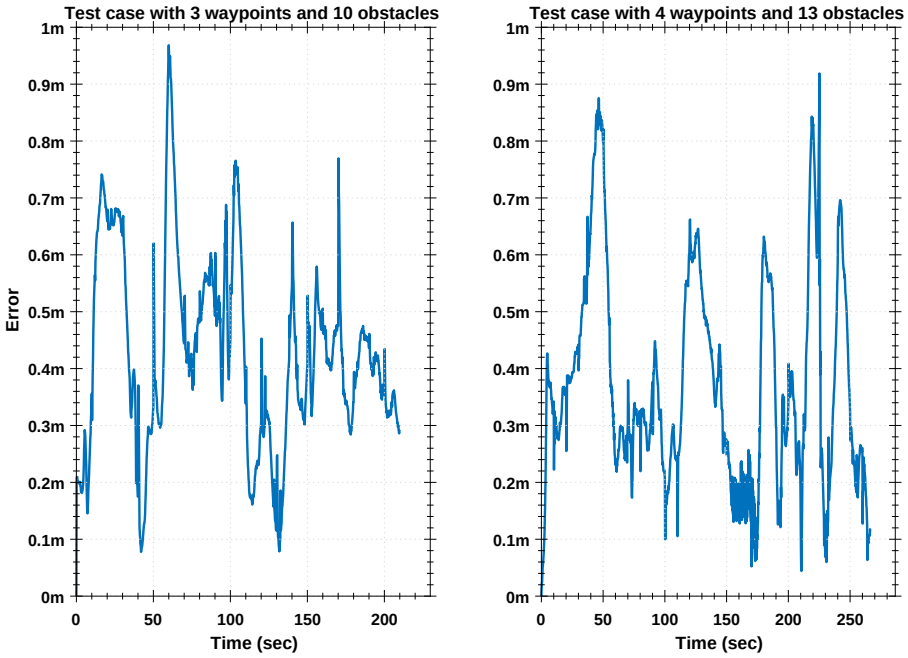

Figure 6.4: Error in position between reference and actual trajectory of the WMR with GPS and Compass for state-feedback

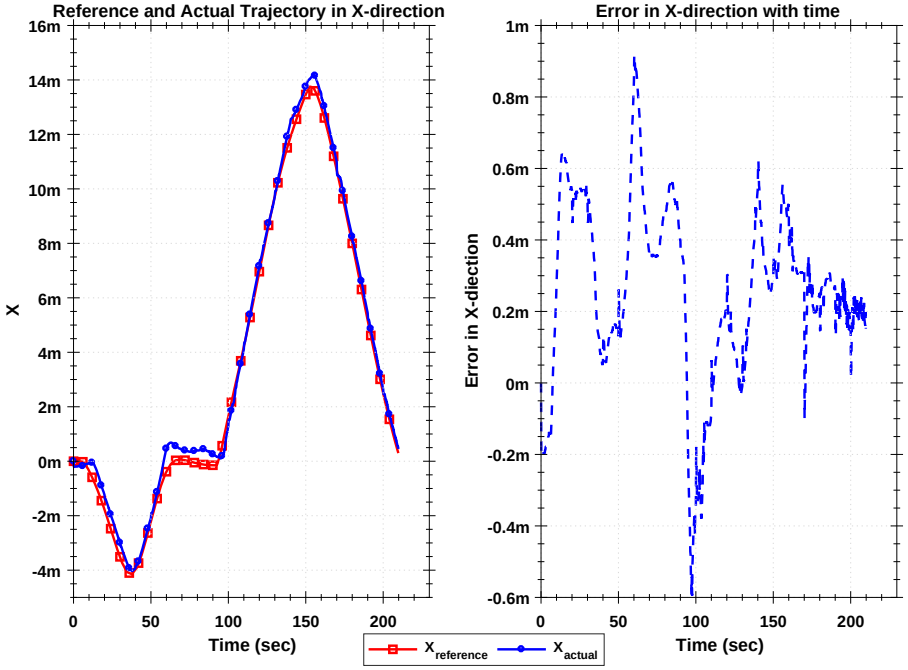

Figure 6.5: Trajectory tracking in X - direction using GPS for isosceles triangle

The trajectory tracking and error of the rover in x, y direction are shown in Fig. 6.5, 6.6, 6.7, and 6.8. When the vehicle enters into the safe zone of a waypoint, the trajectory generator starts to create the trajectory for next waypoint. As the new trajectory starts from the current position of the rover, the position error of the rover at the waypoint transition is less.

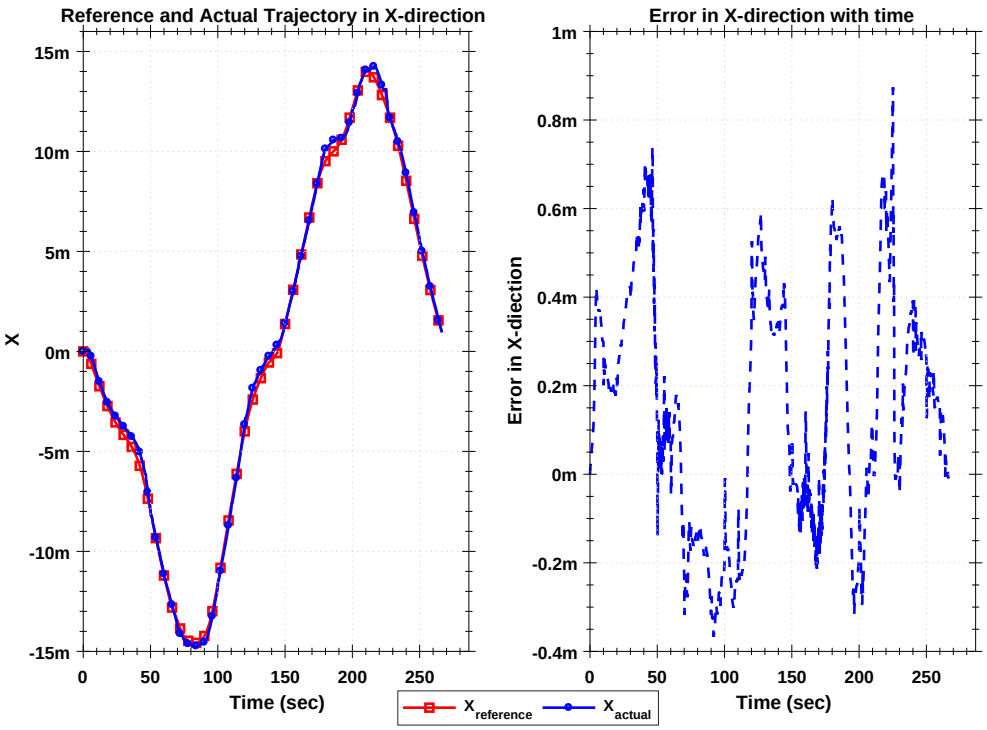

Figure 6.6: Trajectory tracking in X - direction using GPS for parallelogram

During the obstacle avoidance, errors in the x, y direction is varies about -0.4 *to* 0.8 m, which is expected. This error might be due to certain issues,such as different heading angle at initial node or GPS signal loss or the network communication delay.

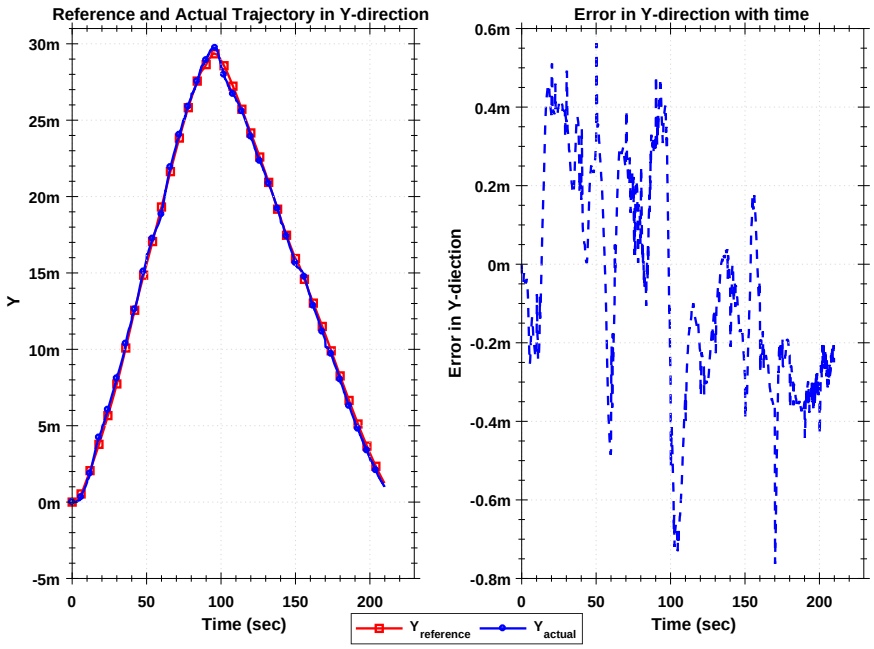

Figure 6.7: Trajectory tracking in Y - direction using GPS for isosceles triangle

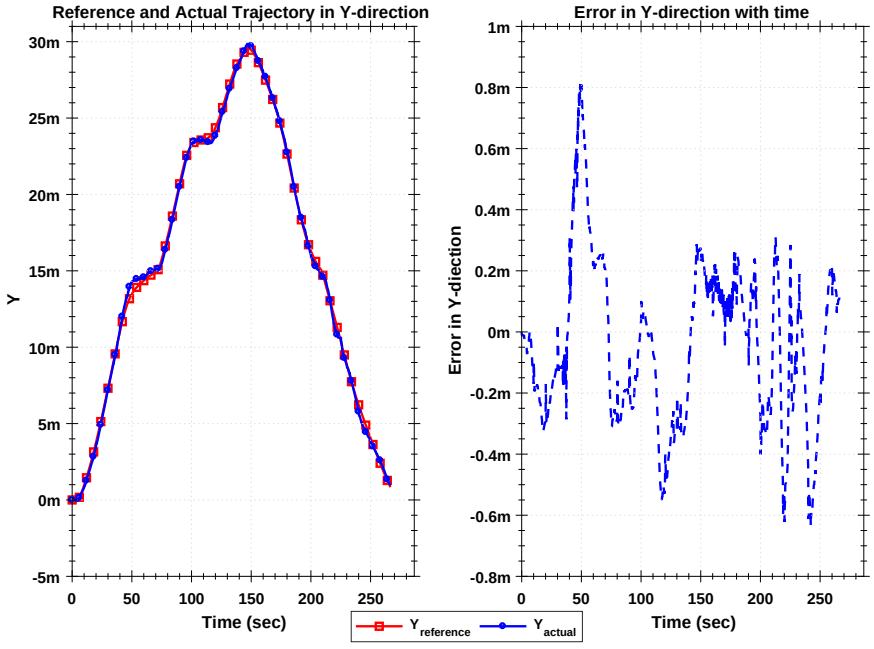

Figure 6.8: Trajectory tracking in Y - direction using GPS for parallelogram

In test case 1, initial heading angle error was around 20° between reference and actual trajectory, due to difference in initial states of WMR and reference trajectory. The controller tries to track the reference trajectory properly, but sometime in complex scenario i.e. wall of obstacles and also presence of mechanical error i.e. compass drift and wheel drift or communication delay there are some peaks in the error time histories. Moreover, due to sudden waypoint switching there are peaks in heading angle error time histories.

