International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 09 Sep. 2024, pp: 723-737 www.ijaem.net ISSN: 2395-5252

Model Predictive Control for Advanced Path Tracking and Stabilization in Autonomous Mobile Robots Using Linearized Kinematic and Dynamic Models

Adeleke Olorunnisola Oyeymi[1], Hamza Habib Oladayo[2], Oladunjoye Oladele Olawale[3], Ugorji Leonard Uchechukwu[4], Ofuenweuche Andrew Chidi[5]

[1,2,3,4]*Federal University of Technology, Akure. Ondo State. Nigeria.*
[5]*Obafemi Awolowo University, Ile-Ife. Osun State. Nigeria.*
Corresponding Author: Adeleke Olorunnisola Oyeyemi

Date of Submission: 20-09-2024 Date of Acceptance: 30-09-2024

ABSTRACT
This research presents a comprehensive study on the development and application of Model Predictive Control (MPC) for advanced path tracking and stabilization in autonomous mobile robots, utilizing both linearized kinematic and dynamic models. The mobile robot is modeled using a bicycle model, capturing its essential motion dynamics. These models are linearized at specific operating points to simplify control design while preserving system behavior near those points. Linear MPC controllers are designed for both the linearized kinematic and dynamic models, ensuring robust performance in two key tasks: set point stabilization and tracking a sinusoidal trajectory. The MPC effectively adjusts control inputs—velocity, steering angle, and yaw dynamics—to minimize tracking errors and achieve smooth stabilization. Results demonstrate that the proposed controllers successfully enable precise trajectory following and stabilization, providing a strong foundation for real-time navigation and control of autonomous mobile robots in dynamic environments.

KEYWORDS: Model Predictive Control, Dynamic modelling, Kinematic modelling, and mobile robot.

I. INTRODUCTION

There has been a notable rise in the development of mobile robot due their growing relevance across various sectors due to their ability to automate repetitive tasks and enhance autonomy. [1][2][3][4][5][6] describes some of their areas of application in healthcare, agriculture, manufacturing, warehousing and transportation. As autonomous vehicles gain prominence, the need for advanced control system like Model Predictive Control (MPC) becomes essential enabling robots to make autonomous, real-time decisions while acclimatizing to the dynamic operating conditions[7].

Controllers such as Proportional Integral Derivative (PID) controllers [8]and Linear Quadratic Regulator (LQR) controllers [9]which are effective yet cannot handle complex, multivariable system, constraints and predict future behaviour. This research paper delves into implementation of MPC on mobile robot for stabilization at set points, and precise trajectory tracking.

II. RELATED WORKS

Several researchers have worked on various kinds of mobile robot using MPC. [10][11] used MPC to solve path following problem of an omnidirectional mobile robot. [12] proposed using MPC for control of a P2AT mobile robot and compared it with other controllers.[13] developed a nonlinear predictive controller to control a unicycle-like mobile robot (PIONNER 3-DX) for trajectory tracking. [14] performed a comparative study of Proportional Integral (PI) controller and Model Predictive controller of a wheeled mobile robot. [15] also did a comparative study of PID controller and compared it with the generalized

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 09 Sep. 2024, pp: 723-737 www.ijaem.net ISSN: 2395-5252

predictive control (GPC) and Linear Quadratic model predictive control (LQMPC) algorithms. [16] designed a Non-linear Model Predictive Control (NMPC) for omnidirectional wheeled robot with a guaranteed stability to the non-linear kinematic model.

III. SYSTEM MODEL

The system modelling of a mobile robot, especially a differential drive mobile robot, can be divided into two parts namely kinematics and dynamics which are discussed in section 3.1 and 3.2.

Mobile robot parameters

The system parameters for the vehicle dynamics model are defined as follows:

- $m = 1500$ kg: Mass of the vehicle
- $L = 2.5$ m:Wheelbase length
- $I_z = 3000\text{kg} \cdot \text{m}^2$: Moment of inertia about the z-axis
- v_0=10 m/s: Operating point for velocity
- $\delta = 0.1$ rad:Operating point for steering angle
- F_x: Operating point for longitudinal force

3.1 Kinematic modeling

Kinematic models describe the position and orientation of mobile robot as it evolves over time, based on the wheel velocities. Bicycle kinematic model was used in developing the model as shown in Fig 1.0

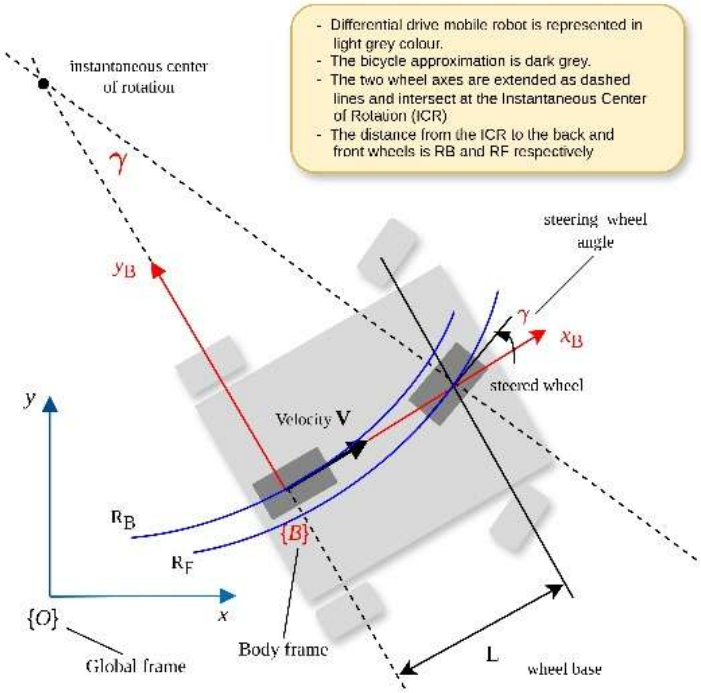

Figure 1.0: Bicycle model approximation of a mobile robot

The bicycle model used assumes the robot has two wheels (front and rear), steers by changing the angle of the front wheel and rear wheel is assumed to follow the front wheel in a continuous path.

