Path Planning of Wheeled Mobile Robot in Static Environment

In partial fulfilment for the award of the degree

of

Bachelor of Technology

in

Mechanical Engineering

Submitted by:

Shubham Kumar – B18ME006

Mebanshem Lyngdoh – B18ME002

Ngangbam Mekey Singh – B18ME003

Jayaditya Shanmugan Tenampet – B18ME010

Under the supervision

of

Dr. Bikash Kumar Sarkar, Assistant Professor

&

Mr. Avilash Sahoo, Trainee Teacher

Department of Mechanical Engineering
National Institute of Technology Meghalaya
Bijni Complex, Laitumkhrah, Shillong-793003, Meghalaya.

राष्ट्रीय प्रौद्योगिकी संस्थान मेघालय

NATIONAL INSTITUTE OF TECHNOLOGY MEGHALAYA

Bijni Complex, Laitumkhrah, Shillong 793003

CERTIFICATE

This is to certify that this report, entitled ***"Path Planning Of Wheeled Mobile Robot in Static Environment"***, which has been submitted ***by Shubham Kumar (B18ME006), Mebanshem Lyngdoh (B18ME002), Ngangbam Mekey Singh (B18ME003), Jayaditya Shanmugan Tenampet (B18ME010)*** in partial fulfilment for the requirements for the award of ***Bachelor of Technology Degree in Mechanical Engineering*** at National Institute of Technology Meghalaya (Deemed University) is authentic and carried out under our supervision and guidance.

To the best of our knowledge, the contents in this report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Place: ***Shillong***

Dated:

Dr. Bikash Kumar Sarkar
Assistant Professor
Department of Mechanical Engineering
National Institute of Technology Meghalaya

Mr. Avilash Sahoo
Trainee Teacher
Department of Mechanical Engineering
National Institute of Technology Meghalaya

ACKNOWLEDGEMENT

It was a privilege and a valuable learning process to be able to contribute to a topic that is now a extensive study. The project work assigned is truly an educational enrichment, brought to fruition through the efforts of some very dedicated people.

We would like to express our sincere gratitude thanks to the supervisors of our project ***Dr. Bikash Kumar Sarkar*** & ***Mr. Avilash Sahoo,*** for their patience, invaluable suggestions, helpful information, practical advice and unceasing ideas which have helped us tremendously at all times during the project. Their guidance has helped us to learn a lot of new things regarding this topic. It was a great opportunity to work under such personalities who have always helped and showed us ways whenever much needed.

Last but not the least we would like to thank all our friends and peers for their constant support and motivation till the successful completion of this project.

Shubham Kumar **Mebanshem Lyngdoh**

Jayaditya Shanmugam Tenampet **Ngangbam Mekey Singh**

ABSTRACT

Nowadays mobile robot is used almost everywhere. It is used in a variety of applications such as delivering components between assembly stations in factories, delivering food and medication in hospitals, cleaning rooms, mowing lawns or agricultural tasks. Path Planning is an essential component of mobile robot navigation. In this paper we have formulated the equations for kinematics and dynamics of the mobile robot. The kinematic and dynamic modelling of wheeled mobile robot helps in determining the state of motion of the robot. For a given set of inputs the resultant path of the robot can be formulated with the help of equations of Jacobian which correlates the local and global parameters of the robot. The robot is then modelled using Solidworks to get an approximate of its mass distribution and centre of mass and moment of inertia. A Simulink model is then created using the various equations formulated earlier and the parameters found in the CAD model is incorporated in the model. Forward modelling of the mobile robot takes the input of the wheel torques and gives the output in terms of linear and rotational velocities and position coordinates of the robot. Inverse modelling of the mobile robot takes the input in terms of path generated using path planning algorithm and gives the output in terms of wheel torques and forces. Both the forward and kinematic models are then combined to get the error between them. A PID controller is then used in Simulink models to tune the results and reduce the error between the models.

CONTENT

Chapter	Chapter name	Page No.
1	**Introduction**	7
1.1	Introduction	7
1.2	Types of robot	7
1.3	Types of wheels in mobile robot	8
1.4	Differential Drive Robot	9
2	**Literature Review and Objective**	10
2.1	Literature review	10
2.2	Objective	11
3	**Mathematical Modelling of Mobile Robot**	12
3.1	Kinematic Modelling	12
3.2	MATLAB Simulation of Kinematic model	17
3.3	Dynamics of wheeled mobile robot	23
4	**CAD Modelling**	26
5	**Results and Discussions**	29
5.1	Forward Modelling of Mobile robot in Simulink	29
5.2	Inverse Modelling of Mobile Robot in Simulink	32
5.3	Forward and Inverse Combined Model for Checking Error	35
6	**Conclusions and Future Scope**	38
6.1	Conclusions	38
6.2	Future Scope	38
	References	

Table of figures

Fig No	Caption	Page No
1.1	Fixed wheel	8
1.2	Orientable wheel	8
1.3	Ball wheel	8
1.4	Omni wheel	8
1.5	Mecanum wheel	8
1.6	Differential Drive robot	9
3.1	Diagram showing global and local frame	12
3.2	Kinematics of wheel	13
3.3	Vector diagram of wheel kinematics	14
3.4	Vector diagram showing relation between wheel and body frame	15
3.5	Kinematics of differential mobile robot	16
3.6	Plot of robot path between x and y axes	18
3.7	Robot motion in y axis with respect to time	19
3.8	Robot motion in x axis with respect to time	19
3.9	Path plotted for w1=0.8 rad/s and w2=0.6 rad/s	21
3.10	Path plotted for w1=0.6 rad/s and w2=0.6 rad/s	22
3.11	Path plotted for w1=0.8 rad/s and w2=-0.6 rad/s	22
3.12	Diagram showing centroid and centre of gravity for robot	24
4.1	Image of the physical robot RM100	26
4.2	Isometric view of robot CAD model	26
4.3	Front view of robot CAD model	27

4.4	Bottom view of robot CAD model	27
4.5	Combined view of robot CAD model	27
5.1	Forward model of robot in Simulink	29
5.2	x-y plot generated for case1	30
5.3	x-y plot generated for case2	30
5.4	x-y plot generated for case3	31
5.5	x-y plot generated for case4	31
5.6	Inverse model of robot in Simulink	32
5.7	Subsystem in inverse model to find out ψ	32
5.8	Position plot X vs Y	33
5.9	Plot of local velocities with respect to time	33
5.10	Plot of Fx, Fy and Mz with respect to time	34
5.11	Plot of tw1 and tw2 with respect to time	34
5.12	Combined Simulink model without PID controller	35
5.13	Desired position vs actual position in model without PID controller	35
5.14	Simulink block of PID controller	36
5.15	Combined Simulink model with PID controller	37
5.16	Desired position vs actual position in model with PID controller	37

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Mobile robots are suitable for a multitude of applications such as delivering components between assembly stations in factories, delivering food and medication in hospitals, cleaning rooms, mowing lawns or agricultural tasks, as a result mobile robots control has been a very active research fields for many years. Out of all mobile robots the most important are the wheeled mobile robots, due to their simplicity and robustness. Wheeled mobile robots are cheaper to build than other roots and can be used in dangerous environments, for example nuclear waste facilities and mines. All these applications require precise control. Precise control can be achieved using mathematical modelling and trajectory tracking algorithm, which has been studied by several researchers.

Kinematics is the study of the mechanical systems behaviour without considering the forces that affect the motion. In mobile robotics, it is necessary to familiar with the mechanical behaviour of the robot both in order to design suitable mobile robots for tasks and to understand how to create control software for an instance of mobile robot hardware.

Dynamics is the study of systems that undergo changes of states as time. In mechanical systems such as robots, the change of states involves motion.in other words, dynamics is the science of motion. It describes why and how a motion occurs when forces and motions are applied on massive bodies. This motion can be considered as evolution of the position, orientation, and their time derivatives.

Path Planning is an important task for autonomous mobile robots. The main requirement of the optimal and accurate path planning is the accessibility of environment and odometric information. This depends upon how accurate the environmental and odometric information is attained by the robot. Another requirement in path planning is that robot must reach the destination without colliding with obstacles and path must be shortest. The shortest path is considered as the time saving constraint for mobile robots and the safest path is considered as the safety constraint for mobile robots.

1.2 TYPES OF ROBOTIC SYSTEMS

There are various types of robots used in today’s world. All the robots can be broadly classified into three different categories of robotic system:-

- Manipulation Robotic System – They are the most common type of robot used in the manufacturing industries. They consist of robots with manipulator arms with varying degree of freedom. They can be used to perform various tasks such as drilling, welding, material handling, packing and other such tasks.
- Mobile Robotic System – They consist of robots which move from one place to another. They can be used in transporting equipment, carrying products, traversing in risky environment. They are the next generation of robotic system which is also used in the autonomous movement of the robot.
- Data Acquisition and Control Robotic System – They are used to gather, process and transmit data for a variety of signals. They are incorporated in the mobile robotic system to get the data required in self-controlling robots.

1.3 TYPES OF WHEELS IN MOBILE ROBOT

The wheels used in mobile robots are broadly classified as conventional and non-conventional wheels.

Conventional wheels:

- Fixed wheels- Fixed Wheels have two degrees of freedom and can move a robot backward and forward. These wheels are commonly used as drive wheels on a rear-wheel or four-wheel drive robot.
- Orientale Wheels- they are generally used to stabilize and robot, and some of them allow for a tighter turning radius since rotate around their Z-axis.
- Ball wheel/Castor- These wheels are made up of either a metal or nylon ball mounted in a holder. The ball has 360-Degrees of freedom and they are usually just used to balance a robot.

Figure 1.1 : Fixed wheel

Figure 1.2 : Orientable wheel

Figure 1.3 : Ball wheel

Non-conventional wheels:

- Omni wheel-These wheels are made up of multiple roller wheels mounted around their circumference. The rollers on Omni wheels allow them to move forward, backward, and sideways.

- Mecanum wheels- they are a type of Omni wheel that have rollers attached to the centre spoke at a 45-degree angle. This configuration allows for faster dynamic motion in every direction. This is especially true for lateral motion since the rollers on Mecanum wheels are designed to quickly and efficiently transfer forward wheel spin into sideways motion.

Figure1.4 : Omni wheel

Figure 1.5 : Mecanum wheel

1.4 DIFFERENTIAL DRIVE ROBOTS

Differential Drive robot typically has two powered wheels, one on each side of the robot. Sometimes there are other passive wheels that keep the robot from tipping over.

- When both wheels turn at the same speed in the same direction, the robot moves straight in that direction.
- When one wheel turns faster than the other, the robot turns in an arc toward the slower wheel.
- When the wheels turn in opposite directions, the robot turns in place

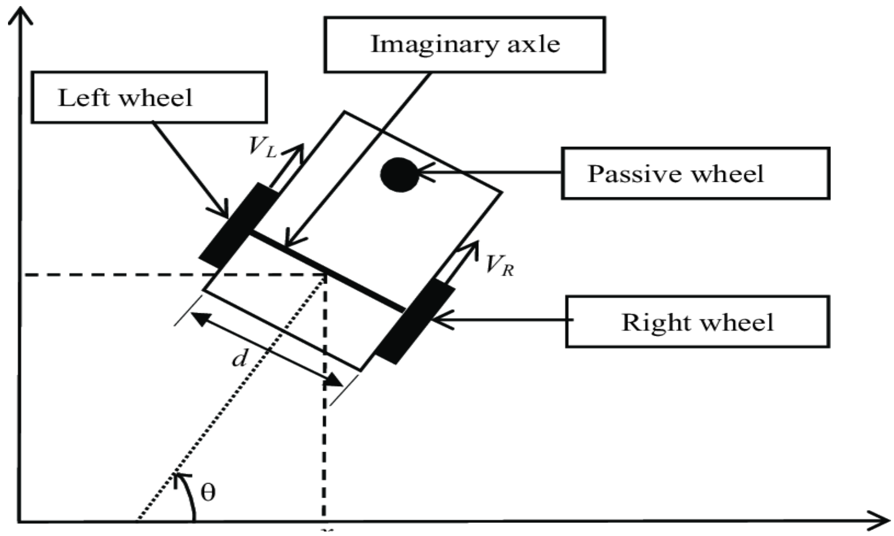

Figure 1.6 : Differential drive robot (Source : Google Images)

CHAPTER 2

LITERATURE REVIEW AND OBJECTIVE

2.1 LITERATURE REVIEW

The control of mobile robots' local guidance is a vital issue. Robots must advance towards the target while avoiding any obstacles. If there are multiple obstacles in the area, the robot must recognise them and avoid colliding with them. The distance and direction of the goal and obstacles, as well as the angular breadth and linear velocity of impediments, the time of contact, and the linear and angular velocity and acceleration of the robots, are all important aspects in robot navigation. To measure distance to obstacles and targets, most robots employ SONAR or Laser rangefinders.