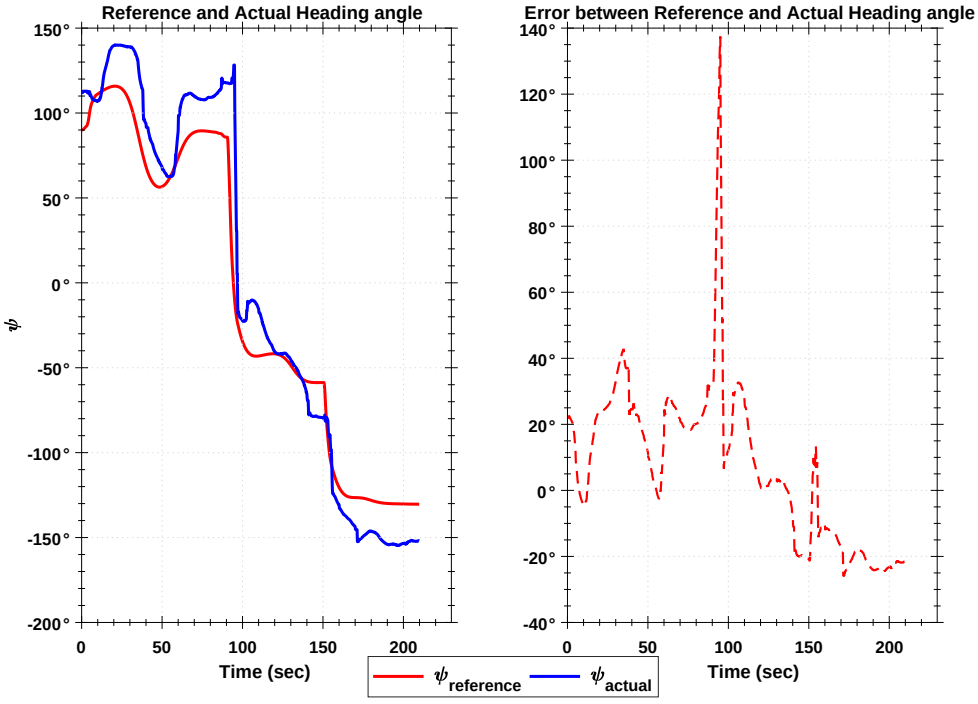

Figure 6.9: Heading angle using GPS and Compass for isosceles triangle

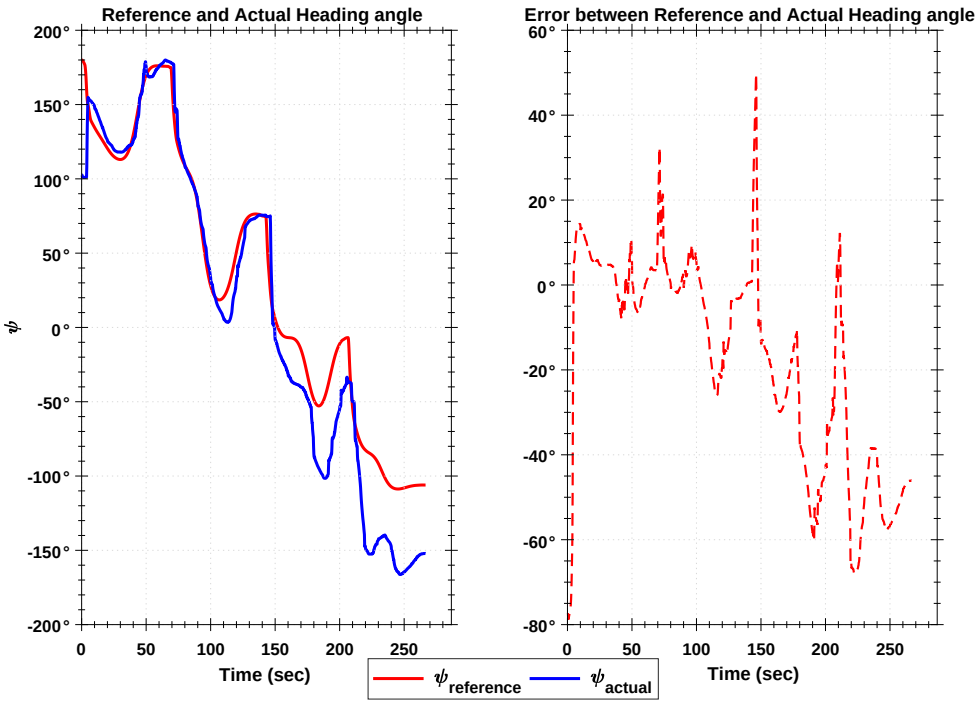

Figure 6.10: Heading angle using GPS and Compass for parallelogram

The actual angular velocity of the vehicle is calculated using the motor encoders readings and Eq. 6.5,

$$\omega_{actual} = \frac{(\delta_{et})}{r\eta_{et}\delta t} \tag{6.5}$$

As desired reference trajectory requires minimum control effort, angular velocity of WMR smoothly increase to a maximum at $\frac{number\ of\ nodes}{2}$ and gradually decrease at come nearer to final node point.

Note, where r is radius of the wheel (m). δ_{et} are the number of ticks that elapsed since the last known encoder measurement. η_{et} is number of encoder ticks that would be obtained if rover travels $1m$. Time elapsed since the last known encoder measurement is denoted as δt.

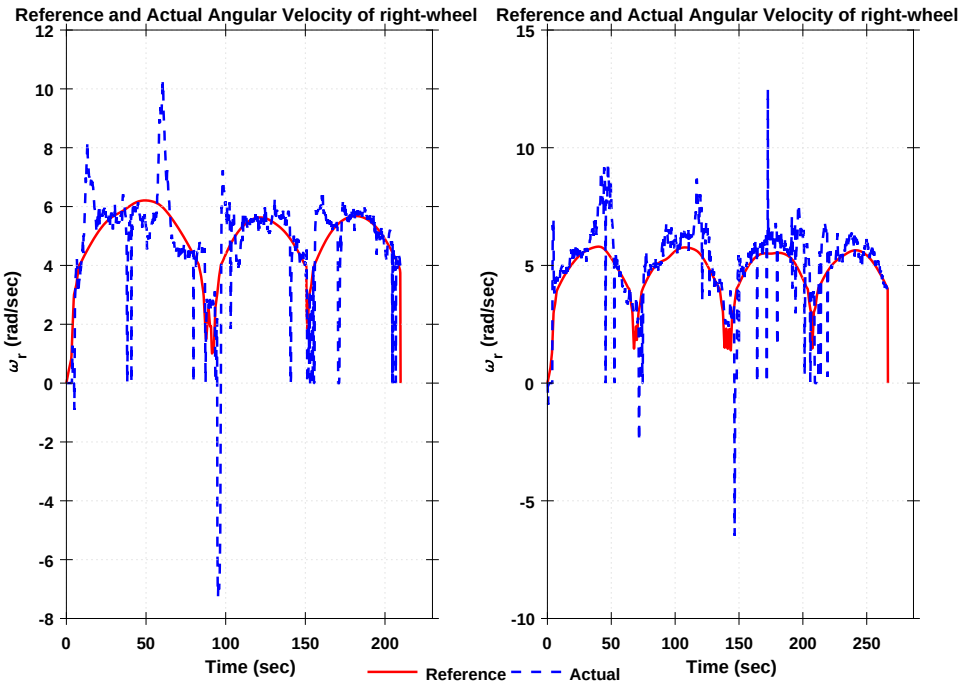

Figure 6.11: Angular acceleration of right wheel using GPS

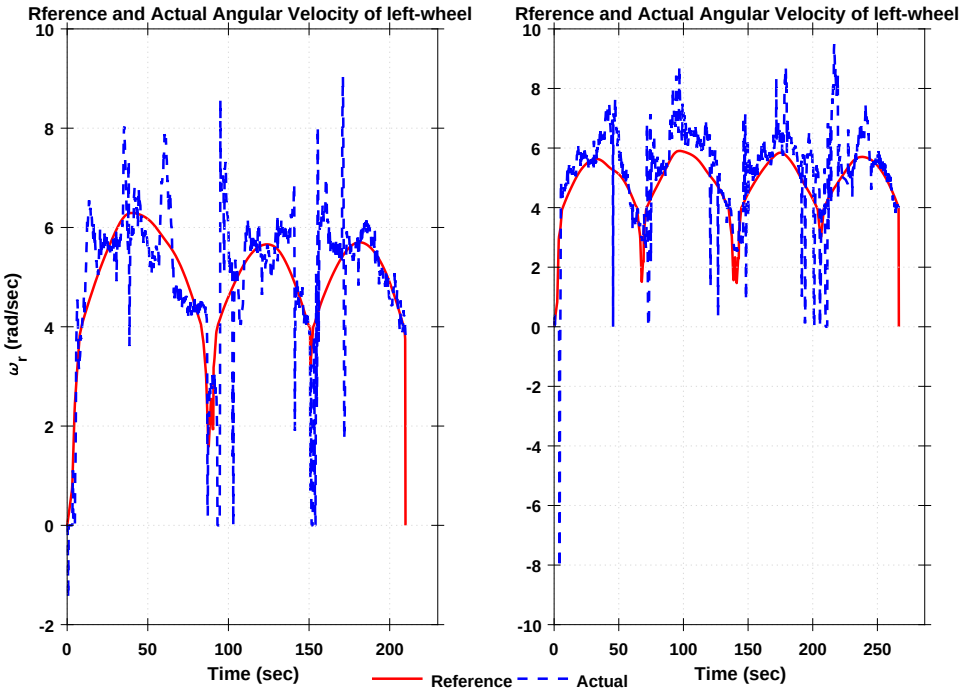

Figure 6.12: Control of the WMR

6.3 Method 2: With Encoder and compass

In this subsection , quadrature encoders and compass are used for position and heading angle feedback respectively. The quadrature encoder has a resolution of 4500 pulses/wheel rotation (or ticks/wheel rotation). The position of WMR is calculated, by integrating the state differential equation using the count of encoder pulses. Wheel slippage introduces some measurement noise, while update rate of micro-controller due to high wheel speed introduces some bias. This measurement noise and bias creates linear drift in encoder reading. Due to that for long distance trajectory, position measurement with encoder reading is not much reliable while for short distance its somewhat accurate.