Parameters definitions:

- (x, y): the coordinates of the robot in the global frame
- θ: orientation of the robot's body frame relative to the global frame
- v: Linear velocity of the robot
- δ: steering angle of the front wheel relative to the robot's longitudinal axis
- L:distance between the front and rear wheels
- $\dot{x}, \dot{y}, \dot{\theta}$: Time derivatives of position and orientation.

The velocity components in the x and y directions are related to the robot's orientation:

$$\dot{x} = v \cdot \cos(\theta) \quad (1)$$

$$\dot{y} = v \cdot \sin(\theta) \quad (2)$$

The robot changes its heading by turning its front wheels, which results in a change in

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 09 Sep. 2024, pp: 723-737 www.ijaem.net ISSN: 2395-5252

orientation. The rate of change of the robot's orientation $\dot{\theta}$ is a function of the steering angle the velocity v.

The relationship between the steering angle and the robot's turning rate is derived from geometry. The instantaneous turning radius R is:

$$R = \frac{L}{\tan(\delta)} (3)$$

Where L is the distance between the front and rear axles (wheelbase). The turning rate $\dot{\theta}$ is given as

$$\dot{\theta} = \frac{v}{R} = \frac{v}{L} \cdot tan(\delta) \ (4)$$

Equation 1 to 4 describes the motion of the robot in terms of velocity, position and steering angle.

3.1.1 State and Control variables

State vector:

$x(t) -$ position in x axis

$y(t) -$ position in y axis

$\theta(t) -$ orientation (heading angle)

$v(t) -$ velocity

Thus, the state vector $\boldsymbol{x}(t)$ is:

$$\boldsymbol{x}(t) = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} (5)$$

Control input:

$\delta(t) -$ steering angle

$v(t) -$ velocity

Thus, the control input $\boldsymbol{u}(t)$ is:

$$\boldsymbol{u}(t) = \begin{bmatrix} \delta(t) \\ v(t) \end{bmatrix} (6)$$

The state space model for the kinematic model of the system is in the form:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) (7)$$

From equations 1, 2 and 4, we then have;

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cdot cos(\theta) \\ v \cdot sin(\theta) \\ \frac{v}{L} \cdot tan(\delta) \end{bmatrix} (8)$$

Hence the non-linear state-space model is:

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ \frac{1}{L} \cdot tan(\delta) & 0 \end{bmatrix} \boldsymbol{u}(t) (9)$$

3.1.2 Linearization

Given the full nonlinear state space model for the kinematics of the mobile robot in the form:

$$\dot{\boldsymbol{x}} = f(x, u) \quad (10)$$

The jacobian of $f(x,u)$with respect to the state vector x gives the A matrix and the jacobian of $f(x,u)$ with respect to the input vector u is gives the B matrix and is written as:

$$A = \frac{\partial f}{\partial x}\Big|_{x_0, v_0, \delta_0}, \quad B = \frac{\partial f}{\partial x}\Big|_{u_0} \quad (11)$$

With the following specified operating points$v = v_0, \theta = 0 \ and \ \delta = 0$, we now have our linearized state space representation

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & v_0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & \frac{v_0}{L} \end{bmatrix}$$

The output matrix C and feedthrough matrix D are defined as:

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

3.2Dynamic modeling

The dynamic modelling considers the forces and torques acting on the robot to describe its motion. Newton's second law was used for both translational and rotational motion.

The robot's mass m and the forces acting on it in x and y directions determine the robot's translational acceleration. Newton's second law is:

$$F = m \cdot a \quad (12)$$

Where F is the net force, and a is the linear acceleration.

The linear acceleration (x, y) in x and y directions can be written as:

$$\ddot{x} = \frac{F_x}{m} (13)$$

$$\ddot{y} = \frac{F_y}{m} (14)$$

Where F_x and F_y are the components of the forces acting on the car in the x and y directions, respectively.

F_x: Longitudinal force generated by the motor

F_y: Lateral force generated by steering

The forces are functions of velocity v, friction f, and steering angle δ:

$$F_x = m \cdot a_x = m \cdot \dot{v} (15)$$

$$F_y = m \cdot a_y = m \cdot \left(\frac{v^2}{R}\right) = m \cdot \frac{v^2 \cdot tan(\delta)}{L} (16)$$

Herem the lateral force is derived based on how robot moves along a circular path when turning.

The rotational motion is governed by the moment of inertia I_z and the torque τ. Newton's second law for rotational motion is:

$$\tau = I_z \cdot \ddot{\theta} (17)$$

Where:

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 09 Sep. 2024, pp: 723-737 www.ijaem.net ISSN: 2395-5252

τ is the torque applied about the center of mass
I_z is the moment of inertia about the z-axis
$\ddot{\theta}$ is the angular acceleration
The torque τ comes from the forces acting on the front wheels during steering:

$$\tau = L \cdot F_y = L \cdot m \cdot \frac{v^2 \cdot tan(\delta)}{L} (18)$$

$$I_z \cdot \ddot{\theta} = m \cdot v^2 \cdot tan(\delta) (19)$$

Equations 15, 16 and 19 describes the forces and torques necessary to control the robot's movement, taking into account its mass, velocity, and steering inputs.

3.2.1 State and Control variables

State vector:
$v(t) -$ velocity
$\theta(t) -$ Orientation angle (yaw angle)
$\dot{\theta}(t) -$ Angular velocity (yaw rate)
Hence the state vector is:

$$\boldsymbol{x}(t) = \begin{bmatrix} v(t) \\ \theta(t) \\ \dot{\theta}(t) \end{bmatrix} (20)$$

Control inputs:
$F_x(t) -$ Longitudinal force,
$\delta(t) -$ Steering angle

Thus, the control input vector is:

$$\boldsymbol{u}(t) = \begin{bmatrix} F_x(t) \\ \delta(t) \end{bmatrix} (21)$$

From equations 13, 14 and 17, we have;

$$\dot{v}(t) = \frac{F_x(t)}{m} (22)$$

$$\dot{\theta}(t) = \dot{\theta}(t) (23)$$

$$\ddot{\theta}(t) = \frac{m \cdot v^2 \cdot tan(\delta)}{I_z \cdot L} (24)$$

Finally, the full nonlinear state-space model is:

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} \dot{v}(t) \\ \dot{\theta}(t) \\ \ddot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \frac{F_x(t)}{m} \\ \dot{\theta}(t) \\ \frac{m \cdot v^2(t) \cdot tan(\delta(t))}{I_z \cdot L} \end{bmatrix} (25)$$