There are several research papers written for the kinematic and dynamic modelling of the wheeled mobile robot. Zainuri et al. [1] researched on the inverse kinematics of the wheeled robot. The robot's orientation toward a lined track was watched in order to determine the correct location by managing the speed disparities between the two-wheeled rears individually. The angular speed of the rear wheels is calculated based on the robot's desired linear speed, and an error occurs in the angle of the robot's orientation, which is influenced by the radius of the rear wheel, the distance between the two driven rear wheels, and the curvature of the mobile robot's turn movement. Tang and Pei [2] have also researched on differential flatness based kinematic and dynamic control of differential driven robot. Petrovic et al[3] did the dynamic modelling of robot based on physical modelling and on experimental identification of robot dynamic features. Simple kinematic odometry models are valid if the mobile robot is moving at a slow speed with little acceleration. Parhi et al. [4] researched on kinematic analysis of wheeled mobile robot. A wheeled mobile robot was defined as a planer rigid body with an arbitrary number of wheels. It was demonstrated that five sorts of categories may be created for a vast class of conceivable wheel configurations. A kinematical model was constructed based on the geometrical restrictions of these wheels, and the degree of mobility, steerability, and manoeuvrability was investigated. Finally, numerous three-wheeled mobile robots are subjected to this study.

Lena et al. [5] have researched on the modelling and trajectory control of wheeled mobile robot by taking into account the kinematics, actuator, dynamics and rolling resistance of the wheels. Various trajectories that are similar to real-life events were developed, and the model and control algorithm were observed to provide accurate trajectory tracking. Filipescu et al. [6] researched on discrete time sliding mode control of wheeled mobile robots. The trajectory tracking problem of wheeled mobile robots is addressed in this study using discrete-time sliding mode control. The discrete-time sliding mode controller is a variable structure controller that performs measurements and control signal applications at regular intervals of time while maintaining a steady control signal. Raja et al. [7] researched on optimal path planning of mobile robots. The research gave an overview of various optimisation algorithms for path planning of mobile robots

2.2 OBJECTIVE

Path planning of mobile robots is a very important aspect in the guidance of robots in a crowded environment. The main objective of our project is to find a suitable method for guidance of robots and avoidance of unexpected obstacles in the path of robot from its initial position to the target position.

- The first goal is to develop the mathematical equations of the differential wheeled mobile robot so as to get the understanding of the kinematics of mobile robot.
- The second objective is to generate MATLAB code to simulate the kinematic equations.
- The third objective is to develop the dynamic equations of the mobile robot
- The fourth objective is to develop the Simulink model using both the kinematic and dynamic equations of the wheeled mobile robot.
- The fifth objective is to compare the forward and inverse Simulink models to get the error between the desired path and actual path.
- The final objective is to use the PID controller in the combined model and tune it to reduce the error

CHAPTER 3
MATHEMATICAL MODELLING

The first step in the path planning of mobile robot is the mathematical modelling of the robot. This includes kinematic and dynamic modelling of the mobile robot where various equations of body motion and dynamics are used to get a final equation that can define the parameters of the mobile robot and predict its movement and loads applied.

3.1 KINEMATIC MODELLING OF WHEELED ROBOT

Consider a wheeled mobile robot that has 3 degrees of freedom- 2 translational and 1 rotational.

Consider a global Cartesian frame at I

And local Cartesian frame at B which is at distance of (x,y) from I

Assuming x is the forward displacement of mobile robot with respect to I
y is the lateral displacement of mobile robot with respect to I
ψ is the angular displacement of mobile robot with respect to I.

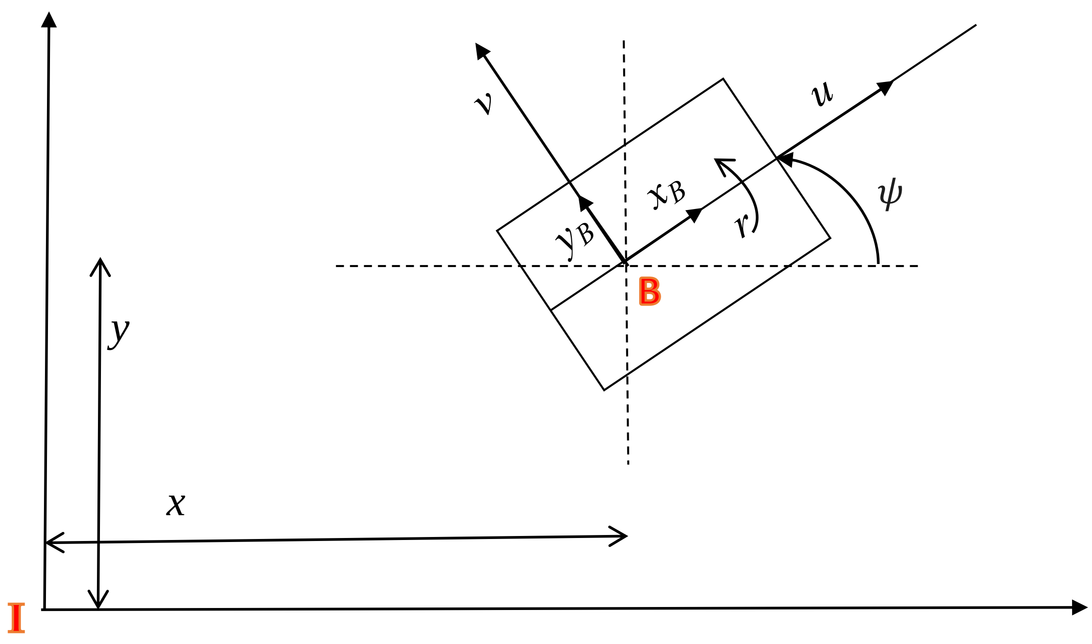

Figure 3.1 : Diagram showing the global frame and body frame

u is the forward velocity with respect to B

v is the lateral velocity with respect to B

r is the angular velocity with respect to B

From the above given diagram, the velocity vector of the mobile robot in matrix form can be given as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} u \ \cos\psi - v\sin\psi \\ u \ \sin\psi + v\cos\psi \\ r \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix}$$

In simplified manner it is written as:

$$\dot{\eta} = J(\psi)\,\zeta$$

The above equation is called Jacobian matrix. This equation describes the relation between velocity input command ζ and derivative of generalised coordinated η.

<u>FORWARD KINEMATICS</u> –

Forward kinematics refers to the use of kinematic equations of the robot to compute the end position of robot from specified values. In simple words, the velocities of robot is given and the system motion is computed

<u>INVERSE KINEMATICS-</u>

It refers to the computation of the various motion parameters of the mobile robot on the basis of motion. In simple words, for a given position trajectory, the linear and angular velocity are computed.

It is the reverse of forward kinematic equation given by:

$\zeta = J^{-1}(\psi)\,\eta$.

Kinematics of wheel

Consider a is the radius of the wheel

ω is angular velocity of the wheel

C_i is the centre of the wheel

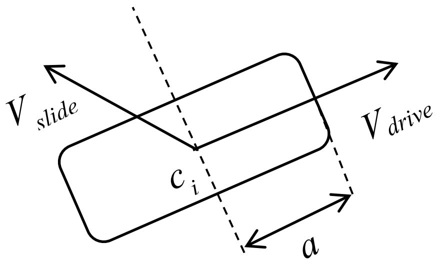

Figure 3.2 : Kinematics of wheel

There are two velocities which act on the wheel – driving velocity and sliding velocity

$V_{drive} = \omega_i a_i$

$V_{slide} = \beta_i \rho_i$ where β_i is the rotational velocity of passive rollers

ρ_i is the radius of passive

Let $\dot{x}_{ci}$ and $\dot{y}_{ci}$ be the forward and lateral velocity respectively

and ϕ_i is the angle between roller axis and vertical axis of wheel

Now considering the vector diagram we have

$\dot{y}_{ci} = \rho_i \beta_i \cos \phi_i$

$\Rightarrow \rho_i \beta_i = \dot{y}_{ci} / \cos \phi_i$

Similarly, $\dot{x}_{ci} = \omega_i a_i - \rho_i \beta_i \sin \phi_i$

$\Rightarrow \dot{x}_{ci} = \omega_i a_i - \dot{y}_{ci} \tan \phi_i$

$\Rightarrow \omega_i a_i = \dot{x}_{ci} + \dot{y}_{ci} \tan \phi_i$

$\Rightarrow \omega_i = 1/a_i (\dot{x}_{ci} + \dot{y}_{ci} \tan \phi_i)$

$\Rightarrow \omega_i = \begin{bmatrix} \frac{1}{a_i} & \frac{1}{a_i} \tan \phi_i \end{bmatrix} \begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{bmatrix}$

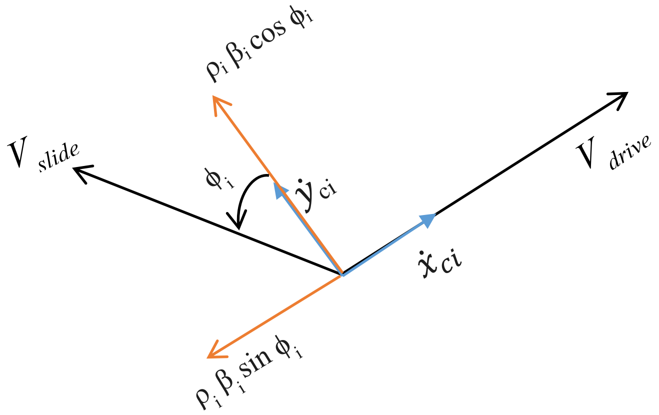

Figure 3.3: Vector diagram of wheel kinematics

Now considering ${}^{ci}v_{ci}$ to be the absolute velocity of the wheel and ${}^{B}v_{ci}$ to be the velocity of wheel with respect to body frame B

And, θ_i is the angle between wheel frame and body frame

So we have,

$${}^{ci}v_{ci} = \begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{bmatrix}$$

$${}^{B}v_{ci} = {}^{B}_{ci}R(\theta_i) \begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{bmatrix}$$

$$= \begin{bmatrix} \dot{x}_{ci} cos\theta_{Bi} - \dot{y}_{ci} sin\theta_{Bi} \\ \dot{x}_{ci} sin\theta_{Bi} + \dot{y}_{ci} cos\theta_{Bi} \end{bmatrix}$$

$$= \begin{bmatrix} cos\theta_{Bi} & -sin\theta_{Bi} \\ sin\theta_{Bi} & cos\theta_{Bi} \end{bmatrix} \begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{bmatrix}$$

Suppose c_i frame is at distance dx_i and dy_i from B in x and y direction and

B moves with u, v linear and lateral velocities respectively and rotational velocity.

So,

u-rdy_i = $\dot{x}_{ci}cos\theta_{Bi} - \dot{y}_{ci}sin\theta_{Bi}$

v+rdx_i =$\dot{x}_{ci}sin\theta_{Bi} + \dot{y}_{ci}cos\theta_{Bi}$

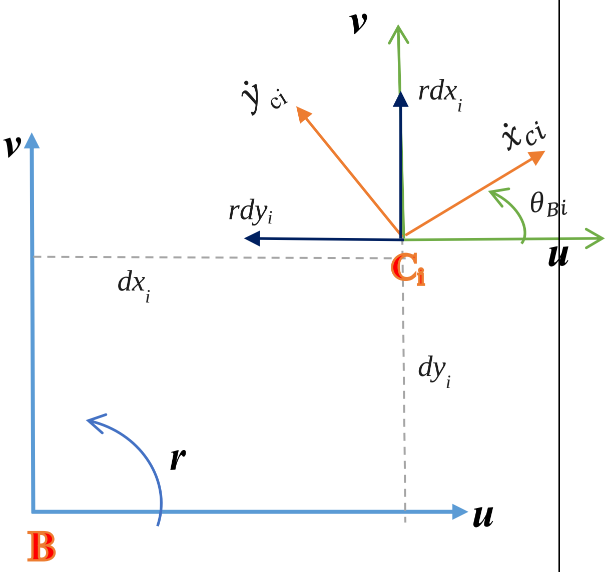

Figure 3.4: Vector diagram showing relation between wheel and body frame

So, ${}^{B}v_{ci}$ == $\begin{bmatrix} 1 & 0 & -\mathrm{dy}_i \\ 0 & 1 & dx_i \end{bmatrix}\begin{bmatrix} u \\ v \\ r \end{bmatrix}$

$$\Rightarrow \begin{bmatrix} cos\theta_{Bi} & -sin\theta_{Bi} \\ sin\theta_{Bi} & cos\theta_{Bi} \end{bmatrix}\begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\mathrm{dy}_i \\ 0 & 1 & dx_i \end{bmatrix}\begin{bmatrix} u \\ v \\ r \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \dot{x}_{ci} \\ \dot{y}_{ci} \end{bmatrix} = \begin{bmatrix} cos\theta_{Bi} & sin\theta_{Bi} \\ -sin\theta_{Bi} & cos\theta_{Bi} \end{bmatrix}\begin{bmatrix} 1 & 0 & -\mathrm{dy}_i \\ 0 & 1 & dx_i \end{bmatrix}\begin{bmatrix} u \\ v \\ r \end{bmatrix}$$