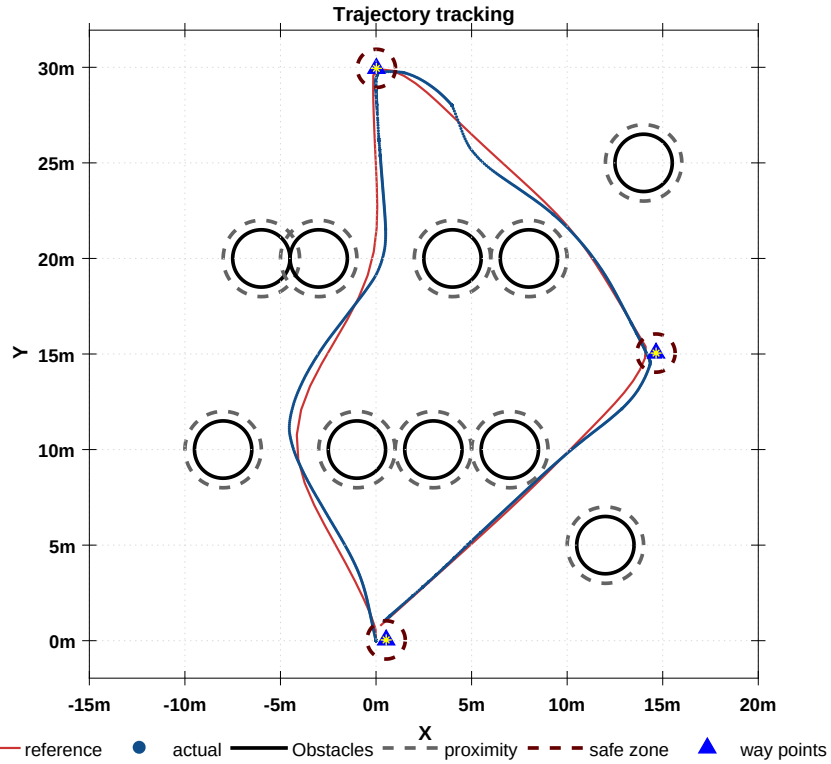

Figure 6.13: Real time trajectory tracking using Encoder and Compass for isosceles triangle

In simulation the trajectory tracking of the rover for both the test cases by using encoders and compass for state-feedback is depicted in Fig. 6.13 and 6.14. It seems that the rover followed the reference trajectory without penetrating the obstacles and travels with less drift. But in the actual test the scenario was different, rover drifted a lot during the maneuver and finished the trajectory 4 *to* 6 m away from the goal. The rover reached the waypoints even though there was drift in position, because a magnetic compass was used for heading angle reference.

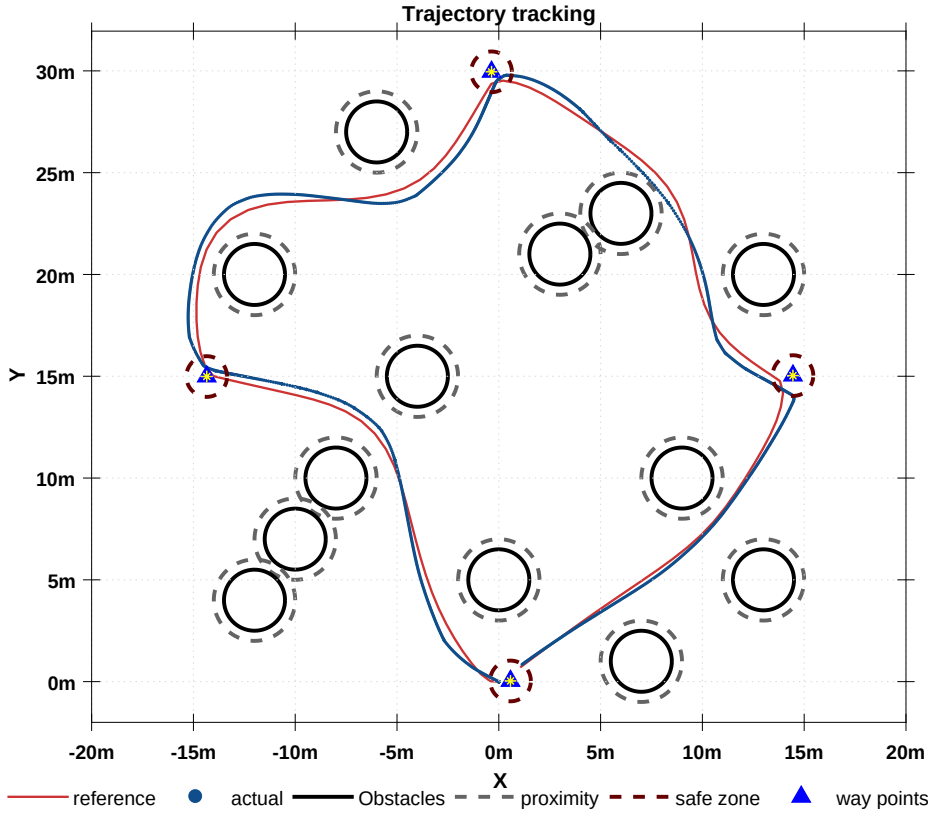

Figure 6.14: Real time trajectory tracking using Encoder and Compass for parallelogram

Fig. 6.15 and 6.4 reveals that trajectory tracking using encoder reading has more trajectory error compared to GPS reading, still encoder measurements help to

complete the multiple waypoints trajectory without colliding with obstacles. There are some peaks in the error trajectory, which is due to the communication delay.

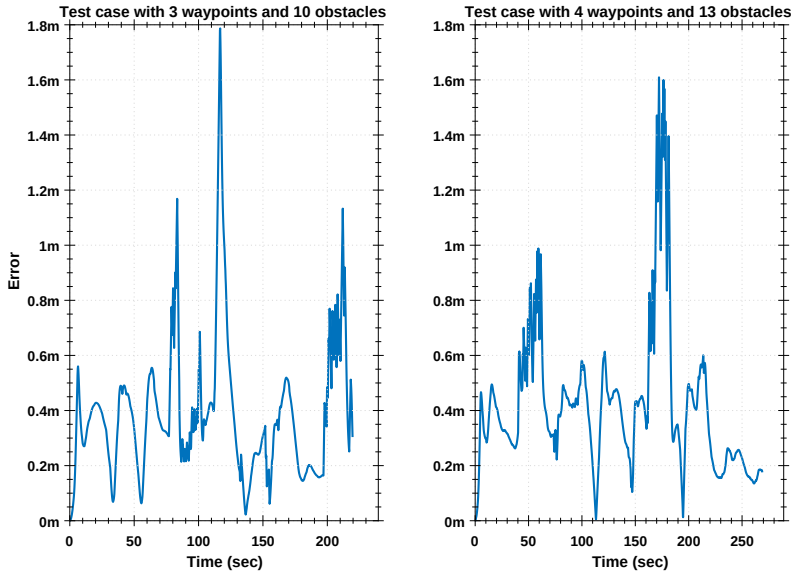

Figure 6.15: Error in position between reference and actual trajectory of the WMR with GPS and Compass for state-feedback

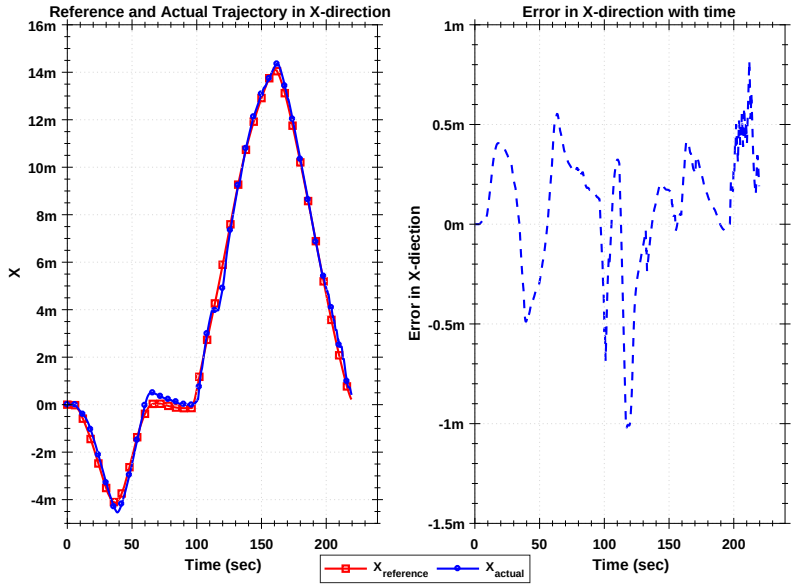

Figure 6.16: Trajectory tracking in X - direction using Encoder for isosceles triangle

The trajectory tracking and error of the rover in x, y direction are shown in Fig. 6.16, 6.18, 6.17 6.19. For both cases X-Y direction trajectory nicely tracked the reference trajectory. There are some peaks, which is mainly due to communication delay.