3.2.2 Linearization

Given the full nonlinear state space model for the dynamics in the form:

$$\dot{\boldsymbol{x}} = f(x, u)$$

Where:

$$f(x,u) = \begin{bmatrix} \frac{F_x(t)}{m} \\ \dot{\theta}(t) \\ \frac{m \cdot v^2(t) \cdot tan(\delta(t))}{I_z \cdot L} \end{bmatrix}$$

The Jacobian of $f(x,u)$with respect to the state vector $x$gives the Amatrix and the Jacobian of $f(x,u)$ with respect to the input vector u is gives the Bmatrix and is written as:

$$A = \frac{\partial f}{\partial x}, \quad B = \frac{\partial f}{\partial u}$$

To analyze the system around an operating point, the nonlinear system is linearized around the following steady-state conditions:

$$v = v_0, \theta = 0, \dot{\theta} = 0, F_x = 0 \; and \; \delta = 0$$

Substituting the operating points into the Jacobians, we obtain the linearized system matrices A and B:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} \frac{1}{m} & 0 \\ 0 & 0 \\ 0 & \frac{mv^2}{I_z L \cos^2(\delta)} \end{bmatrix}$$

The output matrix C and feedthrough matrix D are defined as:

$$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

3.3Control objectives for a Car-like robot

The two control objectives for the mobile robots are set point stabilization and trajectory tracking. These objectives are essential to check the behaviour of robots during various operations.

3.3.1 Set point stabilization objective

The goal of a set point stabilization is to bring a robot to a desired final state and maintain that state. This is required for tasks like parking. The control system should move the robot from its initial position (x_o, y_o, θ_o) to a desired position $(x_d, y_d, \theta_d)$and stabilize and that point. The robot will stop at this desired point with a velocity $v = 0$.

Mathematical formulation:
Given that:
(x_d, y_d, θ_d) is desired final state of the robot
(x, y, θ) is the current state of the robot
e_x is the error between x_d and x
e_y is the error between y_d and y
e_θ is the error between θ_d and θ

$$e_x = x_d - x \quad (26)$$

$$e_y = y_d - y \quad (27)$$

$$e_\theta = \theta_d - \theta \quad (28)$$

The control objective is to design control inputs $v(t) -$ linear velocity and $\delta(t) -$ steering angle such that:

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 09 Sep. 2024, pp: 723-737 www.ijaem.net ISSN: 2395-5252

$$\lim_{t\to\infty} e_x(t) = 0, \lim_{t\to\infty} e_y(t) = 0, \lim_{t\to\infty} e_\theta(t) = 0$$

and

$$v(t) \to 0 \; as \; t \to \infty$$

3.3.2. Trajectory tracking objective

The goal of trajectory tracking is for the robot to follow a predefined path over time $(x_r(t), y_r(t), \theta_r(t))$. This is critical in applications like autonomous driving, where the robot must move along a desired route. To achieve this, the control inputs $v(t)$ – linear velocity and $\delta(t) -$ steering angle should be adjusted to ensure the robot tracks the trajectory with minimal error.

Mathematical formulation:

Given that:

(x_r, y_r, θ_r) is the reference trajectory

$e_x(t)$ is the error between $x_r(t)$ and $x(t)$

$e_y(t)$ is the error between $y_r(t)$ and $y(t)$

e_θ is the error between $\theta_r(t)$ and $\theta(t)$

$$e_x = x_r(t) - x(t) (29)$$

$$e_y = y_r(t) - y(t) (30)$$

$$e_\theta = \theta_r(t) - \theta(t) (31)$$

The control objective is to design control inputs $v(t) -$ linear velocity and $\delta(t) -$ steering angle such that:

$$\lim_{t\to\infty} e_x(t) = 0, \lim_{t\to\infty} e_y(t) = 0, \lim_{t\to\infty} e_\theta(t) = 0$$

This means that the robot's position and orientation should closely follow the reference trajectory over time, minimizing tracking errors.

IV. MODEL PREDICTIVE CONTROL DESIGN

MPC is a model-based optimization for computing the control input using feedback control approach. This optimal control strategy uses the model along with current state (either measured or estimated) to predict the future state of the system for a control input sequence over a short a sampling time. Based on the prediction, the objective function is minimized with respect to future sequence of inputs, thus requiring the solution of a constrained optimization problem for each sampling instant [17]. The first control input applied to system determines the next state, and the algorithm is repeated at the next time step which results in a receding horizon scheme. Hence MPC is also known is as receding horizon control (RHC). Figure 2.0 showsthe MPC scheme works.