Substituting this in wheel motion equation we have,

ω_i

$$=\begin{bmatrix} \frac{1}{a_i} & \frac{1}{a_i}\tan\phi_i \end{bmatrix}\begin{bmatrix} cos\theta_{Bi} & sin\theta_{Bi} \\ -sin\theta_{Bi} & cos\theta_{Bi} \end{bmatrix}\begin{bmatrix} 1 & 0 & -\mathrm{dy}_i \\ 0 & 1 & dx_i \end{bmatrix}\begin{bmatrix} u \\ v \\ r \end{bmatrix}$$

where ω_i is the angular velocity of the wheel

a_i is the radius of i_{th} wheel

θ_{Bi} is the angle between vehicle frame(B) and wheel frame (c_i)

dx_i and dy_i are position of c_i with respect to B

ϕ_i is the angle between roller axis to x axis

u is the forward velocity of mobile robot with respect to B

v is the lateral velocity of mobile robot with respect to B

r is the angular velocity of mobile robot with respect to B

Now, for the two wheel differential drive robot, there are no passive rollers so ϕ_1 and $\phi_2 = 0$. Also the centre of wheel is considered to be in alignment to the centre of the body frame.

i.e., dy_1 = d, dx_1 =0, a_1 =a and, dy_2 = -d, dx_2 =0, a_2 =a

and $\theta_{B1}, \theta_{B2} = 0$

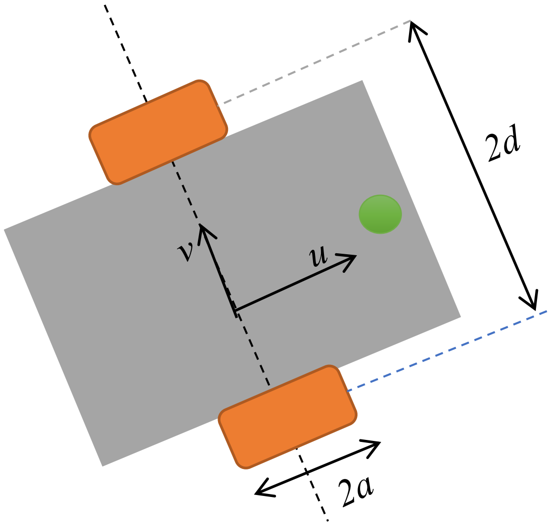

Figure 3.5: Kinematics of differential mobile robot

Putting the values of dy, dx and a in equation of wheel motion we get,

$$\omega_1 = \begin{bmatrix} \frac{1}{a} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & -d \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} = \frac{1}{a}(u - rd)$$

$$\omega_2 = \begin{bmatrix} \frac{1}{a} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & d \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} = \frac{1}{a}(u + rd)$$

so, $\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} 1/a & -d/a \\ 1/a & d/a \end{bmatrix} \begin{bmatrix} u \\ r \end{bmatrix}$

or, $\begin{bmatrix} u \\ r \end{bmatrix} = \begin{bmatrix} a/2 & a/2 \\ -a/2d & a/2d \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}$

or, $\begin{bmatrix} u \\ v \\ r \end{bmatrix} = \begin{bmatrix} a/2 & a/2 \\ 0 & 0 \\ -a/2d & a/2d \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}$

In simplified term, $\zeta = W\omega$

The above equation of wheel motion can be used in the kinematic equation of the mobile robot to get the final inverse kinematic equation:-

From eq 1, $\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix}$

Or, $\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a/2 & a/2 \\ 0 & 0 \\ -a/2d & a/2d \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}$

Therefore, $\begin{bmatrix}\omega_1\\ \omega_2\end{bmatrix} = \begin{bmatrix} a/_2 & a/_2 \\ 0 & 0 \\ -a/_{2d} & a/_{2d}\end{bmatrix}^{-1} \begin{bmatrix}\cos\psi & -\sin\psi & 0\\ \sin\psi & \cos\psi & 0\\ 0 & 0 & 1\end{bmatrix}^{-1}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\psi}\end{bmatrix}$

This is the final mathematical relation to find out the angular velocities of the wheels, if all other parameters are provided.

3.2 MATLAB SIMULATION OF THE KINEMATIC MODEL

The above given matrix relation is incorporated in the MATLAB environment to get an understanding of the movement of the mobile robot in real time.

Below is the matlab code that describes the relation between velocity input command ζ and derivative of generalised coordinated η.

```
%% mobile robot
clear all; clc; close all;

%%simulation parameter
dt = 0.1 %step size
ts = 10 %simulation time
t = 0:dt:ts; %time span

%%initial conditions
x0 = 0;
y0 = 0;
psi0 = pi/2;

eta0 = [x0;y0;psi0];

eta(:,1) = eta0;

%%loop starts here
for i = 1:length(t)
    psi = eta(3,i);
    J_psi = [cos(psi),-sin(psi),0;
             sin(psi),cos(psi),0;
             0,0,1];
    %%inputs
    u = 0.1;  %x axis vel
    v = 0.05;    %y axis vel
    r = 0.3;    %angular vel

    zeta(:,i) = [u;
                 v;
                 r];



    %%time derivative of generalized corrdinates

    eta_dot(:,i) = J_psi * zeta(:,i);

    %% position propogation using euler method
```

```
    eta(:,i+1) = eta(:,i) + dt* eta_dot(:,i);

end

%% plotting functions starts here

figure
plot(t, eta(1,1:i), 'r-');
set(gca,'fontsize',14')
xlabel('t,[s]');
ylabel('x,[m]');

figure
plot(t, eta(1,1:i), 'b-');
set(gca,'fontsize',14')
xlabel('t,[s]');
ylabel('y,[m]');

figure
plot(t, eta(1,1:i), 'g-');
set(gca,'fontsize',14')
xlabel('t,[s]');
ylabel('psi,[rad]');

%% animaion

l = 0.6; %length
w = 0.4 %width

mr_co = [-l/2,l/2,l/2,-l/2,-l/2;
          -w/2,-w/2,w/2,w/2,-w/2;];
figure
for i = 1:length(t)
    psi = eta(3,i);
    R_psi = [cos(psi),-sin(psi);
        sin(psi), cos(psi);];
    v_pos = R_psi*mr_co;
    fill(v_pos(1,:)+eta(1,i),v_pos(2,:)+eta(2,i),'g')
    hold on, grid on
    axis([-1 3 -1 3]), axis square
    plot(eta(1,1:i),eta(2,1:i),'b-');
    legend('MR','Path')
    set(gca,'fontsize',24)
    xlabel('x,[m]'); ylabel('y,[m]');
    pause(0.1)
    hold off
end
```

To test the code we give the following inputs

```
%%initial conditions
x0 = 0;
y0 = 0;
psi0 = pi/2;
```

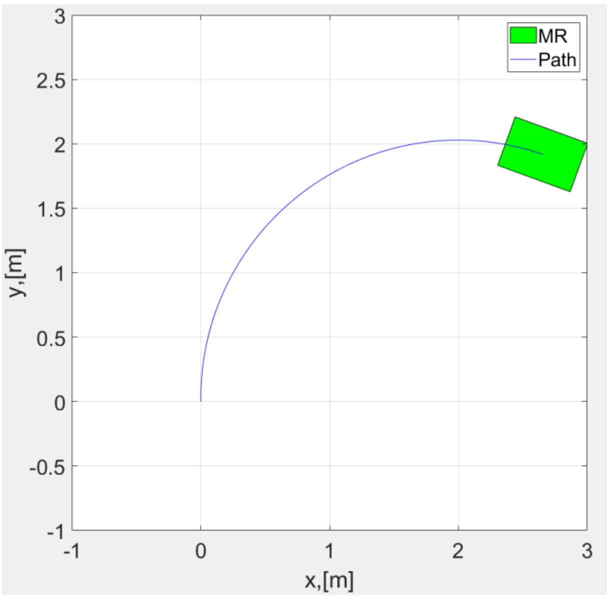

Figure 3.6: x vs y plot wrt time

```
%%inputs
u = 0.6;  %x axis vel
v = 0.0;    %y axis vel
r = - 0.3;    %angular vel
```

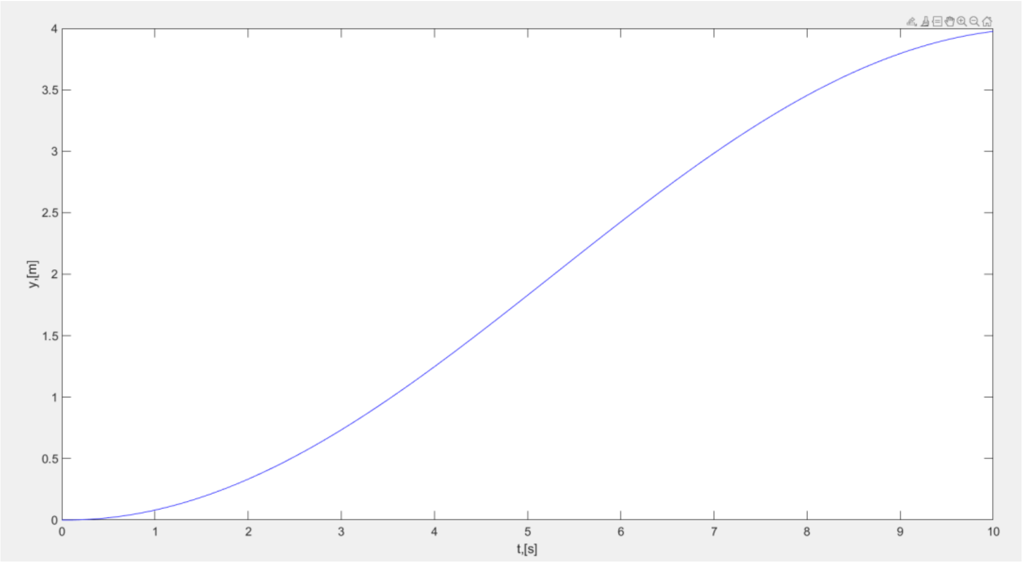

Figure 3.7: Robot motion in y-axis wrt time

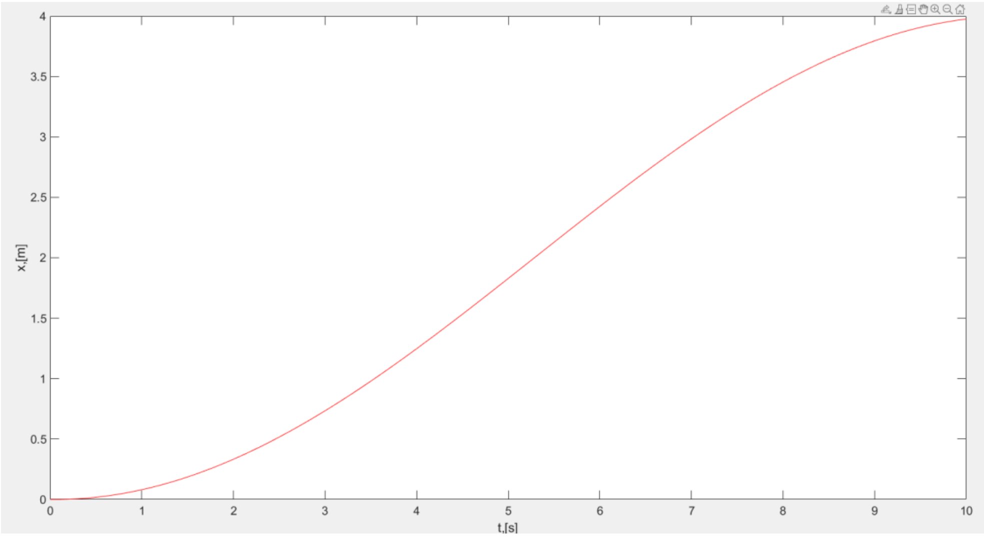

Figure 3.8: Robot motion in x-axis wrt time

Below is the matlab code that describes the relation between wheel angular velocity matrix ω and derivative of generalised coordinated η.

```
%% mobile robot
clear all; clc; close all;

%%simulation parameter
dt = 0.1 %step size
ts = 100 %simulation time
t = 0:dt:ts; %time span

%%mobile robot parameters
a = 0.05; %radius of wheel
d = 0.2; %dist bw wheel and vehicle(along y axis)

%%initial conditions
x0 = 0;
y0 = 0;
psi0 = pi/2;

eta0 = [x0;y0;psi0];

eta(:,1) = eta0;

%%loop starts here
for i = 1:length(t)
    psi = eta(3,i);
    J_psi = [cos(psi),-sin(psi),0;
             sin(psi),cos(psi),0;
             0,0,1];
    %%inputs
    omega_1 = 0.8; %left wheel anugular velocity
    omega_2 = 0.6; %right wheel anugular velocity
    omega = [omega_1;omega_2];

    %% wheel config matrix
    W = [a/2,a/2;
        0,0
        -a/(2*d), a/(2*d);]
    % velocity input commands
    zeta(:,i) = W*omega;

    %%time derivative of generalized corrdinates

    eta_dot(:,i) = J_psi * zeta(:,i);

    %% position propogation using euler method

    eta(:,i+1) = eta(:,i) + dt* eta_dot(:,i);

end

%% plotting functions starts here

figure
plot(t, eta(1,1:i), 'r-');
set(gca,'fontsize',14')
xlabel('t,[s]');
ylabel('x,[m]');
```

```
figure
plot(t, eta(1,1:i), 'b-');
set(gca,'fontsize',14')
xlabel('t,[s]');
ylabel('y,[m]');

figure
plot(t, eta(1,1:i), 'g-');
set(gca,'fontsize',14')
xlabel('t,[s]');
ylabel('psi,[rad]');

%% animaion

l = 0.75; %length
w = 2*d; %width

mr_co = [-l/2,l/2,l/2,-l/2,-l/2;
          -w/2,-w/2,w/2,w/2,-w/2;];
figure
for i = 1:5:length(t)
    psi = eta(3,i);
    R_psi = [cos(psi),-sin(psi);
        sin(psi), cos(psi);];
    v_pos = R_psi*mr_co;
    fill(v_pos(1,:)+eta(1,i),v_pos(2,:)+eta(2,i),'g')
    hold on, grid on
    axis([-1 3 -1 3]), axis square
    plot(eta(1,1:i),eta(2,1:i),'b-');
    legend('MR','Path')
    set(gca,'fontsize',24)
    xlabel('x,[m]'); ylabel('y,[m]');
    pause(0.01)
    hold off
end
```