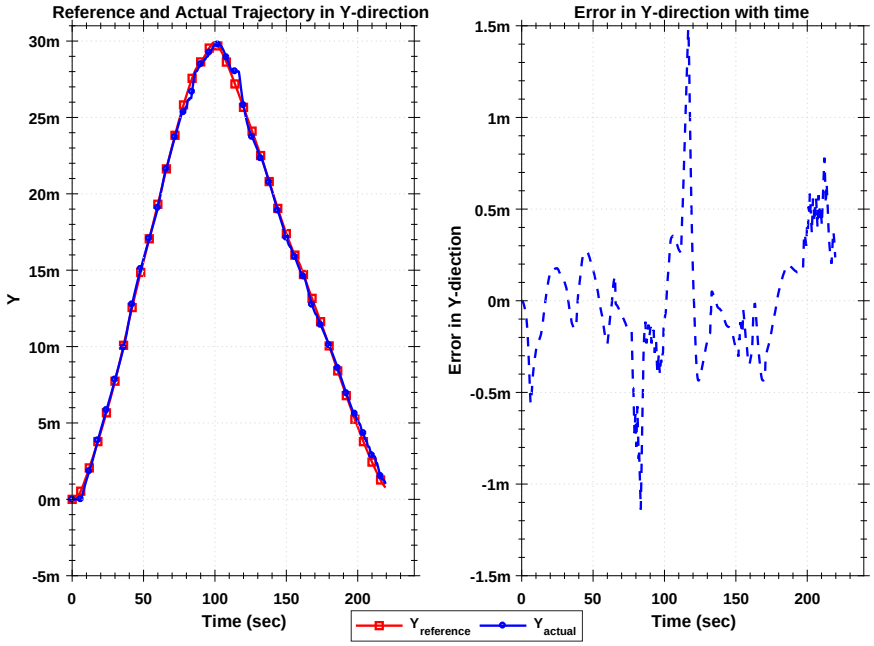

Figure 6.17: Trajectory tracking in Y - direction using Encoder for isosceles triangle

Moreover, when the vehicle enters into the safe zone of a waypoint, trajectory generator starts to create the trajectory for next waypoint. As the new trajectory starts from the current position of the rover, the position error of the rover at waypoint transition become less. On the other side, heading angle error increases due to the sudden switch of the waypoint.

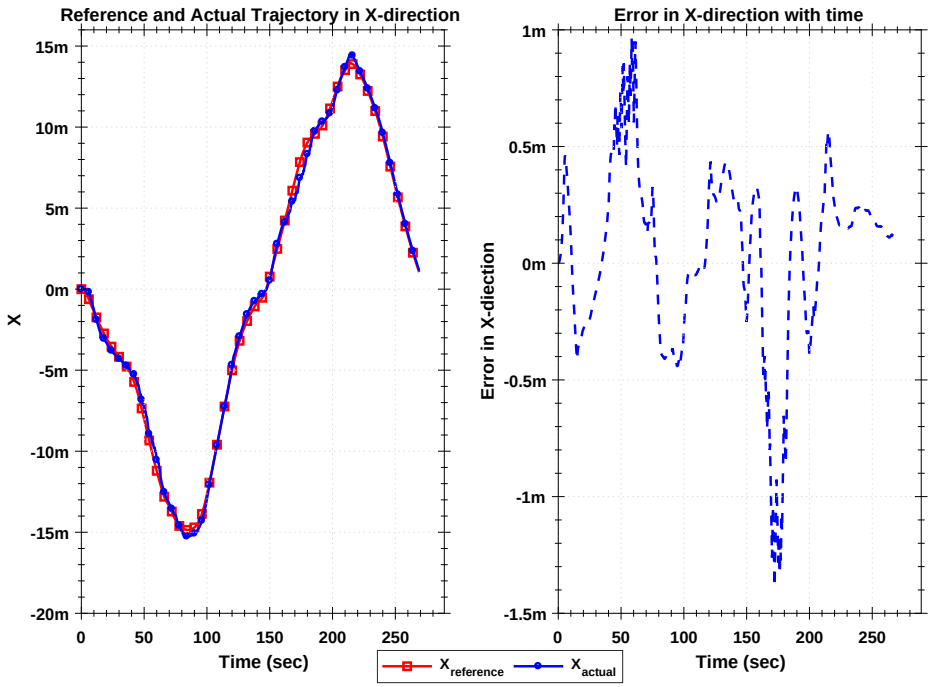

Figure 6.18: Trajectory tracking in X - direction using Encoder for parallelogram

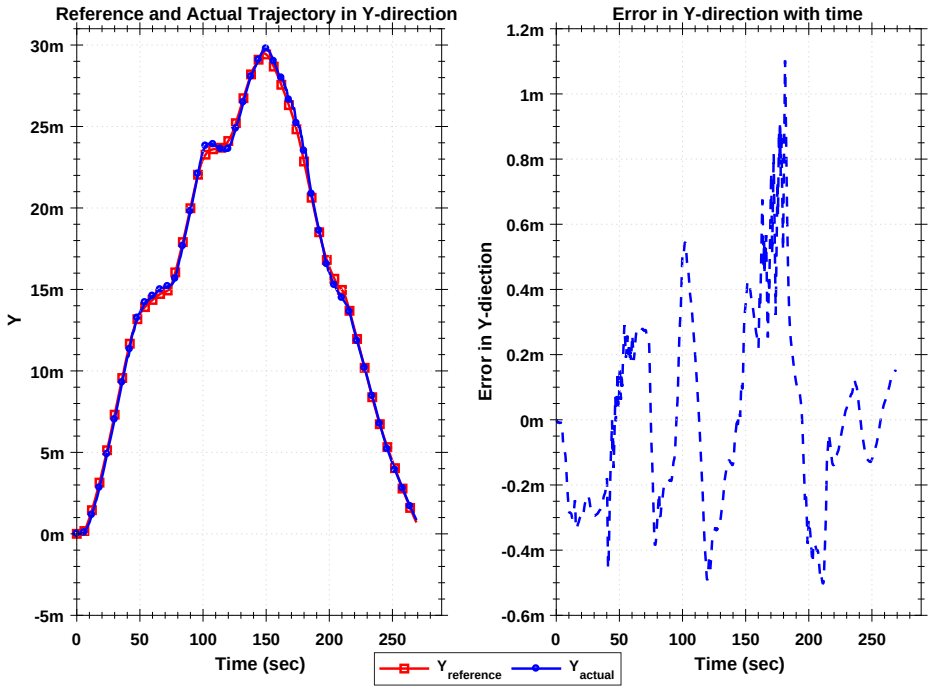

Figure 6.19: Trajectory tracking in Y - direction using Encoder for parallelogram

Comparing Fig. 6.9, 6.10, 6.20, and 6.21, we see that heading angle errors were almost same in bound of $\pm$ 20°. At initial node, error is high due to initial heading difference, and due to communication delay and sudden change of waypoint, we see some peaks in the middle of the trajectory.

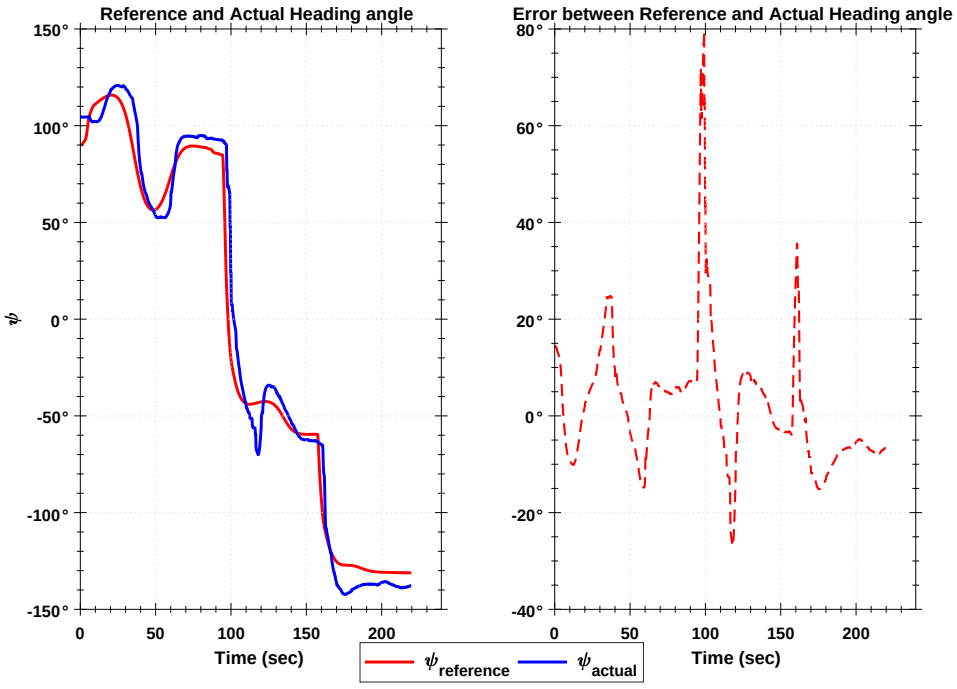

Figure 6.20: Heading angle using Encoder and Compass for isosceles triangle

Fig. 6.22 and 6.23 describe the right and left wheel angular velocity for both cases. The plots shows some peaks at similar time-stamps noted in trajectory error plot. After sudden increase or drop in measurements due to the communication delay, controller tries to reduce the states error by modifying commanded control inputs.