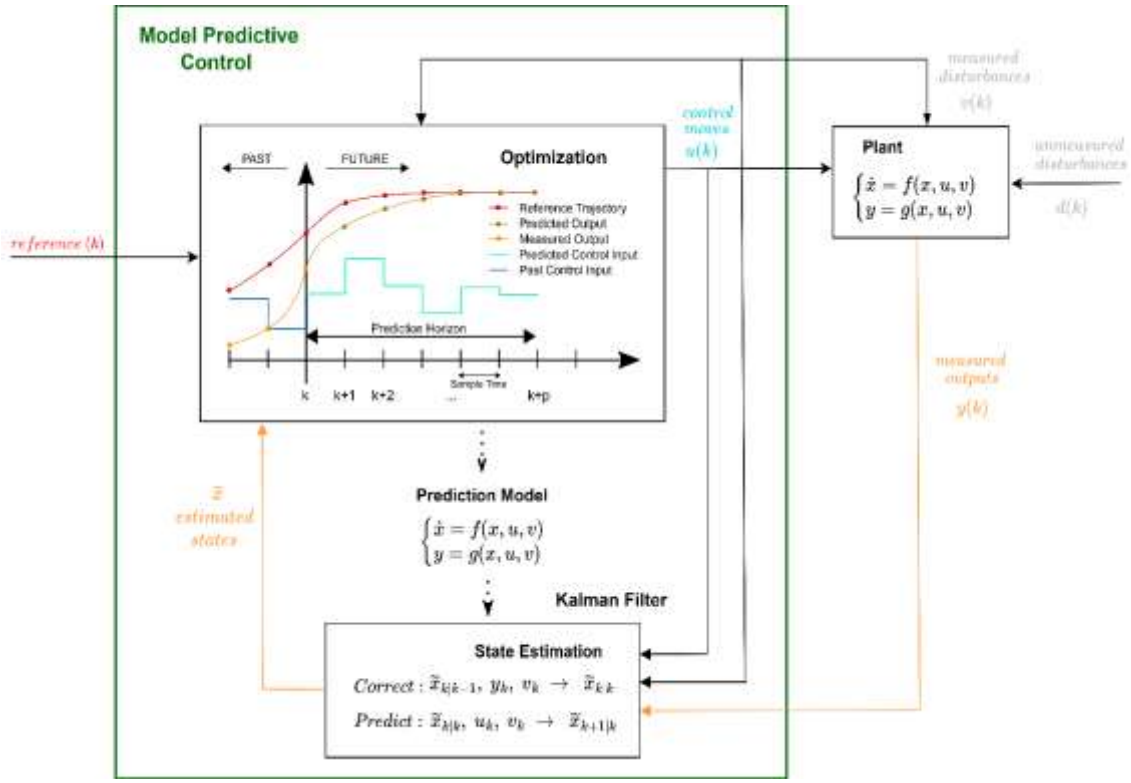

Figure 2.0: Model Predictive control scheme[18][19]

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 09 Sep. 2024, pp: 723-737 www.ijaem.net ISSN: 2395-5252

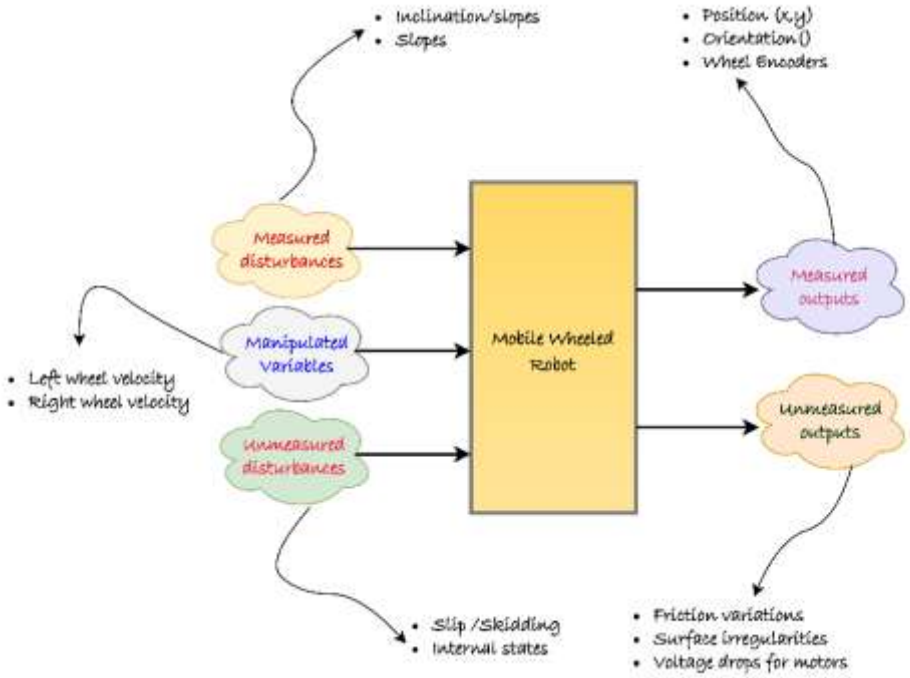

Figure 3.0: Control signals to mobile robot using MPC design

Figure 3.0 depicts the control signals that a mobile wheeled robot receives and sends while utilizing an MPC design. MPC can be classified based on nature of the system model (Linear or Nonlinear MPC), implementation (Implicit or Explicit MPC) and approach (Iterative or Recursive)[20].

4.1. Optimization problem formulation

The obtained non-linear systems are linearized around a specific operating point for control design purposes, resulting in a discrete-time linear time-invariant (LTI) system. Hence, Linear MPC can be applied to the system.

Consider the discrete-time linear time-invariant (LTI) system:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_{k+1} + \boldsymbol{B}\boldsymbol{u}_k \qquad (\mathbf{32})$$

Where:

$k \in \mathbb{T} = \{0, 1, \dots, N_T - 1\}$ ⟶discrete time instant

$\boldsymbol{x}_k \in \mathbb{X} \subseteq \mathbb{R}^n$ ⟶State vector

$\boldsymbol{u}_k \in \mathbb{U} \subseteq \mathbb{R}^m$ ⟶Control input vector

$\boldsymbol{A} \in \mathbb{R}^{n\times n}$ ⟶System matrix

$\boldsymbol{B} \in \mathbb{R}^{n\times m}$ ⟶Input matrix

$\mathbb{X}$ ⟶Constraints set for states

$\mathbb{U}$ ⟶Constraints set for control input

N_T ⟶Time horizon

T ⟶Sampling time

The sets $\mathbb{X}$and $\mathbb{U}$are represented by linear inequalities:

$\mathbb{X} = \{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{F_x x} \leq \boldsymbol{g_x}\}$

$\mathbb{U} = \{\boldsymbol{u} \in \mathbb{R}^m : \boldsymbol{F_u u} \leq \boldsymbol{g_u}\}$

The cost function is selected as a quadratic sum of states and control inputs:

$$J_k = \boldsymbol{x}_{N_T}^T \boldsymbol{Q}_{N_T} \boldsymbol{x}_{N_T} + \sum_{k=0}^{N_T-1} \boldsymbol{x}_k^T \boldsymbol{Q}\boldsymbol{x}_k + \boldsymbol{u}_k^T \boldsymbol{R}\boldsymbol{u}_k \ (\mathbf{33})$$

Where$\boldsymbol{Q}_{N_T} \in \mathbb{R}^{n\times n}, \boldsymbol{Q} \in \mathbb{R}^{n\times n}, \boldsymbol{R} \in \mathbb{R}^{n\times m}$ are weighting matrices used for relatively weighting the sates and control inputs and should be selected such that $\boldsymbol{Q}_{N_T} \geq 0$, $\boldsymbol{Q} > 0$,$\boldsymbol{R} > 0$.