To test the code we give the following inputs

```
%%inputs

omega_1 = 0.8; %left wheel anugular velocity
omega_2 = 0.6; %right wheel anugular velocity
```

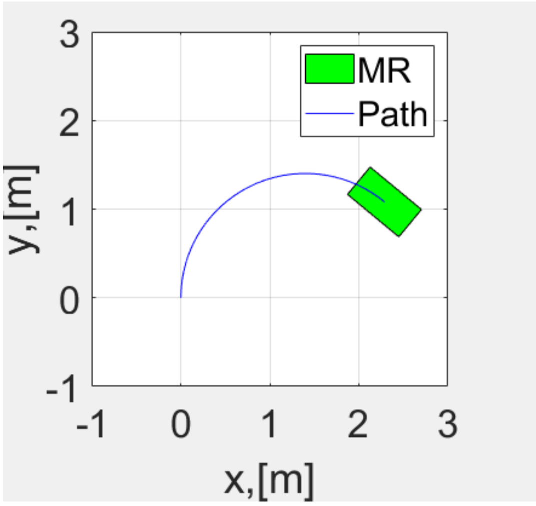

Figure 3.9: Path plotted for w1=0.8rad/s and w2=0.6rad/s

```
%%inputs
omega_1 = 0.6; %left wheel anugular velocity
omega_2 = 0.6; %right wheel anugular velocity
```

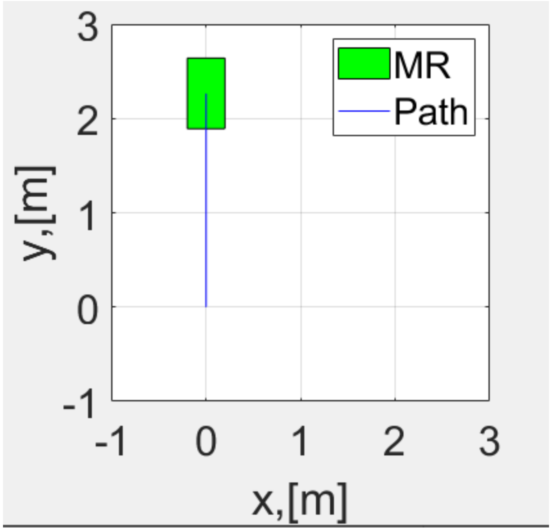

Figure 3.10: Path plotted for w1=0.6rad/s and w2=0.6rad/s

```
%%inputs
omega_1 = 0.6; %left wheel anugular velocity
omega_2 = -0.6; %right wheel anugular velocity
```

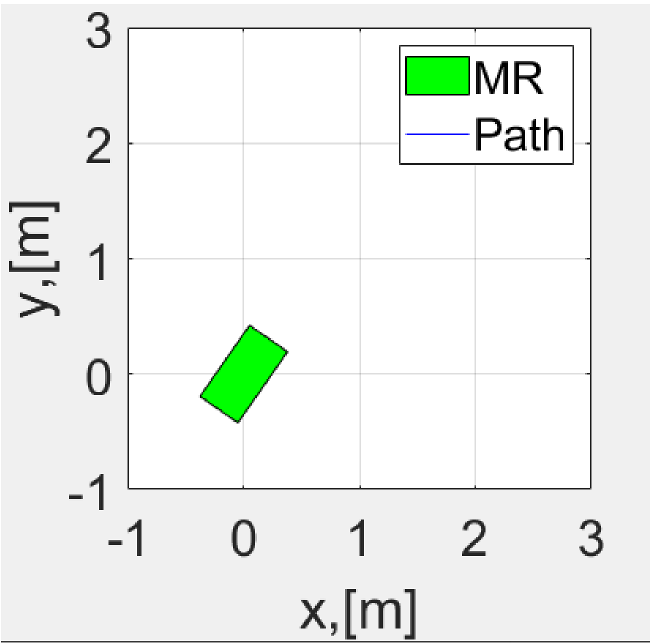

Figure 3.11: Path plotted for w1=0.6rad/s and w2=-0.6rad/s

3.3 DYNAMICS OF WHEELED MOBILE ROBOT

Dynamics is the study of systems that undergo changes of states as time. In mechanical systems such as robots, the change of states involves motion.in other words, dynamics is the science of motion. It describes why and how a motion occurs when forces and motions are applied on massive bodies. This motion can be considered as evolution of the position, orientation, and their time derivatives.

EQUATION OF MOTION-

The equation describes the way in which motion of robotic system responds to the torques/forces applied by the actuators, or from external forces/moments applied to the system.

It is generally represented as:

$$\tau = f(\eta, \dot{\eta}, \ddot{\eta})$$

Where τ represents the generalised matrices of the forces acting on the body

$\eta, \dot{\eta}, \ddot{\eta}$ represents the trajectory coordinates and its 1st and 2nd derivatives respectively.

FORWARD DYNAMICS-

Forward Dynamics uses joint torques/forces to predict resultant motions. In other words, if you know the torques/forces of an articulated body, then Forward Dynamics helps you predict the orientations of each link in the articulated body.

In simple terms, for a given input τ, the motion parameters i.e., $\eta, \dot{\eta}, \ddot{\eta}$ is calculated.

INVERSE DYNAMICS-

Inverse dynamics is a method for computing forces and/or moments of force (torques) based on the kinematics (motion) of a body and the body's inertial properties (mass and moment of inertia).

In simple terms, for a given desired trajectory $(\eta, \dot{\eta}, \ddot{\eta})$, the forces/torques τ are computed.

LAGRANGE EULER METHOD –

The dynamic motion equation derived using Lagrange-Euler formulation is:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial v}\right) - \frac{\partial L}{\partial x} = F$$

Where L stands for the difference in kinematic and potential energy, *v* stands for the velocity of the mobile robot

x stands for the position coordinates of mobile robot

and F stands for generalised force acting on the system.

Now, considering a wheeled mobile robot whose centre of gravity is not aligned with the centroid.

B is the centroid of body

C is the centre of gravity

x_{bc} is the horizontal distance between B and C

y_{bc} is the vertical distance between B and C

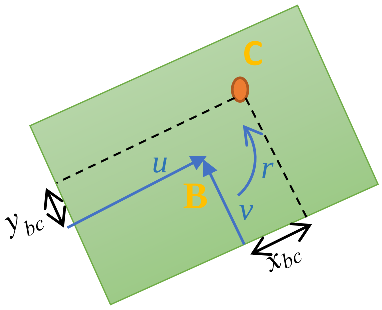

Figure 3.12: Diagram showing centroid and centre of gravity for robot

The kinetic energy is due to the translational velocity and rotational velocity and also the potential energy is 0 as the robot is moving on a flat surface.

K.E.=$\frac{1}{2}m(\dot{x}_c^2 + \dot{y}_c^2\ ) + \frac{1}{2}I_z r^2$

P.E.=0

$\dot{x}_c = u - y_{bc}r$

$y_c = v + x_{bc}r$

So, L=K.E. – P.E.

=$\frac{1}{2}m(\dot{x}_c^2 + \dot{y}_c^2\ ) + \frac{1}{2}I_z r^2$

=$\frac{1}{2}m[u^2 - 2ury_{bc} + v^2 + 2vrx_{bc} + r^2(x_{bc}^2 + y_{bc}^2) + \frac{1}{2}I_z r^2]$

Now,

$\frac{\partial L}{\partial u} = mu - mry_{bc}$

$\frac{\partial L}{\partial v} = mv + mrx_{bc}$

$\frac{\partial L}{\partial r} = -muy_{bc} + mvx_{bc} + mr[x_{bc}^2 + y_{bc}^2] + I_z r$

Therefore,

$F_x = \frac{d}{dt}\left(\frac{\partial L}{\partial u}\right) = m\dot{u} - m\dot{r}y_{bc} - mr\dot{y}_{bc}$

$F_y = \frac{d}{dt}\left(\frac{\partial L}{\partial v}\right) = m\dot{v} + m\dot{r}x_{bc} + mr\dot{x}_{bc}$

$M = \frac{d}{dt}\left(\frac{\partial L}{\partial r}\right) = -m\dot{u}y_{bc} - mu\dot{y}_{bc} + mv\dot{y}_{bc} + m\dot{v}y_{bc} + m\dot{r}[x_{bc}^2 + y_{bc}^2] +$

$2mr[2x_{bc}\dot{x}_{bc} + 2y_{bc}\dot{y}_{bc} + I_z r]$

Now, solving the value of $\dot{x}_{bc}$ and $\dot{y}_{bc}$ in the relation, and depicting the relation in matrix form, we have:

$$\begin{bmatrix} m\dot{u} - mvr - mx_{bc}r^2 - my_{bc}r^2 \\ m\dot{v} + mur - my_{bc}r^2 + mx_{bc}r^2 \\ \left(I_{cx} + m(x_{bc}^2 + y_{bc}^2)\right)\dot{r} + m(x_{bc}[\dot{v} + ur] - y_{bc}[\dot{u} - vr]) \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix}$$

$$\begin{bmatrix} m & 0 & -my_{bc} \\ 0 & m & mx_{bc} \\ -my_{bc} & mx_{bc} & I_{cx} + m(x_{bc}^2 + y_{bc}^2) \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} -mr(v + x_{bc}r) \\ mr(u + y_{bc}r) \\ mr(x_{bc}u + y_{bc}v) \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix}$$

$$\begin{bmatrix} m & 0 & -my_{bc} \\ 0 & m & mx_{bc} \\ -my_{bc} & mx_{bc} & I_{cx} + m(x_{bc}^2 + y_{bc}^2) \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{r} \end{bmatrix} + r \begin{bmatrix} 0 & -m & mx_{bc} \\ m & 0 & my_{bc} \\ mx_{bc} & my_{bc} & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ M_z \end{bmatrix}$$

In simple term it is expressed as:

$$M\dot{V} + CV = \tau$$

Where M is the inertial matrix,

$\dot{V}$ is the acceleration matrix,

CV is the function of velocity matrix,

and, τ is the force/moment matrix.

CHAPTER 4

CAD MODELLING OF MOBILE ROBOT

This CAD model was modelled in Solidworks 2020. An approximate measurement of the robot “Peer Robotics RM100” was taken physically including battery, wheels, frame, and position of castor wheels. The model was made with accurate materials to mimic the mass distribution of the actual robot. Below are the images attached of the actual robot and its CAD model.