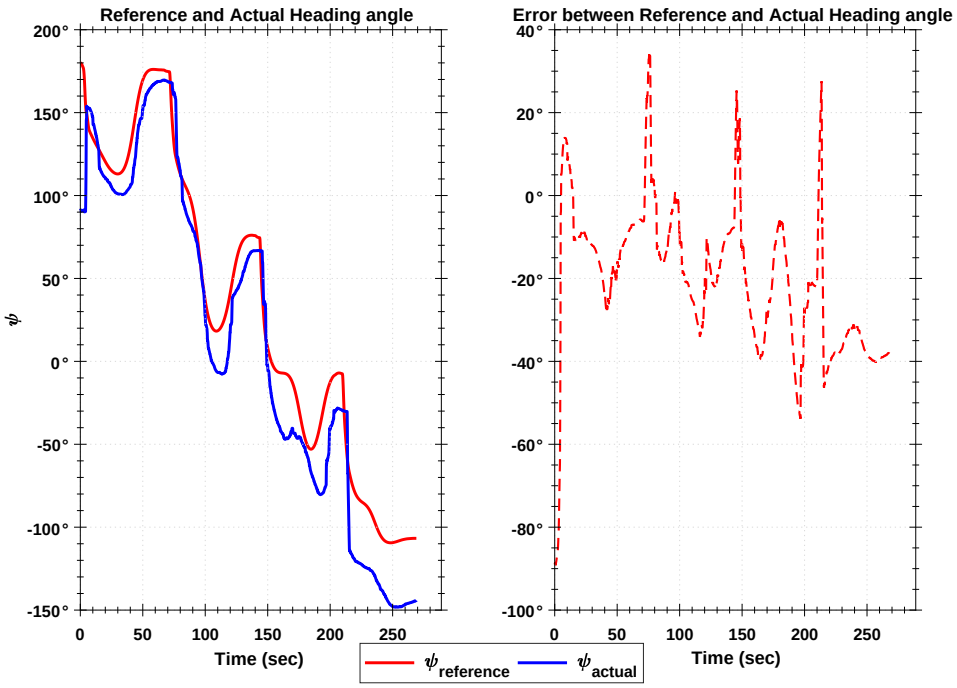

Figure 6.21: Heading angle using Encoder and Compass for parallelogram

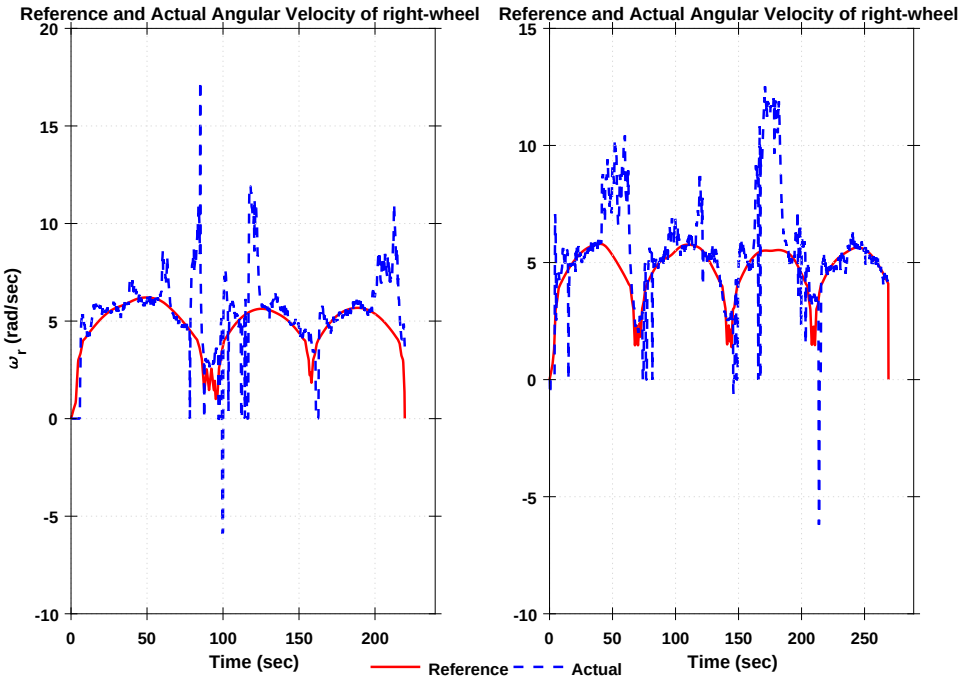

Figure 6.22: Angular acceleration of left wheel using Encoder

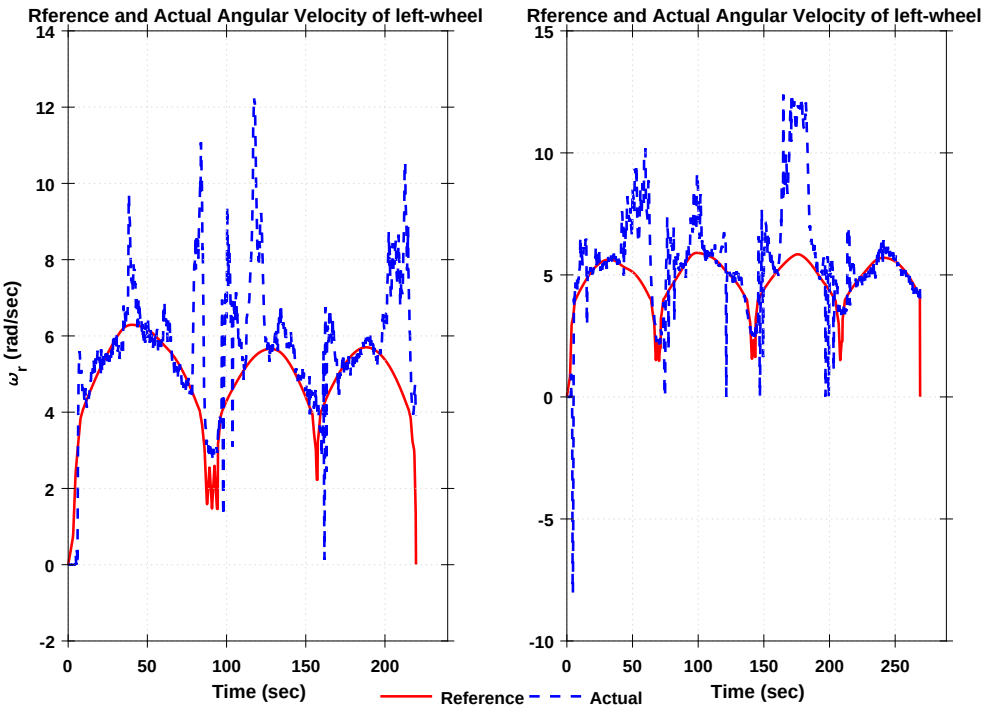

Figure 6.23: Angular acceleration of left wheel using Encoder

6.4 Special Test case:

In this section trajectory tracking with more communication delay and GPS data drop is shown. Figs. 6.24 and 6.25 depicts the trajectory tracking for 3 waypoints with 11 obstacles. The plots reveal that both trajectories reached at goal position without colliding with obstacles and traverse through all waypoints. But during the test, a trajectory with encoder and compass feedback has more drift and rover ends up 2 m apart from the actual goal position. While in case of GPS and compass feedback, the rover has communication delay and GPS data drop. In this assiduous condition, rover reached to the actual goal position without colliding with obstacles. But trajectory had more jerky behaviour as seen from angular velocity plots Figs. 6.33 and 6.34.

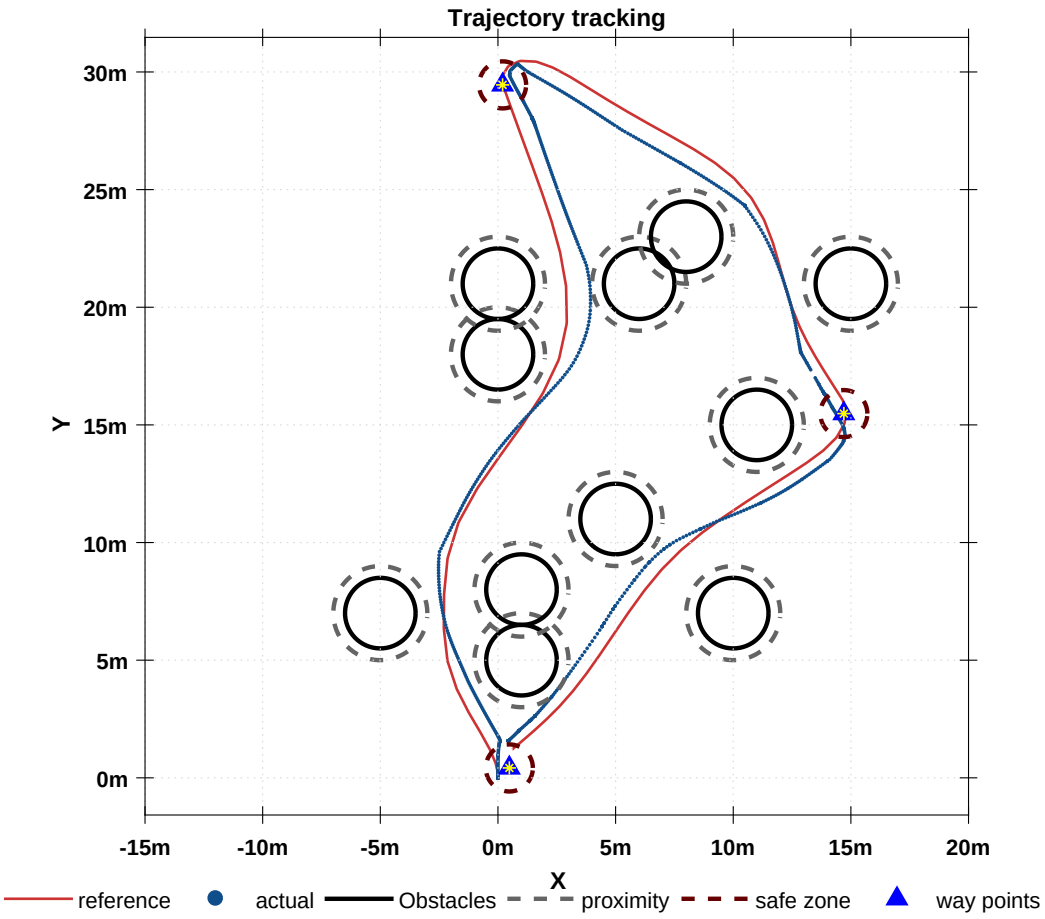

Figure 6.24: Real time trajectory tracking using Encoder and Compass for special case