State and control input sequence is defined as$\boldsymbol{X} = (\boldsymbol{x}_0, \boldsymbol{x}_1, \dots, \boldsymbol{x}_{N_T})$, $\boldsymbol{U} = (\boldsymbol{u}_0, \boldsymbol{u}_1, \dots, \boldsymbol{u}\boldsymbol{x}_{N_T-1})$ over a time horizon.

Terminal running cost is given by $\boldsymbol{x}_{N_T}^T \boldsymbol{Q}_{N_T} \boldsymbol{x}_{N_T}$, State running cost is given by $\boldsymbol{x}_k^T \boldsymbol{Q}\boldsymbol{x}_k$and Control input running cost is given by$\boldsymbol{u}_k^T \boldsymbol{R}\boldsymbol{u}_k$.

where$\boldsymbol{Q}, \boldsymbol{R}$ and $\boldsymbol{Q}_{N_T}$are weight matrices, and N is the prediction horizon.

More explicitly, it can be written as:

Objective function:

$$\inf_{\boldsymbol{U_k}} \boldsymbol{J_k}$$

Subject to: $\boldsymbol{U_k} \in \mathbb{U}^N$, ⟶Control input constraint

$\boldsymbol{X_k} \in \mathbb{X}^{N+1}$ ⟶Control output constraint

$\boldsymbol{x}_{i+1|k} = \boldsymbol{A}\boldsymbol{x}_{i|k} + \boldsymbol{B}\boldsymbol{u}_{\boldsymbol{x}_{i|k}}$ ⟶System dynamics(34)

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 09 Sep. 2024, pp: 723-737 www.ijaem.net ISSN: 2395-5252

$k \in \mathbb{T}, i = k, \ldots, k+N-1 \longrightarrow$ Sequence of time instant

Representing the Optimization problem as a quadratic programming problem, then we have:

$$\begin{bmatrix} \boldsymbol{x}_{k|k} \\ \boldsymbol{x}_{k+1|k} \\ \vdots \\ \boldsymbol{x}_{k+N|k} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{A} \\ \vdots \\ \boldsymbol{A}^N \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \boldsymbol{B} & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{A}^{N-1}\boldsymbol{B} & \boldsymbol{A}^{N-2}\boldsymbol{B} & \ldots & \boldsymbol{B} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{k|k} \\ \boldsymbol{u}_{k+1|k} \\ \vdots \\ \boldsymbol{u}_{k+N-1|k} \end{bmatrix} \quad (35)$$

Where

$$\boldsymbol{x}_k = \begin{bmatrix} \boldsymbol{x}_{k|k} \\ \boldsymbol{x}_{k+1|k} \\ \vdots \\ \boldsymbol{x}_{k+N|k} \end{bmatrix}, \boldsymbol{U}_k = \begin{bmatrix} \boldsymbol{u}_{k|k} \\ \boldsymbol{u}_{k+1|k} \\ \vdots \\ \boldsymbol{u}_{k+N-1|k} \end{bmatrix} \quad \boldsymbol{A}_X = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{A} \\ \vdots \\ \boldsymbol{A}^N \end{bmatrix},$$

$$\boldsymbol{B}_U = \begin{bmatrix} \boldsymbol{u}_{k|k} \\ \boldsymbol{u}_{k+1|k} \\ \vdots \\ \boldsymbol{u}_{k+N-1|k} \end{bmatrix} \quad (36)$$

Which defines the system dynamics in such a way that the predicted state $\boldsymbol{X}_k$ is a function of current state $\boldsymbol{x}_k$ and input sequence $\boldsymbol{U}_k$

$$\boldsymbol{X}_k = \boldsymbol{A}_X \boldsymbol{x}_k + \boldsymbol{B}_U \boldsymbol{U}_k \quad (37)$$

Also, the weighting $\boldsymbol{Q}_X$ and $\boldsymbol{R}_U$ are:

$$\boldsymbol{Q}_X = \begin{bmatrix} \boldsymbol{Q} & \mathbf{0} & \ldots & \mathbf{0} \\ \vdots & & \ldots & \vdots \\ \mathbf{0} & \vdots & \boldsymbol{Q} & \mathbf{0} \\ \mathbf{0} & \ldots & \mathbf{0} & \boldsymbol{Q}_N \end{bmatrix},$$

$$\boldsymbol{R}_U = \begin{bmatrix} \boldsymbol{R} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R} & \ldots & \vdots \\ \vdots & \vdots & & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ldots & \boldsymbol{R} \end{bmatrix} \quad (38)$$

Hence the objective function can be written in terms of $\boldsymbol{X}_k$ and $\boldsymbol{U}_k$ as

$$J_k = \boldsymbol{X}_k^T \boldsymbol{Q}_{N_T} \boldsymbol{X}_k + \boldsymbol{U}_k^T \boldsymbol{R}_U \boldsymbol{U}_k$$

Defining the other terms;

Where: $\left.\begin{matrix} \boldsymbol{Q}_{N_T} \geq \mathbf{0} \\ \boldsymbol{Q} > 0 \\ \boldsymbol{R} > 0 \end{matrix}\right\} \longrightarrow$Condition

$$\boldsymbol{F}_X = \begin{bmatrix} \boldsymbol{F}_X & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{F}_X & \ldots & \vdots \\ \vdots & \vdots & & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ldots & \boldsymbol{F}_X \end{bmatrix}, \quad \boldsymbol{g}_X = \begin{bmatrix} \boldsymbol{g}_x \\ \boldsymbol{g}_x \\ \vdots \\ \boldsymbol{g}_x \end{bmatrix},$$

$$\boldsymbol{F}_U = \begin{bmatrix} \boldsymbol{F}_U & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{F}_U & \ldots & \vdots \\ \vdots & \vdots & & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \ldots & \boldsymbol{F}_U \end{bmatrix}, \quad \boldsymbol{g}_U = \begin{bmatrix} \boldsymbol{g}_u \\ \boldsymbol{g}_u \\ \vdots \\ \boldsymbol{g}_u \end{bmatrix} \quad (39)$$

and the state and control constraints can be represented in terms of $\mathbf{X}_k$ and $\mathbf{U}_k$ as

$$\mathbf{F_X X}_k \leq \mathbf{g_X}$$

$$\mathbf{F_U U}_k \leq \mathbf{g_U}$$

Hence the decision variables can be written as;