Figure 4.1: Image of the physical robot RM100

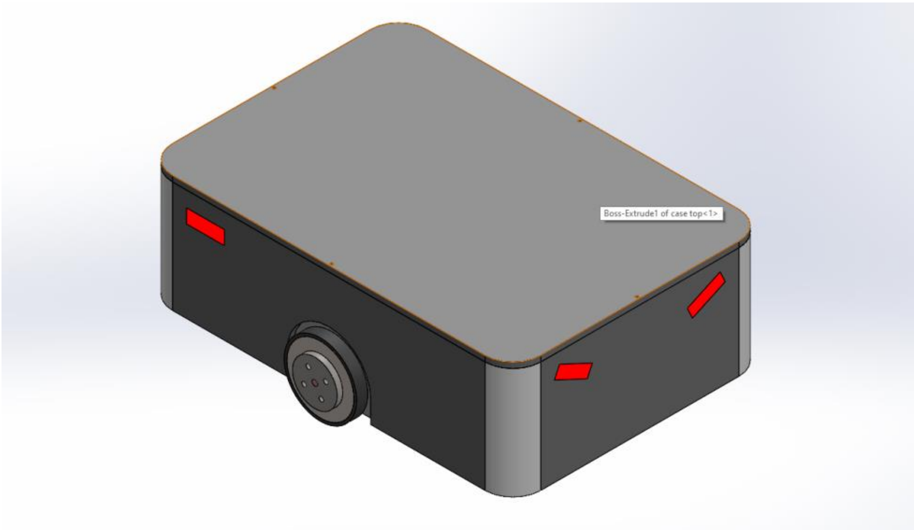

Figure 4.2: Isometric view of robot CAD model

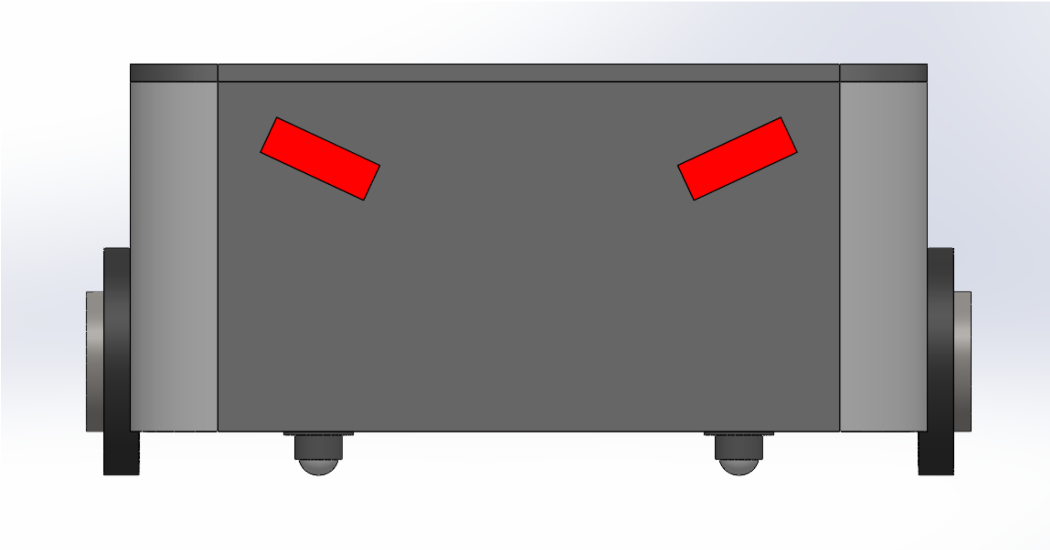

Figure 4.3: Front view of robot CAD model

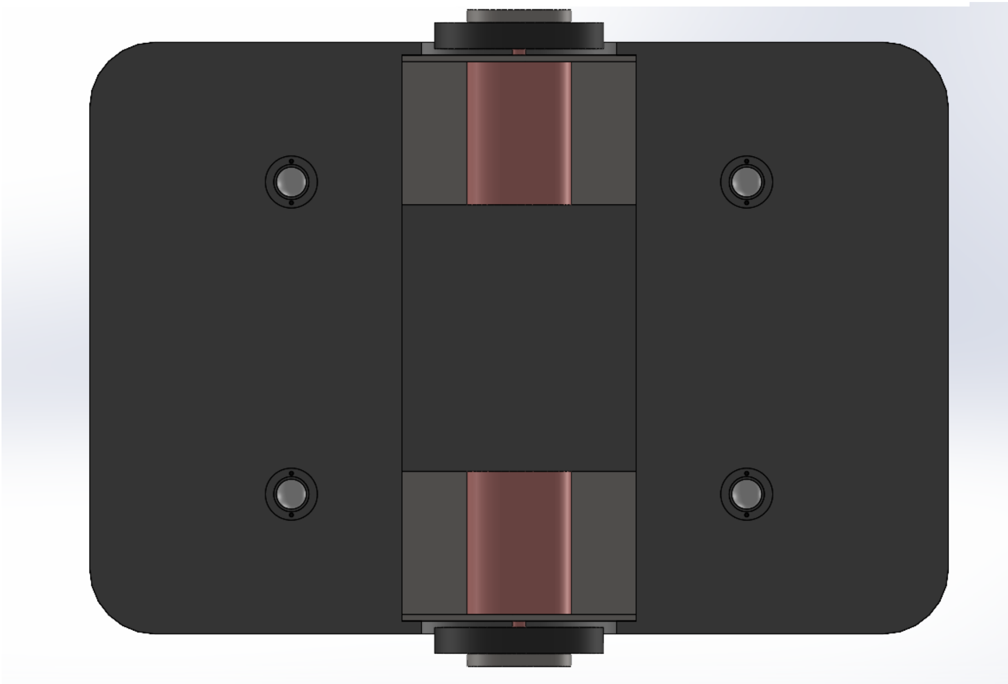

Figure 4.4: Bottom view of robot CAD model

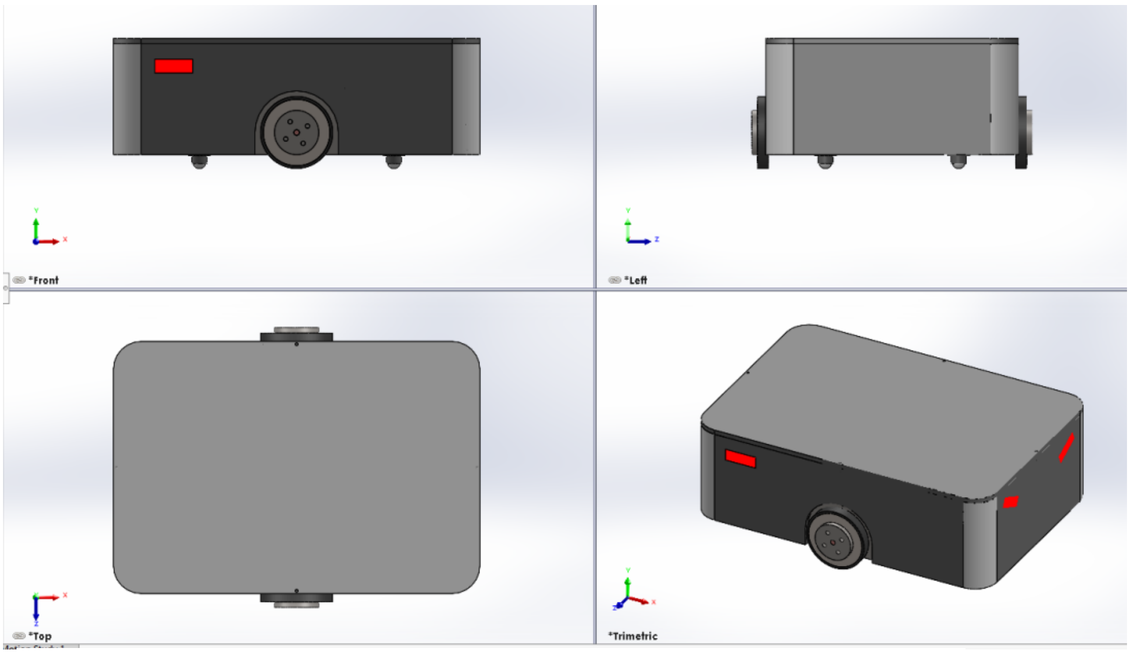

Figure 4.5: Combined view of the robot CAD model

After the CAD model was developed the parameters like weight, centre of mass, moment of inertia were found. The details of the robot parameters are given below.

Robot parameters:

Mass of the robot, m=26.2;

Length of the robot, l=0.66;

Breadth of the robot, b=0.455;

Height of the robot, h= 0.20

Radius of wheel, rw=0.065;

Distance of C.G. from centroid along x direction, xbc=0.01;

Distance of C.G. from centroid along y direction, ybc=0;

Moment of inertia about Z-axis, iz= 1.14;

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 FORWARD MODELLING OF MOBILE ROBOT IN SIMULINK

The input torques is given to both the wheels and is incorporated in the forward dynamics equation.

$$M\dot{V} + CV = \tau$$

$$M^{-1}\tau + M^{-1}CV = \dot{V}$$

The acceleration matrix, $\dot{V}$ is integrated to velocity matrix, V. The velocity matrix,V found is the local velocity of the robot. It is then multiplied with the required Jacobian matrix to get the global velocities(x_dot, y_dot, psi_dot).

$$\begin{bmatrix} u \\ v \\ r \end{bmatrix} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix}$$

The global velocities are then integrated to get the global position matrix.

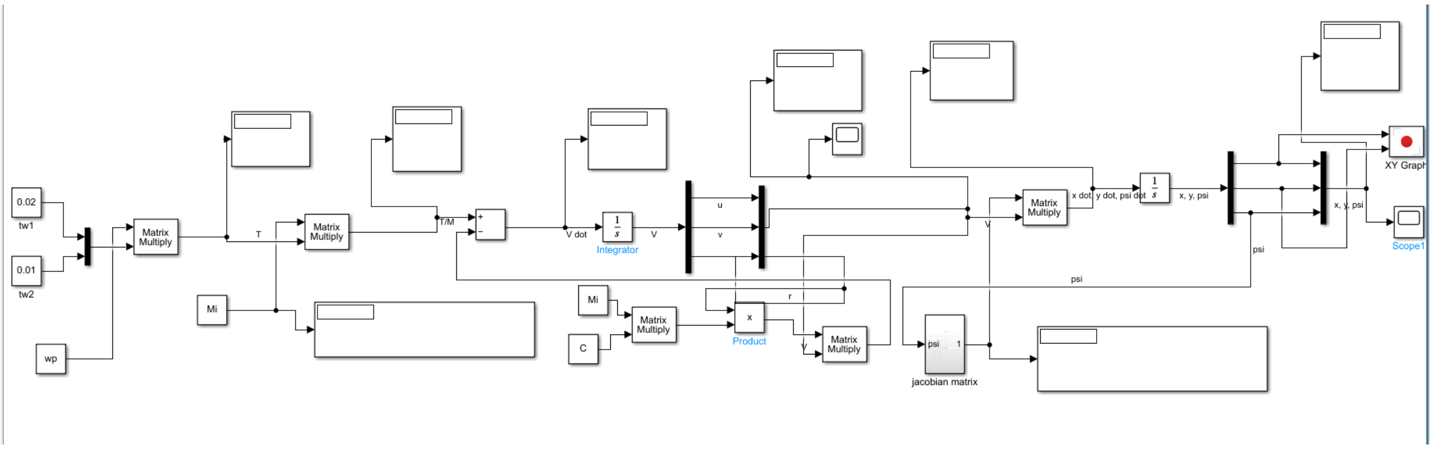

Figure 5.1: Forward model of the robot in Simulink

RUNNING THE SIMULATION AND RECORDING OUTPUTS

The Simulink model is run using various input wheel torques and the x vs. y plots generated are found. Here tw1 is the input torque of left wheel and tw2 is the input torque of right wheel.