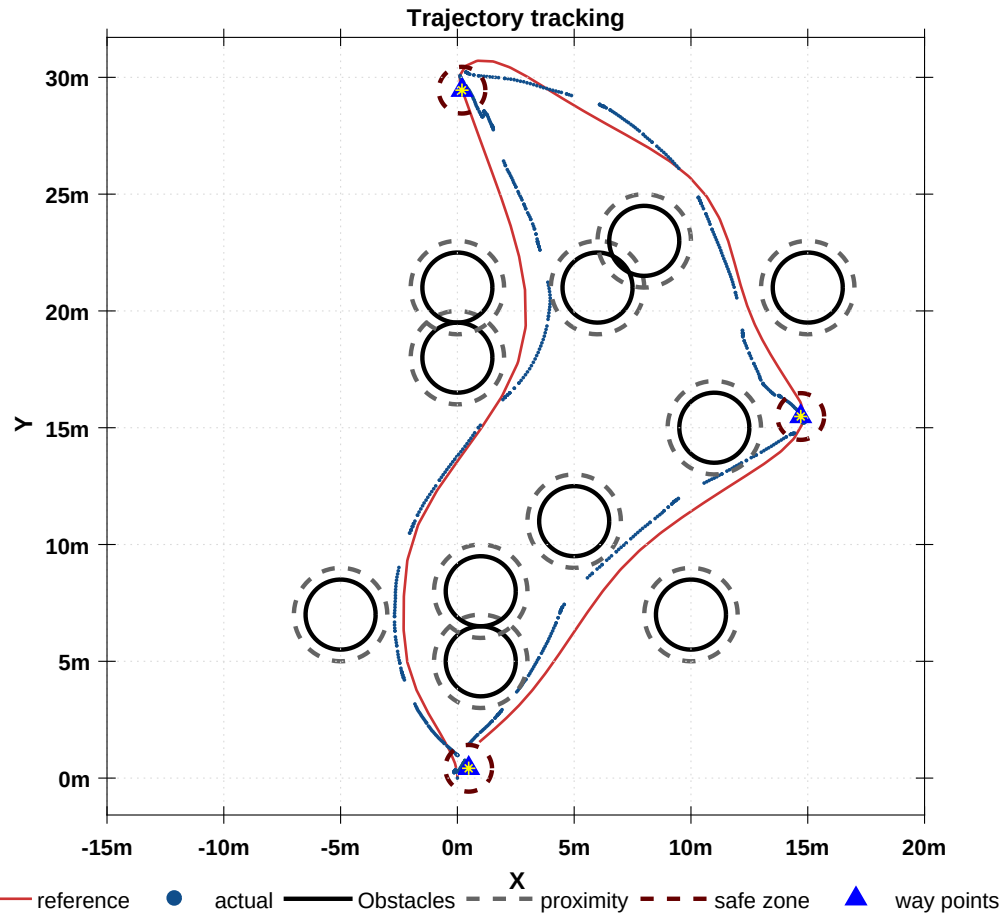

Figure 6.25: Real time trajectory tracking using Encoder and Compass for special case

Fig. 6.26 shows that trajectory tracking with encoder-compass feedback has more trajectory tracking error then GPS-Compass feedback, while less jerk during the maneuver. Error in trajectory tracking in case of encoder compass feedback, due to drift in encoder reading. Contrary in case of GPS-Compass, an error is due to GPS jump and communication delay.

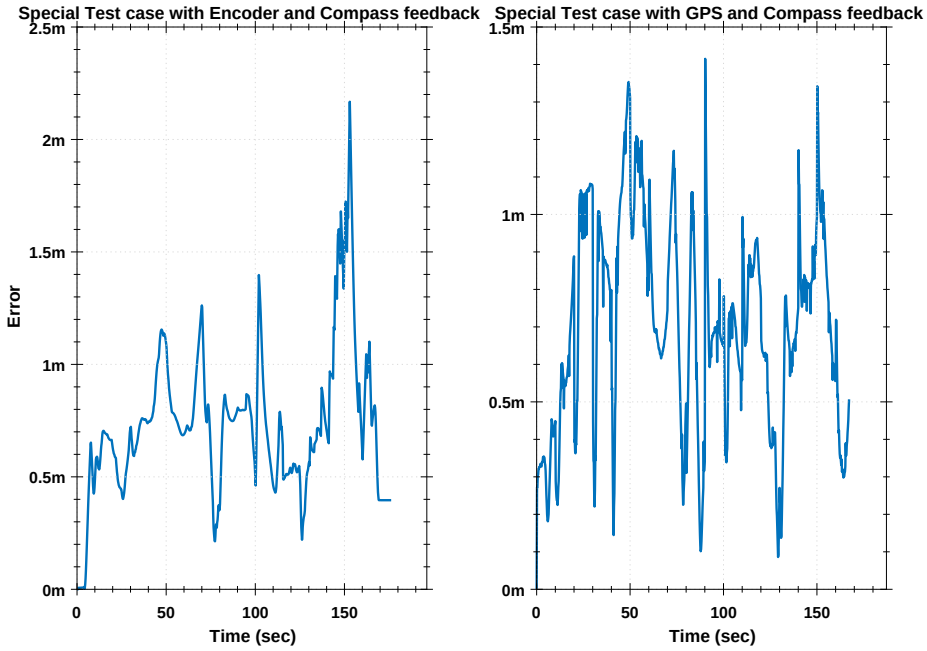

Figure 6.26: Error in position between reference and actual trajectory of the WMR