$$\mathbf{z} = \begin{bmatrix} \mathbf{X}_k \\ \mathbf{U}_k \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} \mathbf{Q_X} & \mathbf{0} \\ \mathbf{0} & \mathbf{R_U} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{F_X} & \mathbf{0} \\ \mathbf{0} & \mathbf{F_U} \end{bmatrix},$$

$$\mathbf{g} = \begin{bmatrix} \mathbf{g_X} \\ \mathbf{g_U} \end{bmatrix}, \quad \mathbf{F}_{\text{eq}} = [\mathbf{I} - \mathbf{B_U}],$$

$$\mathbf{g}_{\text{eq}} = \mathbf{A_X x}_k \quad (41)$$

And the cost function can be rewritten as;

$$\inf_z \mathbf{z}^T \mathbf{H z}$$

Subject to: (43)

$$\mathbf{Fz} \leq \mathbf{g}$$

$$\mathbf{F}_{\text{eq}} \mathbf{z} = \mathbf{g}_{\text{eq}}$$

V. RESULTS AND DISCUSSION

The MPC model was implemented using MATLAB. Table 2.1 and Table 2.2 shows the parameters used for set point tracking and trajectory tracking for both kinematic and dynamic model of the mobile robot as it tries to achieve its goal.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 09 Sep. 2024, pp: 723-737 www.ijaem.net ISSN: 2395-5252

Table 2.1: Parameters used for set point tracking for both kinematic and dynamic model

S/N	Parameters	Values
1.	Prediction Horizon	10 seconds
2.	Control Horizon	3 seconds
3.	Sampling Time	0.1 seconds

Table 2.2: Parameters used for sinusoidal trajectory for both kinematic and dynamic model

S/N	Parameters	Values
1.	Prediction Horizon	30 seconds
2.	Control Horizon	4 seconds
3.	Sampling Time	0.1 seconds

5.1 Result obtained for kinematic model

Figure 4.0 shows how the MPC controls both the velocity and steering angle to achieve the stabilization. The velocity is reduced over time, gradually bringing the robot to rest at the set point. Meanwhile, the steering angle is adjusted dynamically to correct the robot's heading and position, with these adjustments becoming smaller as the robot approaches the target.

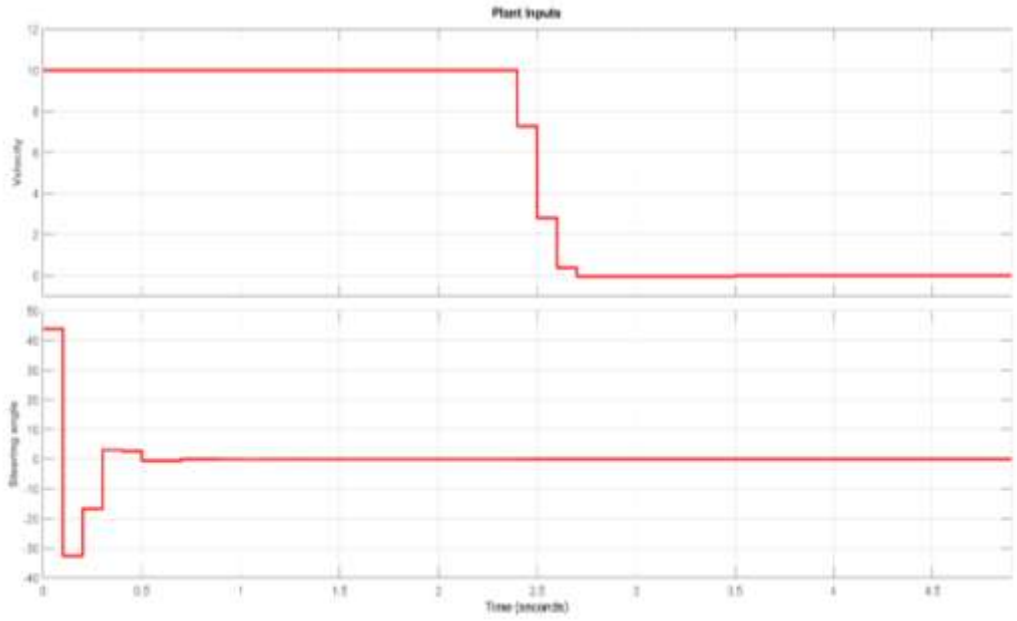

Figure 4.0: Input plot for set point stabilization task using kinematic model

The robot successfully reaches the target position as shown in Figure 5.0. The X-axis position grows and stabilizes, indicating the robot moves toward the desired, Similarly, the Y-axis

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 09 Sep. 2024, pp: 723-737 www.ijaem.net ISSN: 2395-5252

position correction reflects some lateral adjustment due to the slight stabilization performed in the heading angle.