- **Case 1**: tw1=0.01, tw2=0.01

Figure 5.2: x-y plot generated for case 1

- **Case 2** : tw1= 0.01, tw2= 0.012

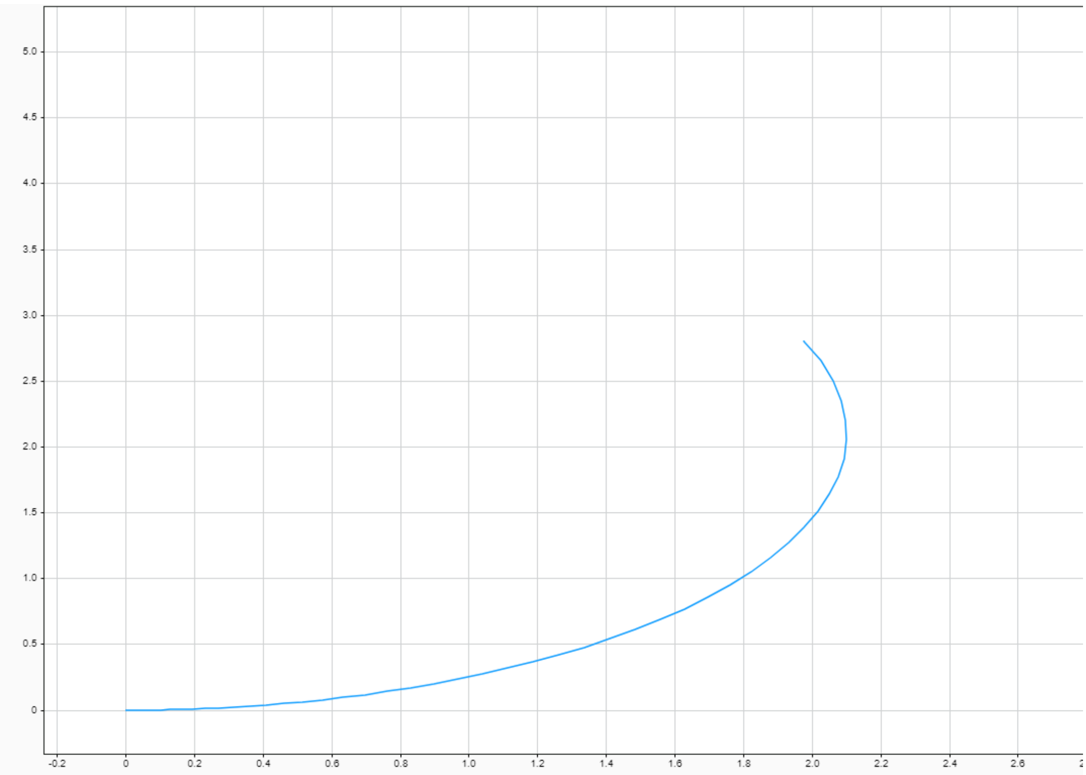

Figure 5.3: x-y plot generated for case 2

○ **Case 3** : tw1= 0.01, tw2= 0

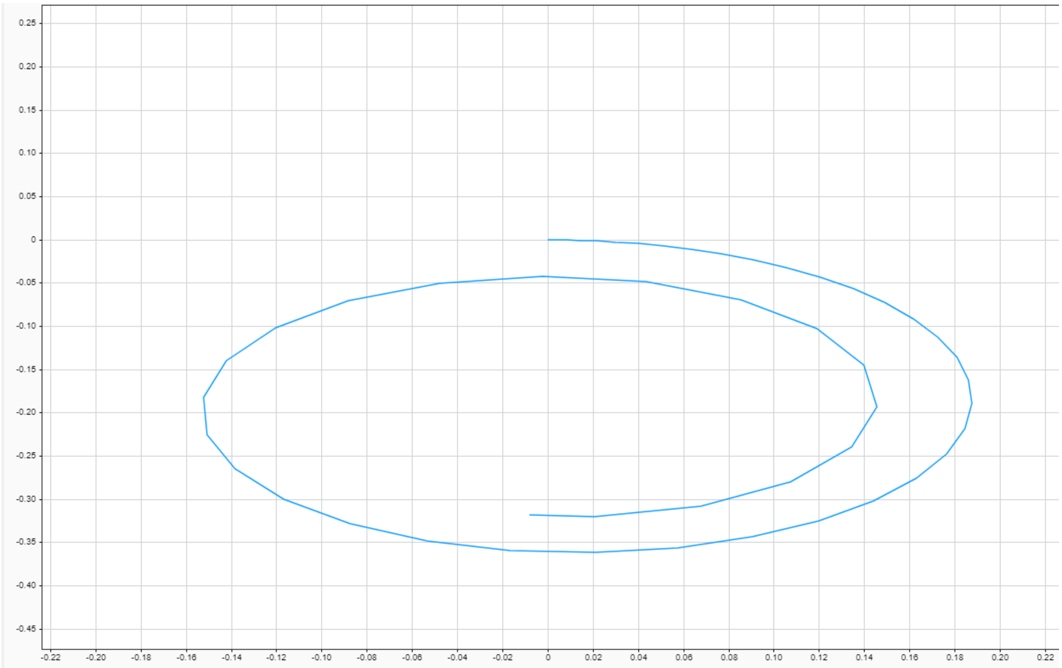

Figure 5.4: x-y plot generated for case 3

○ **Case 4** : tw1= 0.01, tw2= -0.01

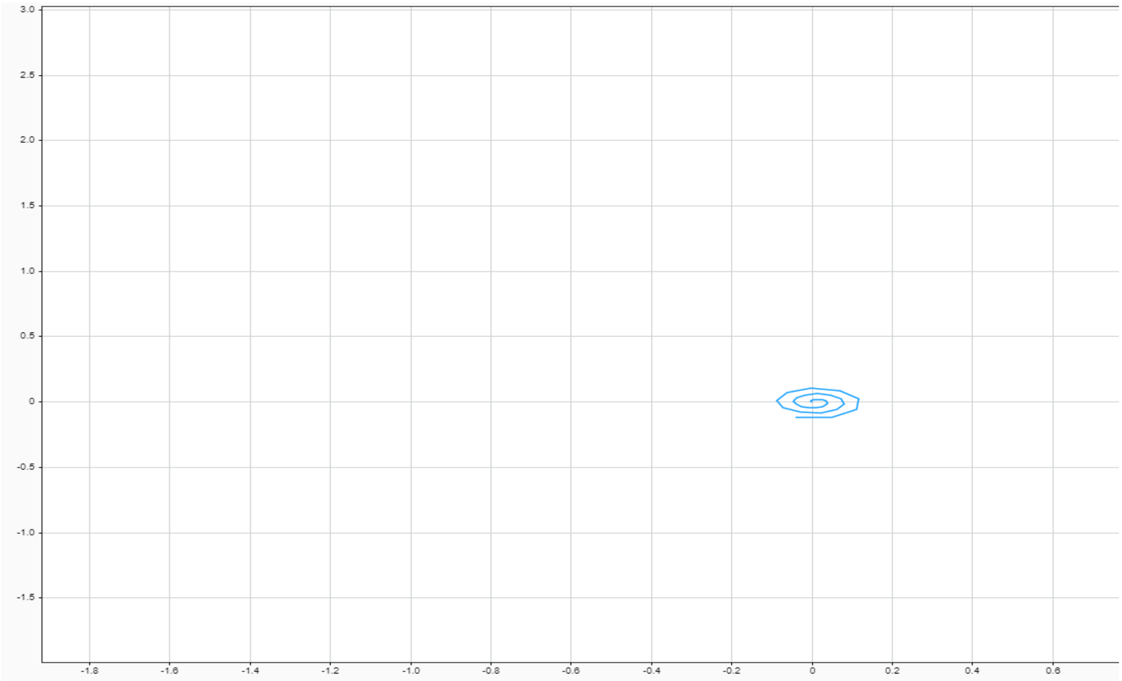

Figure 5.5: x-y plot generated for case 4

5.2 INVERSE MODELLING OF MOBILE ROBOT IN SIMULINK

A trapezoidal path planning blockset is used that takes the waypoints, time and maximum velocity as inputs and generates the trajectory and velocity. This trajectory is then converted to the global velocities and then to the local velocities with the help of jacobian equations formulated earlier.

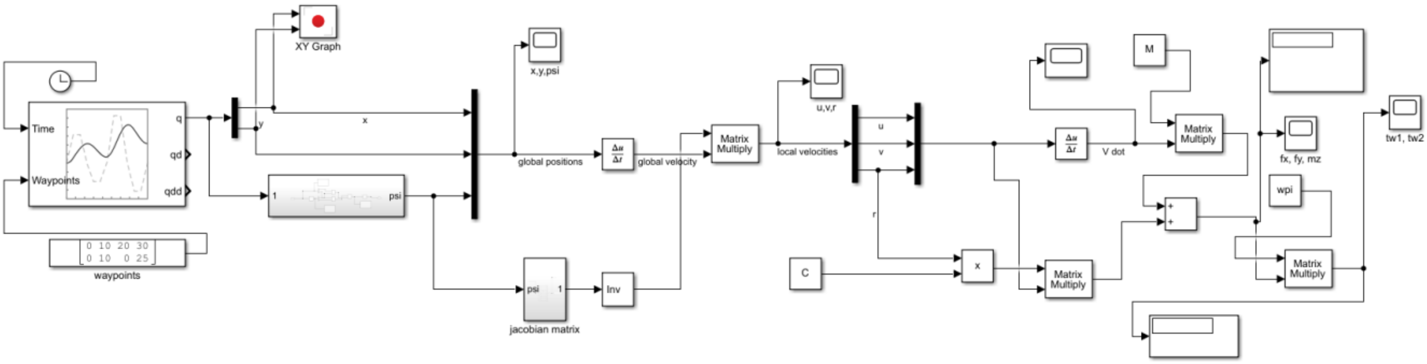

Figure 5.6: Inverse model of the robot in Simulink

A subsystem was created to find the value of ψ where the inputs are taken as y and x for two simultaneous steps and then tan inverse of $\Delta y/\Delta x$ is calculated. This ψ is then fed to the inverse model to finally get the local velocities and wheel torques.

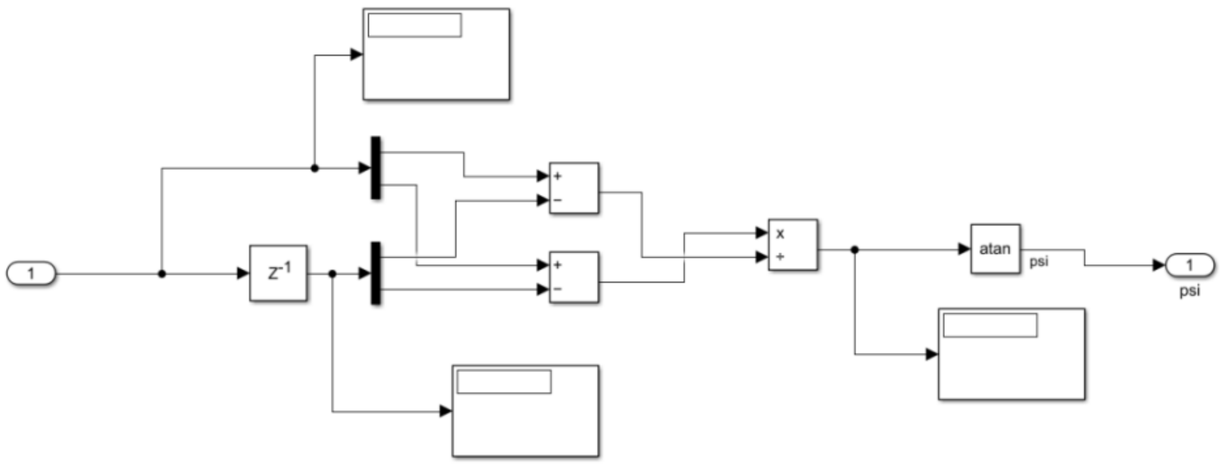

Figure 5.7 : Subsystem in inverse model to find out ψ

RUNNING THE SIMULATION AND RECORDING OUTPUTS

The waypoints provided were $\begin{bmatrix} 0 & 10 & 20 & 30 \\ 0 & 10 & 0 & 0 \end{bmatrix}$ with a peak velocity of [0.1, 0.1] in the x and y direction.

The trajectory generated by the waypoints is plotted. Here the red dots denote the waypoints.

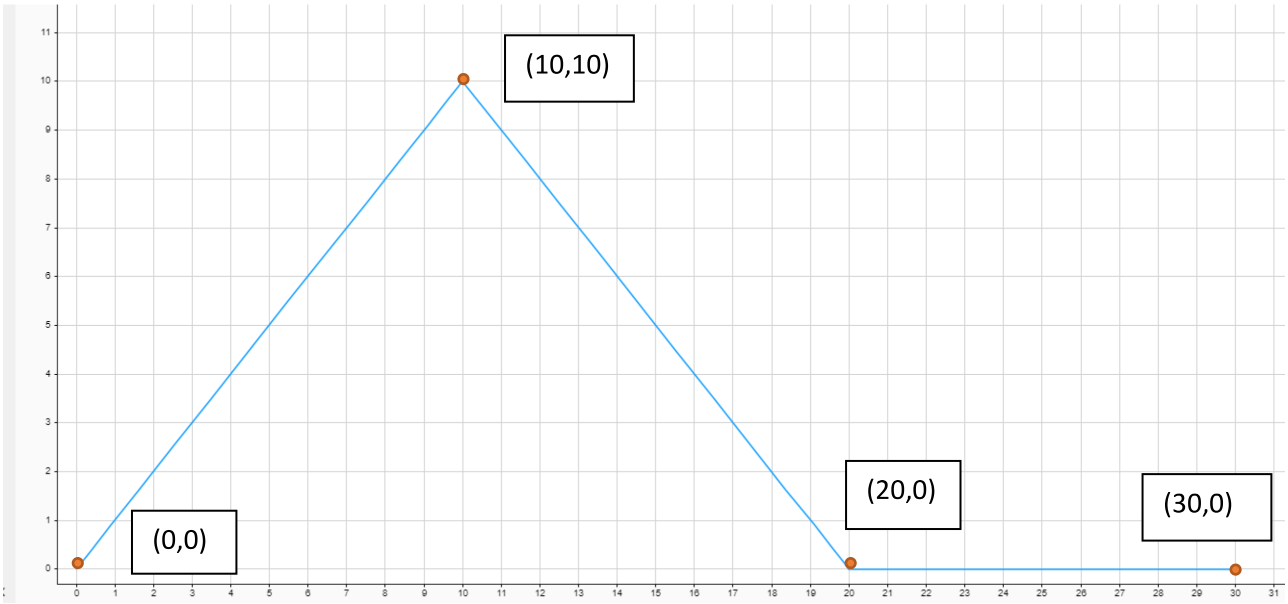

Figure 5.8: position plot X vs Y

This trajectory is fed into the system and then the global velocities are converted into local velocities by multiplying it with jacobian matrix. The plot generated for local velocities are given below. The yellow line depicts u, blue line depicts v and red line depicts r.