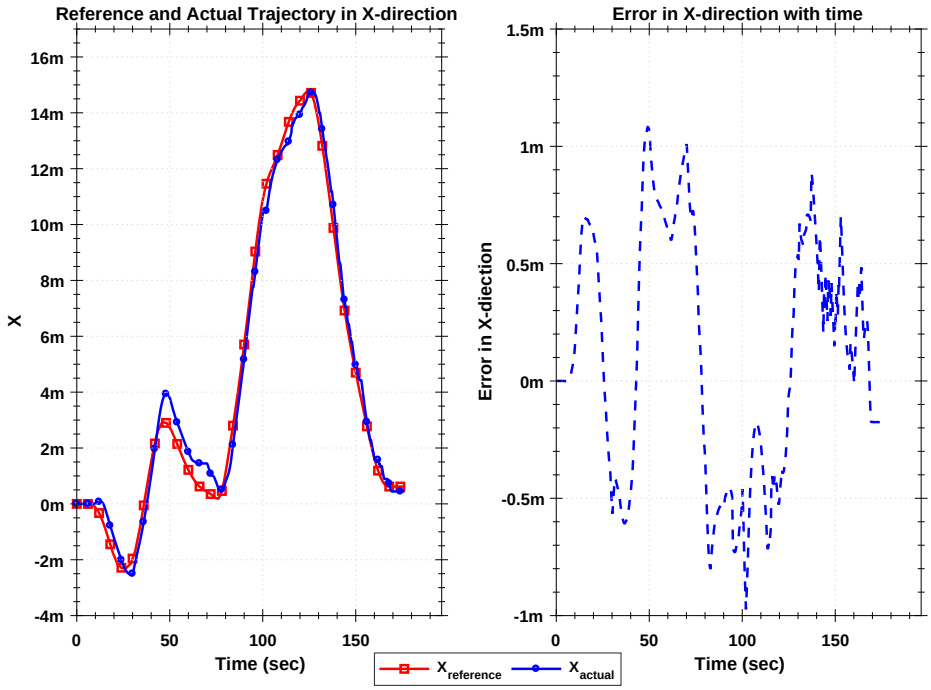

Figure 6.27: Trajectory tracking in X - direction using Encoder

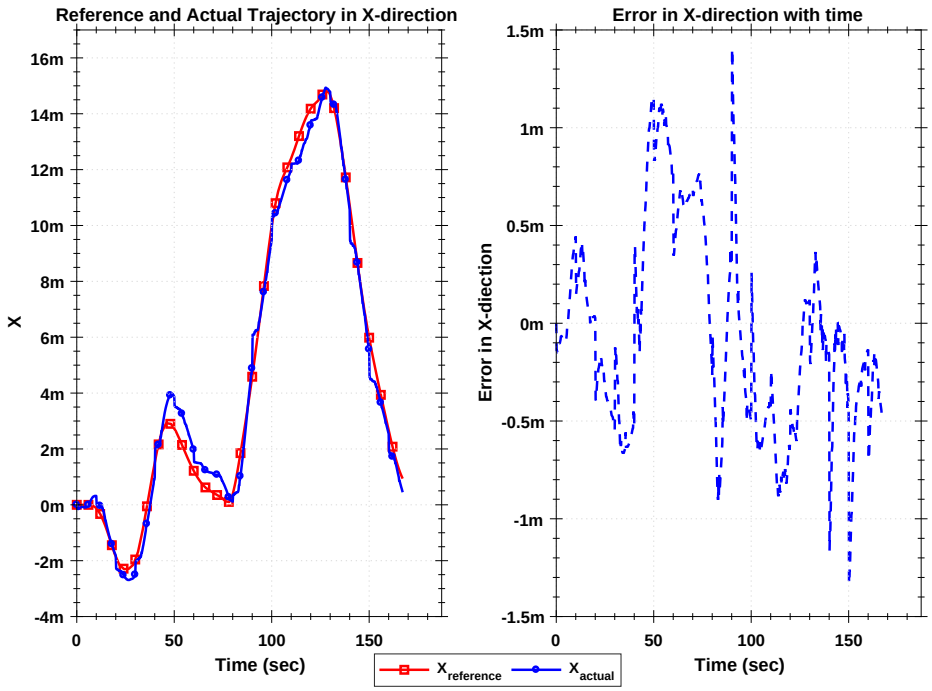

Figure 6.28: Trajectory tracking in X - direction using Encoder for test case 2

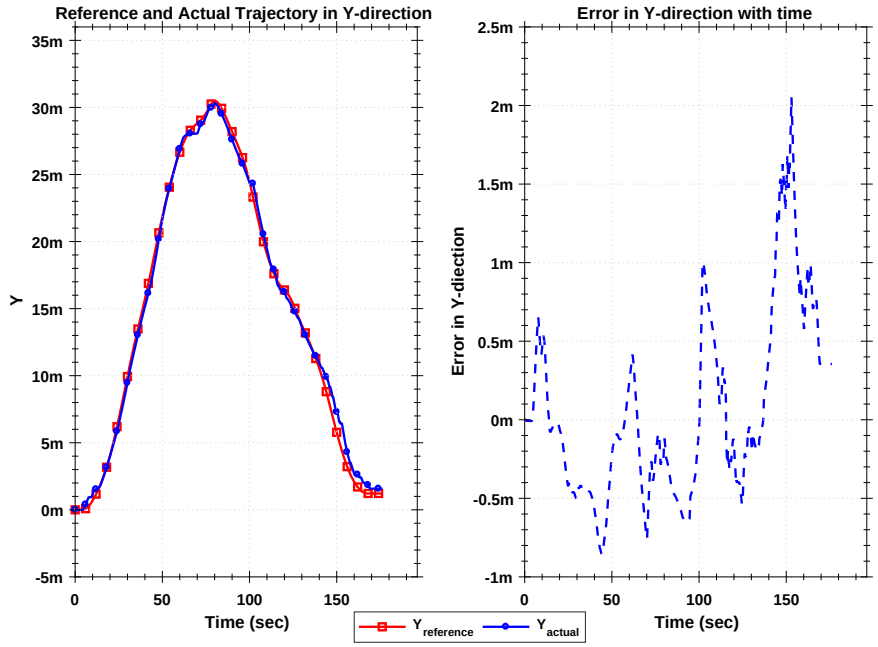

Figure 6.29: Trajectory tracking in Y - direction using Encoder for test case 2

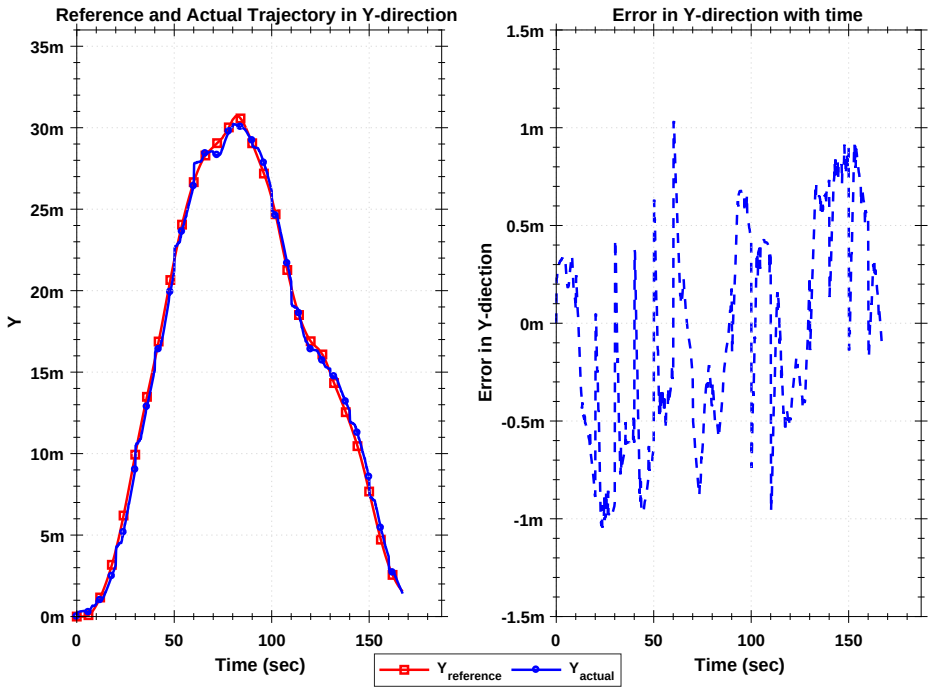

Figure 6.30: Trajectory tracking in Y - direction using Encoder for special case

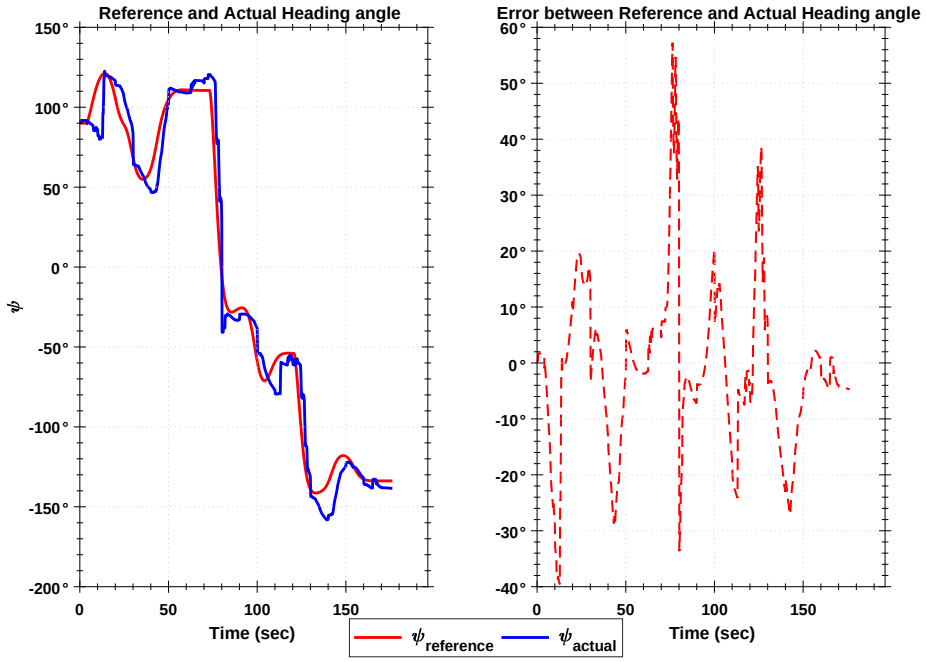

Figure 6.31: Heading angle using Encoder and Compass for special case

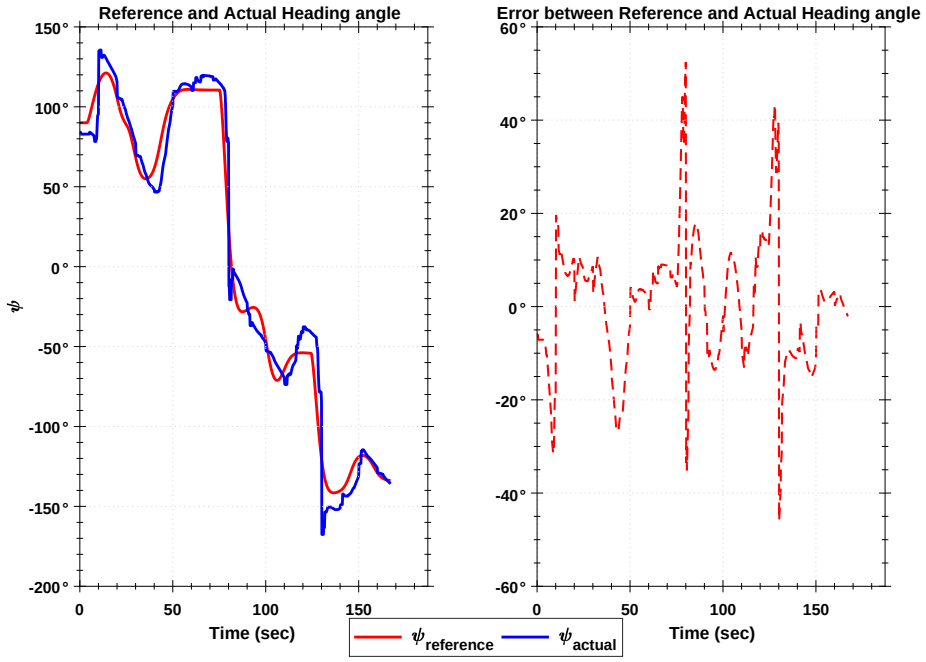

Figure 6.32: Heading angle using GPS and Compass

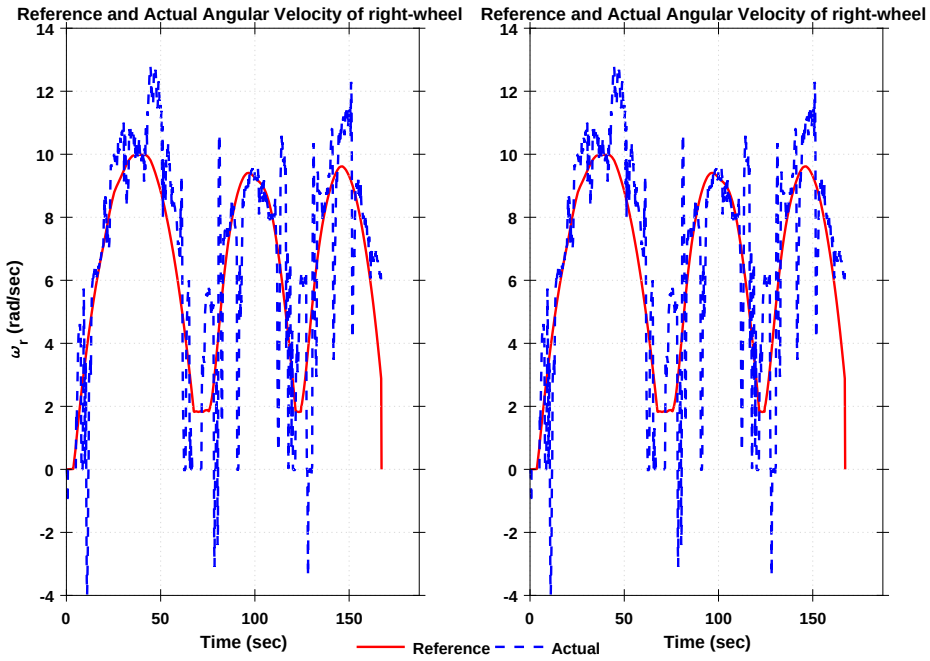

Figure 6.33: Angular acceleration of left wheel using Encoder for special case

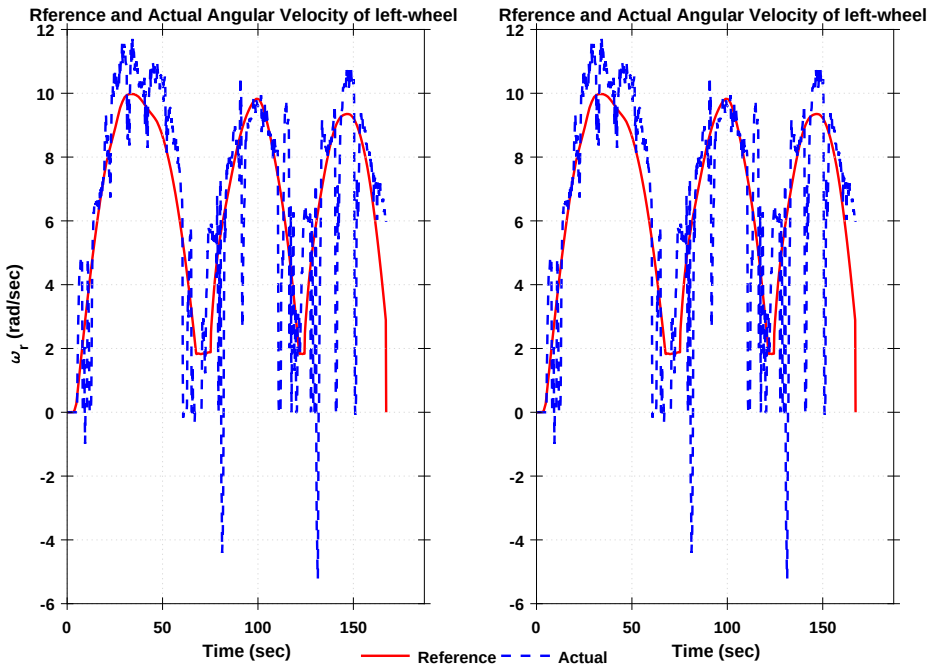

Figure 6.34: Angular acceleration of left wheel using Encoder for special case

6.5 Summary

This section illustrates that the desired reference trajectory generated using Pseudo-Spectral methods can be successfully implemented using cyber-physical system architecture and nonlinear control law. Different test cases shows the robustness of controller to track the trajectory. As there is communication delay and GPS signal drop during the test sometime WMR not able to maintain proximity distance from the obstacles as shown in special case.