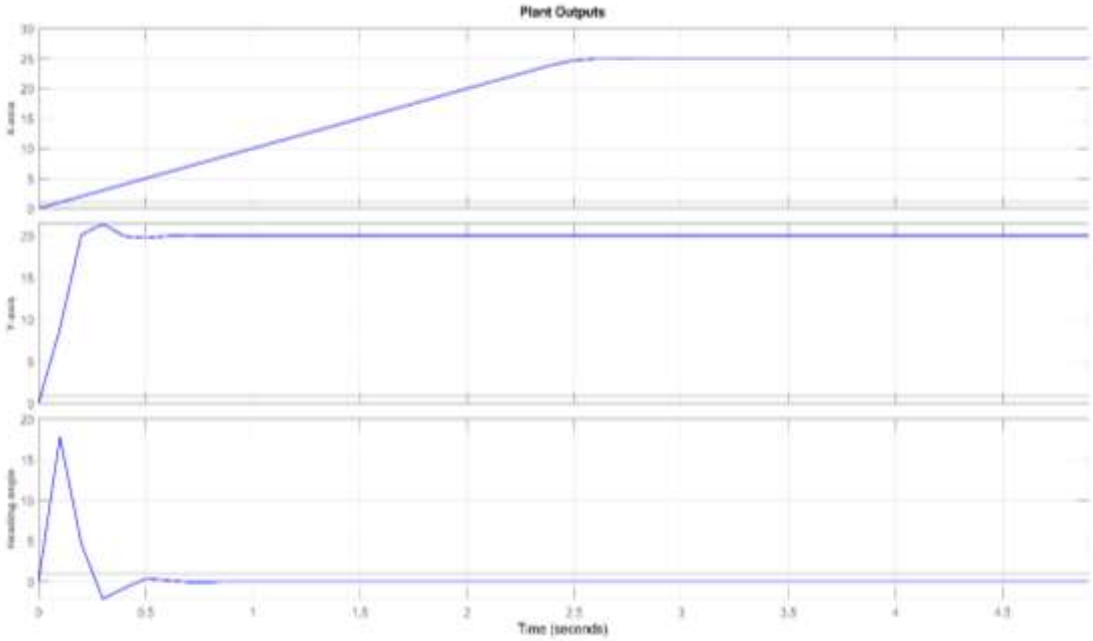

Figure 5.0: Output plot for set point stabilization task using kinematic model

The plot in Figure 6.0 shows the velocity and the steering angle as the control inputs. The velocity curve shows a highly oscillatory pattern which corresponds to the robot trying to adjust its speed to stay on track with the sinusoidal trajectory. The steering angle follows a much smother, sinusoidal-like pattern. This indicates that the robot is making steering corrections that mirror the sinusoidal reference trajectory.

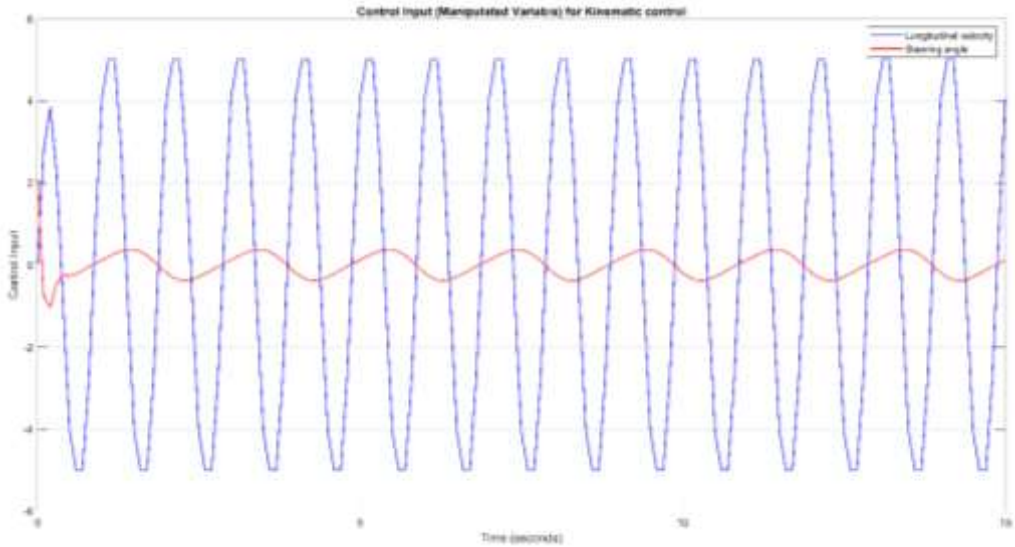

Figure 6.0: Input plot for sinusoidal trajectory for a mobile robot

The sinusoidal trajectory presents a challenging scenario because of the constant changes in direction and speed required, and the robot's response (particularly in velocity) reflects the effort needed to adjust its motion accordingly. Figure 7.0 and Figure 8.0 show that the MPC controller effectively minimized the tracking error for both the X and Y axes. While there are small

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 09 Sep. 2024, pp: 723-737 www.ijaem.net ISSN: 2395-5252

tracking error at the peaks of the sinusoidal curve, the controller manages to keep the robot close to the desired trajectory. These minor errors could be attributed to limitations in the model's kinematic constraints of the robot which may prevent it from perfectly matching the reference at all times.

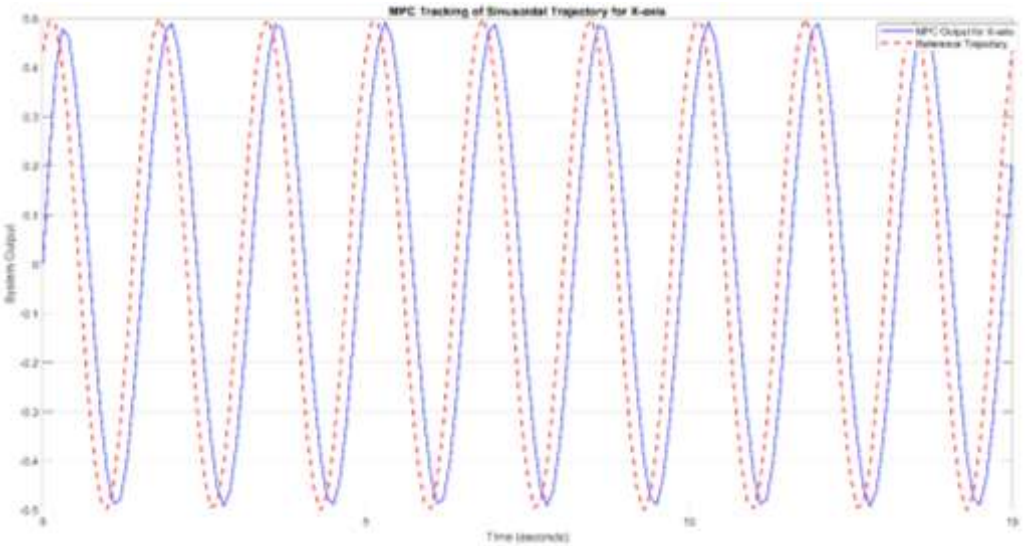

Figure 7.0: MPC tracking of sinusoidal trajectory for X-axis using kinematic model