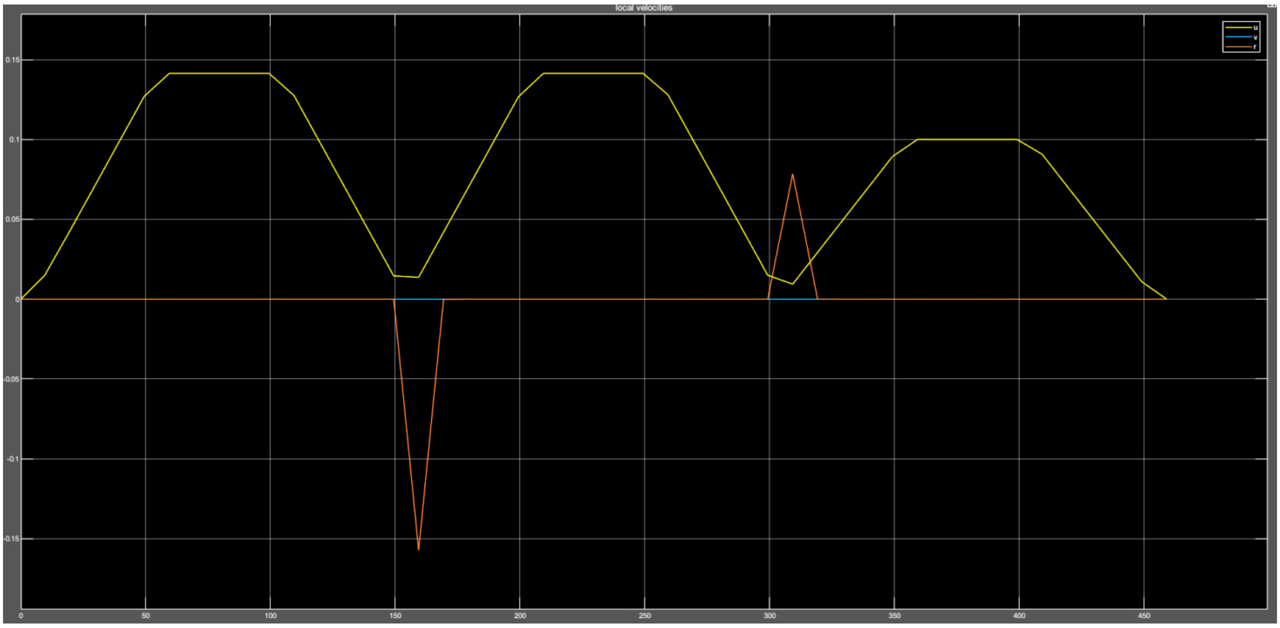

Figure 5.9: Plot of local velocities with respect to time

The local velocities are then differentiated with respect to time to get the acceleration matrix and then multiplied by inertial matrix to get force matrix.

The plot for forceas in local frame of the robot with respect to time is plotted below.
Here yellow line depicts force in x-direction, blue line depicts force in y-direction and red line depicts moment about z-direction

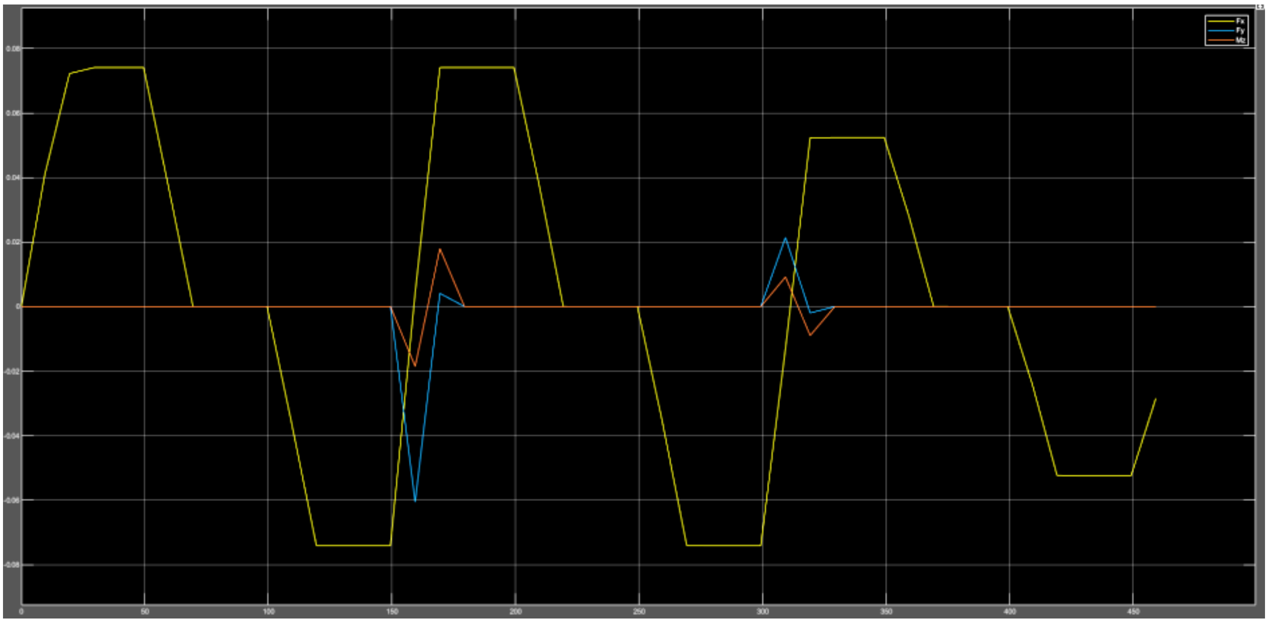

Figure 5.10: Plot of Fx, Fy and Mz with respect to time

The local forces are then multiplies with wheel parameter matrix to get the wheel torques.
The output of wheel torques is plotted with respect to time in the figure below.
Here yellow line depicts tw1- torque on the left wheel and blue line depicts tw2- torque on the right wheel. The wheel torques tw1, tw2 vs time

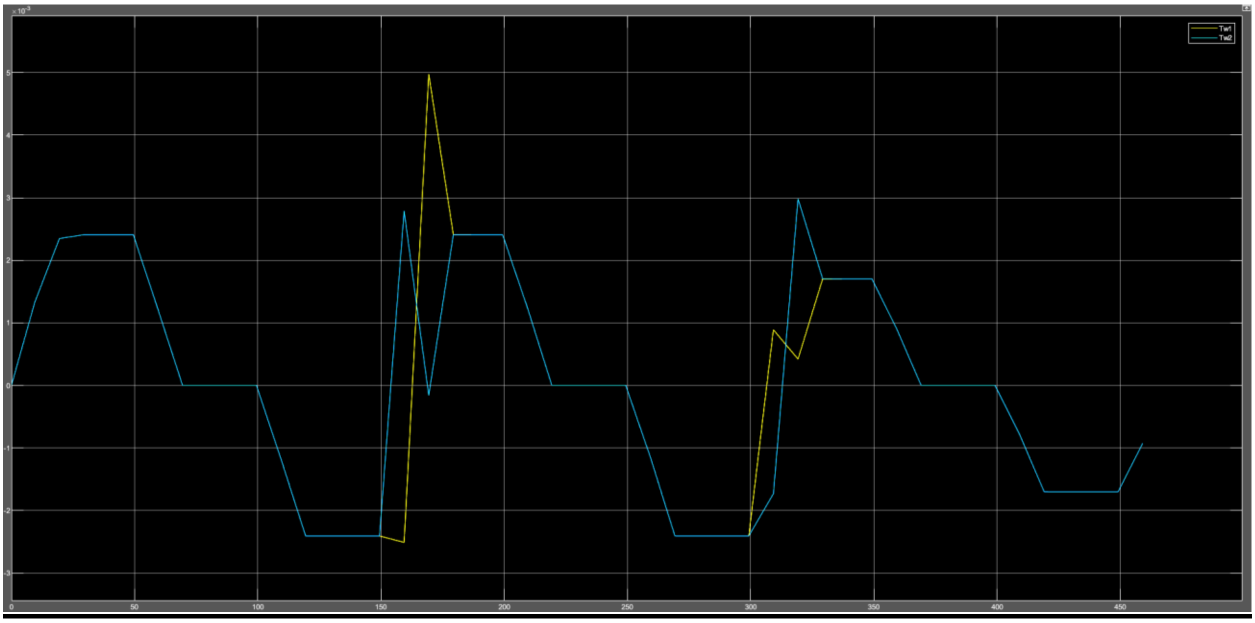

Figure 5.11: Plot of tw1 and tw2 with respect to time

5.3 FORWARD AND INVERSE COMBINED MODEL FOR CHECKING ERROR

To find the error we combine both these models by taking the output from the inverse model that is the wheel torques and feeding it to the inverse model and comparing the trajectories generated as the output of the model and the given trajectory to calculate the error.

5.3.1 COMBINED MODEL FOR CHECKING ERROR WITHOUT PID

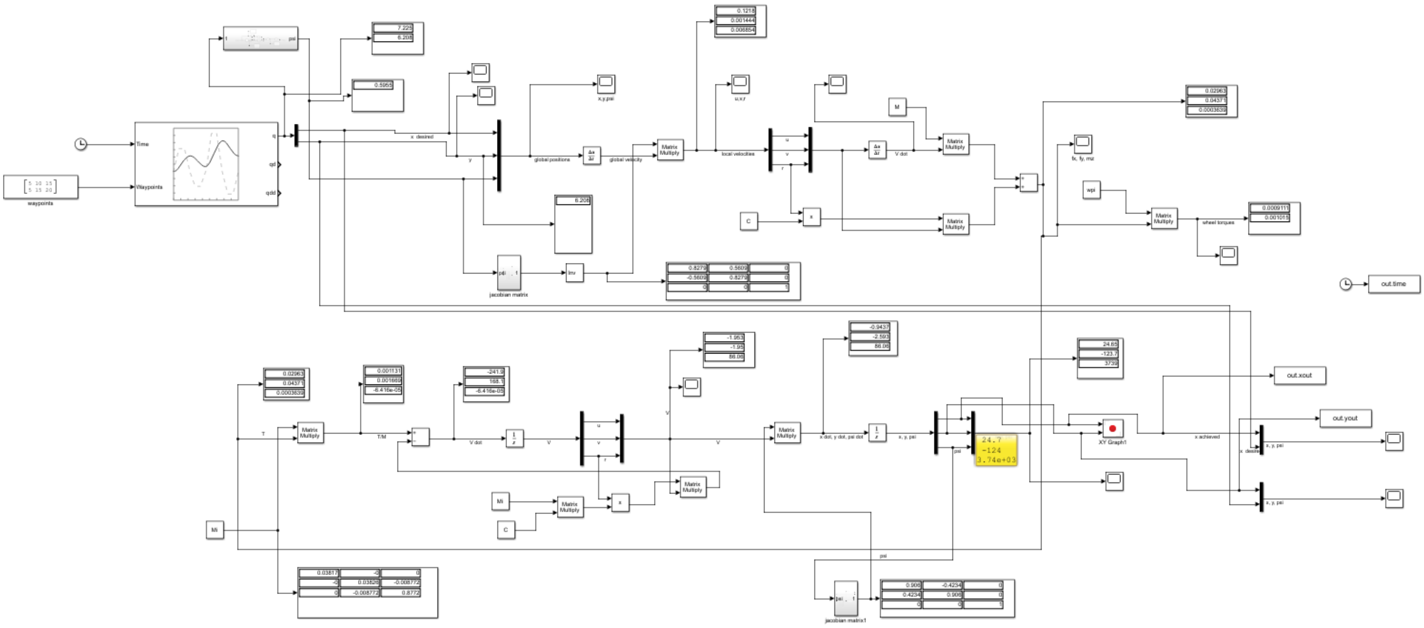

Figure 5.12: Combined Simulink model without PID controller

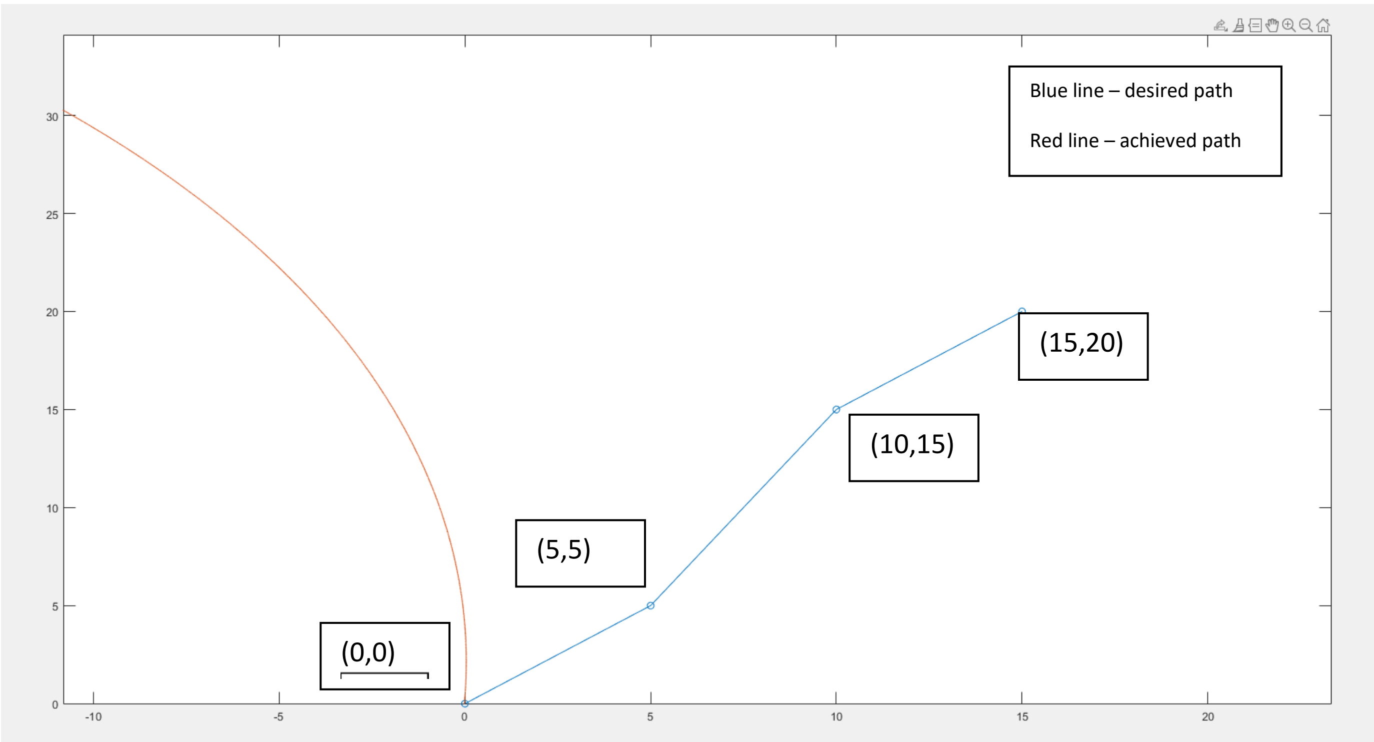

Figure 5.13: Desired position vs actual position in model without PID controller

5.3.2 MODEL WITH PID CONTROLLER AND ITS TUNING

A proportional–integral–derivative controller (PID controller or three-term controller) is a control loop mechanism employing feedback that is widely used in applications requiring continuously modulated control. A PID controller continuously calculates an error value as the difference between a desired position and achieved position and applies a correction based on proportional, integral, and derivative terms (denoted *P*, *I*, and *D* respectively), hence the name.