CHAPTER 7

Summary, Conclusions and Future Work

7.1 Summary and Conclusions

The purpose of this research was to develop an optimal trajectory framework that is suitable to create desired path for unmanned ground vehicles. The developed scheme was extended to track the desired path in real-time using a cyber-physical system architecture and a nonlinear guidance law.

Pseudo-spectral collocation method was found to be robust enough to generate the optimal trajectory which accommodates constraint (i.e. virtual obstacles) by satisfying performance objectives (i.e go to goal, minimum final time, smooth heading change in angle and minimum actuator rate). In order to satisfy all performance objectives, some weights are added to objective functions. Weight tuning becomes sensitive in complicated scenario. By enforcing constraints on the linear and angular acceleration, we can reduce the amount of gain tuning and also satisfy the performance objective. The desired trajectory generated using the trajectory optimization satisfies the performance objectives and is implementable.

The reference trajectory is implemented in real-time on the WMR using a cyber-physical system architecture. The backstepping controller uses the reference trajectory and state feedback (GPS and Compass or Encoder and Compass) provides commanded inputs to reduce the deviation in trajectory tracking.

7.2 Future Work

- The next step is to extend this work by generating collision free trajectories for cooperative mission handling using multiple ground vehicles.
- Faster real-time trajectory generation using orthogonal polynomials and plan for dynamic obstacles.

APPENDIX A

Acronym

CFD	Computational Fluid Dynamics
CPS	Cyber-Physical System
DVR	Discrete variable Representation
ECEF	Earth Centred Earth Fixed
ENU	East North Up
GPS	Global Positioning System
IMU	Inertial Measurement Unit
IP	Interior Point
LG	Lagrange Gauss
LGR	Lagrange Gauss Radau
LGL	Lagrange Gauss Lobatto
NED	North East Down
NLP	Nonlinear Programming
OCP	Optimal Control Problem
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PS	Pseudo Spectral
PWM	Pulse Width Modulation
ROS	Robot Operating System
RRT	Rapidly-exploring Random Tree
SQP	Sequential Quadratic Programming
TPBVP	Two Point Boundary Value Problem
WMR	Wheeled Mobile Robot

REFERENCES

[1] Dowling, K., Guzikowski, R., Ladd, J., Pangels, H., Singh, S., and Whittaker, W. R. L., "NAVLAB: An Autonomous Navigation Testbed," Tech. Rep. CMU-RI-TR-87-24, Pittsburgh, PA, November 1987.

[2] Thorpe, C., Hebert, M., Kanade, T., and Shafer, S., "Toward Autonomous Driving: The CMU Navlab. Part II: System and Architecture," Vol. 6, No. 4, August 1991, pp. 44 – 52.

[3] Betts, J. T., "Survey of numerical methods for trajectory optimization," *Journal of Guidance control and dynamics*, Vol. 21, No. 2, 1998, pp. 193–207.

[4] Bryson, A. E., *Applied optimal control: optimization, estimation and control*, CRC Press, 1975.

[5] Naidu, D. S., *Optimal control systems*, CRC press, 2002.

[6] Shippey, B. M., *Trajectory optimization using collocation and evolutionary programming for constrained nonlinear dynamical systems*, Ph.D. thesis, The University of Texas at Arlington, 2008.

[7] Iyer, S. V., *Time optimal trajectory generation for a differential drive robot*, State University of New York at Buffalo, 2009.

[8] Von Stryk, D. M. O., "Numerical solution of optimal control problems by direct collocation," *Optimal Control*, Springer, 1993, pp. 129–143.

[9] Kelly, M., "An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation," *SIAM Review*, Vol. 59, No. 4, 2017, pp. 849–904.

[10] Atkeson, C. G., "Using local trajectory optimizers to speed up global optimization in dynamic programming," *Advances in neural information processing systems*, 1994, pp. 663–670.

[11] Garg, D., Patterson, M. A., Hager, W. W., Rao, A. V., Benson, D. A., and Huntington, G. T., "An overview of three pseudospectral methods for the numerical solution of optimal control problems," *Advances in the Astronautical Sciences*, Vol. 135, No. 1, 2009, pp. 475–487.

[12] Vera Rendón, S., "Trajectory planning based on collocation methods for multiple aerial and ground autonomous vehicles," 2015.

[13] Stentz, A., "Optimal and efficient path planning for partially-known environments," *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, IEEE, 1994, pp. 3310–3317.

[14] Likhachev, M., Gordon, G. J., and Thrun, S., "ARA*: Anytime A* with provable bounds on sub-optimality," *Advances in Neural Information Processing Systems*, 2004, pp. 767–774.

[15] Paden, B., Čáp, M., Yong, S. Z., Yershov, D., and Frazzoli, E., "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, Vol. 1, No. 1, 2016, pp. 33–55.

[16] LaValle, S. M., Yakey, J. H., and Kavraki, L. E., "A probabilistic roadmap approach for systems with closed kinematic chains," *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, Vol. 3, IEEE, 1999, pp. 1671–1676.

[17] Mathew, R. and Hiremath, S. S., "Trajectory Tracking and Control of Differential Drive Robot for Predefined Regular Geometrical Path," *Procedia Technology*, Vol. 25, 2016, pp. 1273–1280.

[18] Webb, D. J. and van den Berg, J., "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics," *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, IEEE, 2013, pp. 5054–5061.

[19] Perez, A., Platt, R., Konidaris, G., Kaelbling, L., and Lozano-Perez, T., "LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics," *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, IEEE, 2012, pp. 2537–2542.

[20] Otte, M. and Frazzoli, E., "{\mathrm {RRTˆ{X}}}: Real-Time Motion Planning/Replanning for Environments with Unpredictable Obstacles," *Algorithmic Foundations of Robotics XI*, Springer, 2015, pp. 461–478.

[21] Rao, A. V., "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, Vol. 135, No. 1, 2009, pp. 497–528.

[22] Bellman, R., "Dynamic programming and a new formalism in the calculus of variations," *Proceedings of the national academy of sciences*, Vol. 40, No. 4, 1954, pp. 231–235.

[23] Bellman, R., *Dynamic programming*, Courier Corporation, 2013.

[24] P. E. Gill, W. Murray, M. A. S. and Wright, M. H., "Users Guide for NPSOL (Version 4.0): A FORTRAN Package for Nonlinear Programmin," *Department of Operations Research, Stanford University*, 1986.

[25] P. E. Gill, W. M. and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, 2002.

[26] Betts, J. T., *Practical methods for optimal control and estimation using nonlinear programming*, SIAM, 2010.

[27] Reddien, G., "Collocation at Gauss points as a discretization in optimal control," *SIAM Journal on Control and Optimization*, Vol. 17, No. 2, 1979, pp. 298–306.

[28] Elnagar, G., Kazemi, M. A., and Razzaghi, M., "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE transactions on Automatic Control*, Vol. 40, No. 10, 1995, pp. 1793–1796.

[29] Elnagar, G. N. and Kazemi, M. A., "Pseudospectral Chebyshev optimal control of constrained nonlinear dynamical systems," *Computational Optimization and Applications*, Vol. 11, No. 2, 1998, pp. 195–217.

[30] Vlassenbroeck, J. and Van Dooren, R., "A Chebyshev technique for solving nonlinear optimal control problems," *IEEE transactions on automatic control*, Vol. 33, No. 4, 1988, pp. 333–340.

[31] Vlassenbroeck, J., "A Chebyshev polynomial method for optimal control with state constraints," *Automatica*, Vol. 24, No. 4, 1988, pp. 499–506.

[32] Ross, I. M. and Karpenko, M., "A review of pseudospectral optimal control: From theory to flight," *Annual Reviews in Control*, Vol. 36, No. 2, 2012, pp. 182–197.

[33] Bertsekas, D., *Nonlinear Programming* $2^n d$ *Edition*, Athena Scientific, 1999.

[34] Godbole, A., VNV, M., Quillen, P., and Subbarao, K., "Optimal Trajectory Design and Control of a Planetary Exploration Rover," *AIAA SPACE*, 2017.

[35] VNV, M., Baman, D., and Subbarao, K., "Nonlinear Guidance Laws for Trajectory Tracking over a Mobile Communication Network applied to Unmanned Ground Vehicles," *ACC*, 2018(Submitted).

BIOGRAPHICAL STATEMENT

Denish K. Baman was born May 9, 1992 in Surat, India. He did his undergraduate studies in Mechanical Engineering at NMIMS University, India. As he intended to explore his mind in the field of dynamics and robotics, he earned his Bachelor of Technology degree after completing a project on Air muscle actuated Robotic Arm in May 2014. During 2014-2015, he taught an undergraduate Machine Design and Auto-mobile System course as an assistant professor at MSCET, India. He joined the University of Texas at Arlington in Spring 2016 to pursue a M.S. program in Mechanical Engineering.

He is currently working as a Research Assistant in the Aerospace System Laboratory (ASL) with Dr. Kamesh Subbarao on guidance and controls of unmanned ground vehicles. His area of interest include dynamics and control of multi-body systems.