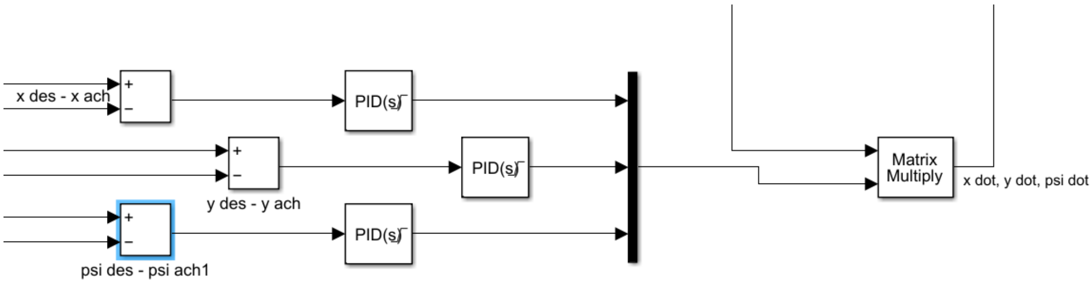

Figure 5.14: Simulink block of PID controller

We first calculate the error between the achieved and desired positions in the global frame and convert it to local frame by multiplying it with jacobian matrix i.e

We then take the output of the pid controller and replace it in the place of $\dot{V}$ in the equation $M\dot{V} + CV = \tau$ which gives us the τ as per the equation, $\tau = Mu - CV$ where u is the output of the pid controller. $u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}$

To tune the PID controller we have used the automated tuning and used the tuning method:

Transfer function based (PID tuner app) $\boldsymbol{\tau}^* = \boldsymbol{J}_q^{\dagger}(\boldsymbol{\eta})^{\top}\boldsymbol{\tau}$

5.3.3 COMBINED MODEL FOR CHECKING ERROR WITH PID

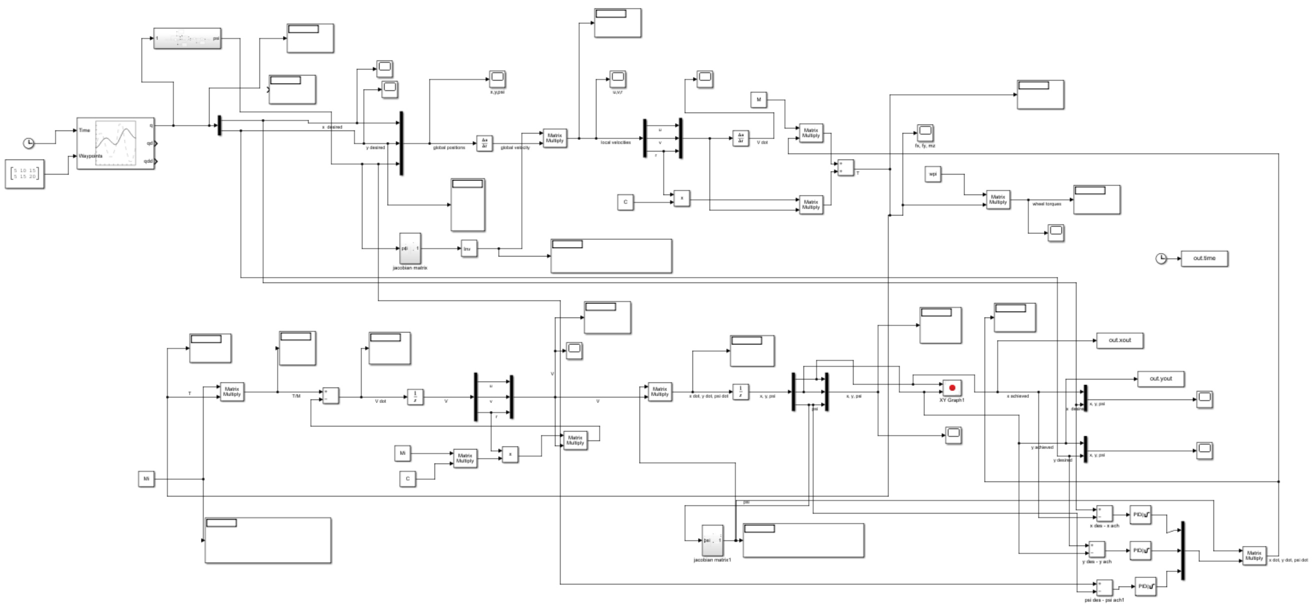

Figure 5.15: Combined Simulink model with PID controller

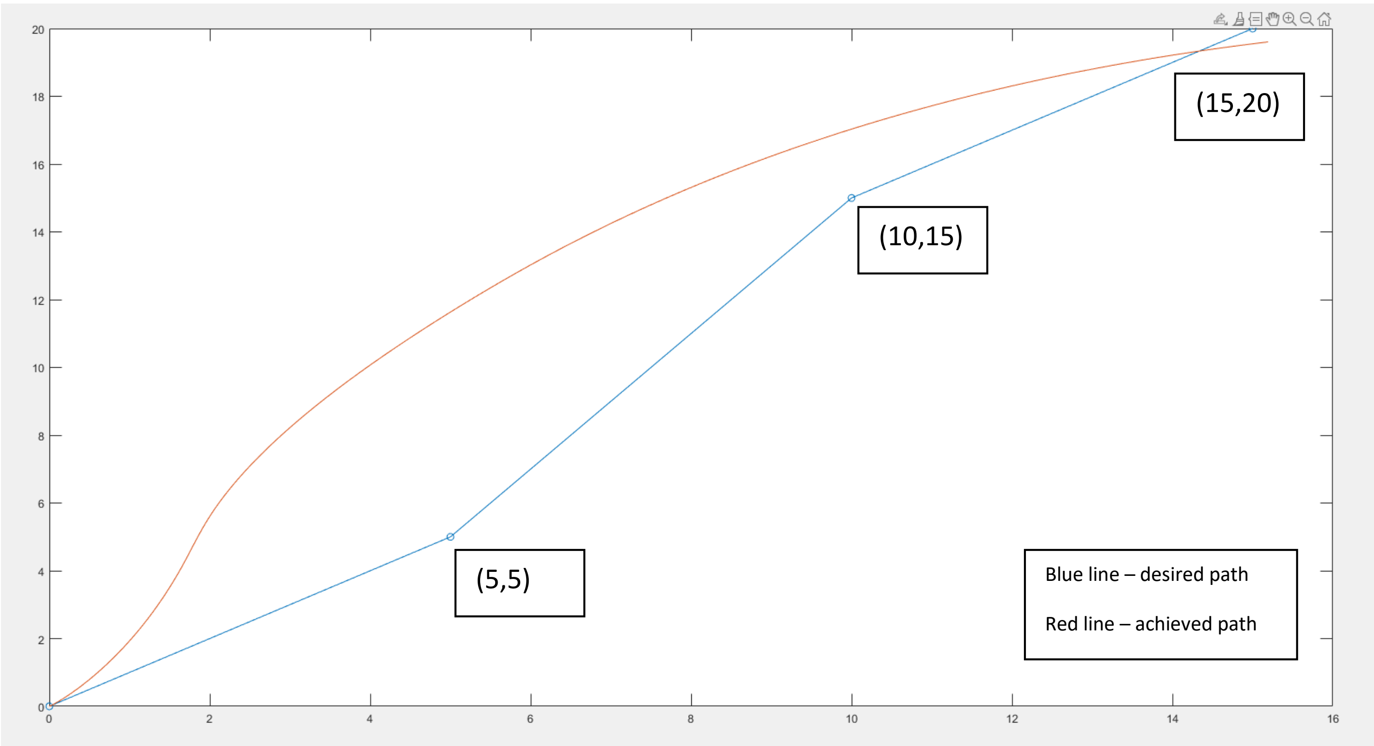

Figure 5.16: Desired position vs actual position in model with PID controller

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 CONCLUSION

In this project the path planning algorithm of mobile robot has been formulated. The first step was to formulate the kinematics and dynamics equations of the movement of mobile robot. The kinematic model of the wheeled robot was created in the MATLAB environment and simulated under various motion parameters to get the plot with respect to time. A CAD model of the physical robot **RM100** by **Peer Robotics** was modelled taking all the dimensions of the real world robot to get an idea of its mass properties. The Simulink models of the forward and inverse equations were created and then they are combined to check its error.

The plots from the forward model in Simulink gave an idea of the motion of the robot. If both the wheels are provided equal torques they will move in straight line. If the left wheel torque is lower than the right wheel torque, then the mobile robot will tend to rotate towards the left direction as it moves because of the differential drive. If the left and right wheel torques is equal and opposite in direction then the robot will rotate around its own axis as there will be no forward motion of the mobile robot.

The inverse model takes in the input in the form of waypoints for the trapezoidal path generation blockset to get the trajectory. The plots from the inverse model give us the local velocities with respect to time, local forces with respect to time and wheel torques with respect to time.

Both the inverse and forward model is combined to get the difference between the path desired and path achieved. This plot gives us an idea of error the model is facing because of certain mathematical conditions such as integration of a function leading to infinity, division of zero by zero which cannot be defined. The path desired vs. path achieved is plotted under the same graph. It can be observed that there is a large difference between the two values.

To reduce the error between the desired and actual path PID controller is incorporated in the model for x, y and psi achieved. The PID blockset is then tuned to get an approximate control of the achieved path. It can be observed in the plots that the difference between the desired path and achieved path is reduced.

6.2 FUTURE SCOPE

- The PID controller need to be tuned more to get an accurate path that is desired.
- The model can be simulated by using other path generation techniques.
- The model can be modified to simulate the movement of robots under loaded conditions and movement along deformable terrains
- There is a need to use various sensors in the physical robot which can automatically determine the obstacle and modify its path according to the obstacle.
- There is still a need to devise the path planning algorithm which are safe as well as cover the shortest path.

REFERENCE

[1] Maulana, Eka, M. Aziz Muslim, and Akhmad Zainuri. "Inverse kinematics of a two-wheeled differential drive an autonomous mobile robot." *2014 Electrical Power, Electronics, Communicatons, Control and Informatics Seminar (EECCIS)*. IEEE, 2014.

[2] Tang, Chin Pei. "Differential flatness-based kinematic and dynamic control of a differentially driven wheeled mobile robot." *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2009.

[3] Ivanjko, Edouard, Toni Petrinic, and Ivan Petrovic. "Modelling of mobile robot dynamics." *7th EUROSIM Congress on Modelling and Simulation*. Vol. 2. 2010.

[4] B V L Deepak , Dayal R Parhi, "KINEMATIC ANALYSIS OF WHEELED MOBLIE ROBOT", J. Automation & Systems Engineering 5-2 (2011): 96-111

[5] Leena, N., and K. K. Saju. "Modelling and trajectory tracking of wheeled mobile robots." Procedia technology 24 (2016): 538-545.

[6] A. Filipescu, V. Minzu, Bogdan Dumitrascu and Eugenia Minca, "Trajectory-Tracking and Discrete Time Sliding Mode Control of Wheeled Mobile Robots", Internalational conference on Information and Automation, Shenzen, China, 2011

[7] Raja, Purushothaman, and Sivagurunathan Pugazhenthi. "Optimal path planning of mobile robots: A review." *International journal of physical sciences* 7.9 (2012): 1314-1320.

[8] https://en.wikipedia.org/wiki/Motion_planning

[9]https://books.google.co.in/books?hl=en&lr=&id=gmYALDVqlLUC&oi=fnd&pg=PP1&dq=dynamic+modelling+of+differential+wheel+mobile+robot&ots=55CQis-jjr&sig=gCagC_mQDWNVFsqco00Sg2mM5c4