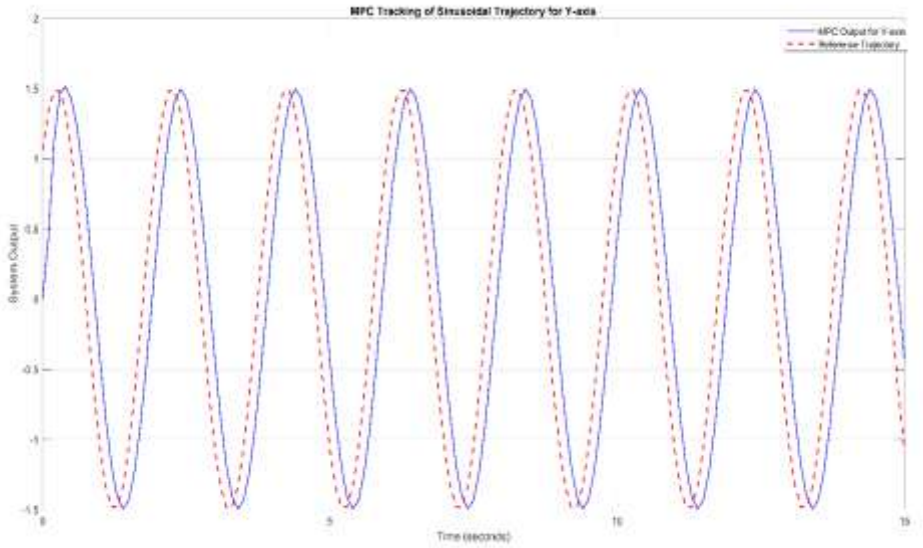

Figure 8.0: MPC tracking of sinusoidal trajectory for Y-axis using kinematic model

The robot's heading angle is key to aligning the robot movement with the sinusoidal trajectory. There are minor deviations in the in the heading angle tracking plot as sheen in Figure 9.0, however, the MPC is able to correct these appropriately.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 09 Sep. 2024, pp: 723-737 www.ijaem.net ISSN: 2395-5252

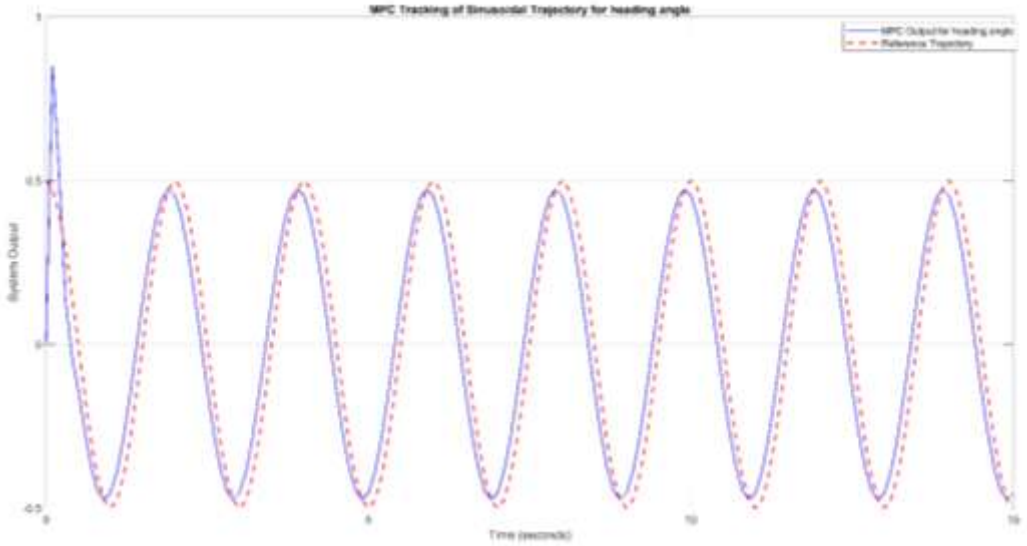

Figure 9.0: MPC tracking of sinusoidal trajectory for heading angle

5.2 Results obtained for Dynamic model

The step input in the longitudinal force reflects the control effort needed to propel the robot toward the set point. As shown in Figure 10.0, after the initial ramp-up, the MPC holds the force constant, likely aiming to stabilize the robot's motion as it approaches the target.

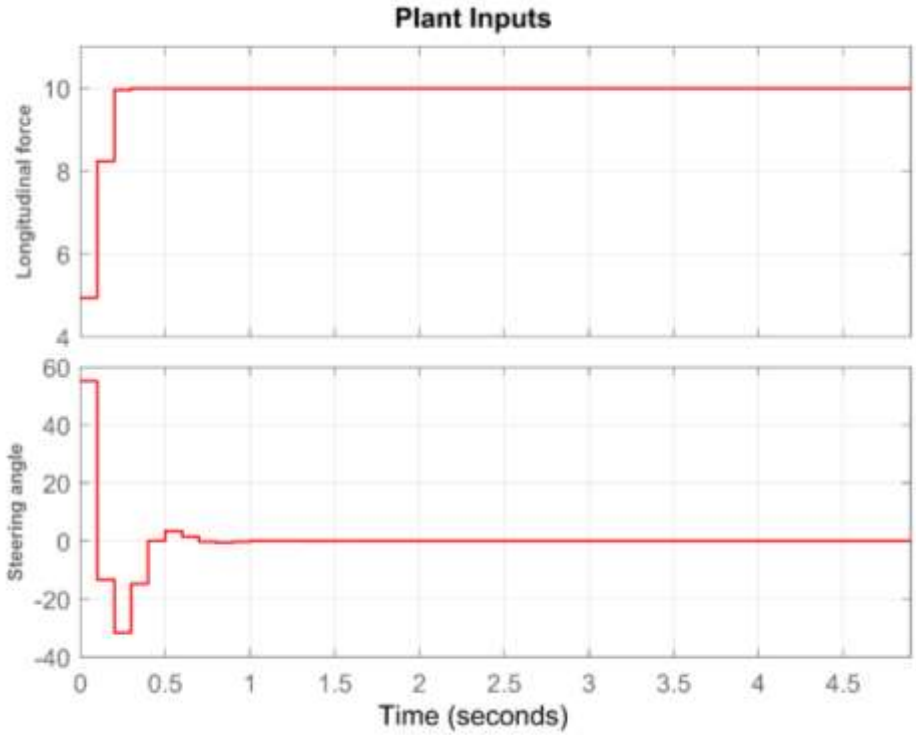

Figure10.0: Input plot for set point stabilization task using dynamic model

Moreover, the steering angle shows some oscillation at the very beginning before quickly dropping to 0 degrees. The initial large steering angle indicates a sharp course correction required to align the robot toward the set point. As the robot stabilizes and moves straight toward the target, the steering angle approaches zero, meaning the robot

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 09 Sep. 2024, pp: 723-737 www.ijaem.net ISSN: 2395-5252

no longer needs to turn as it is already aligned with the desired heading.

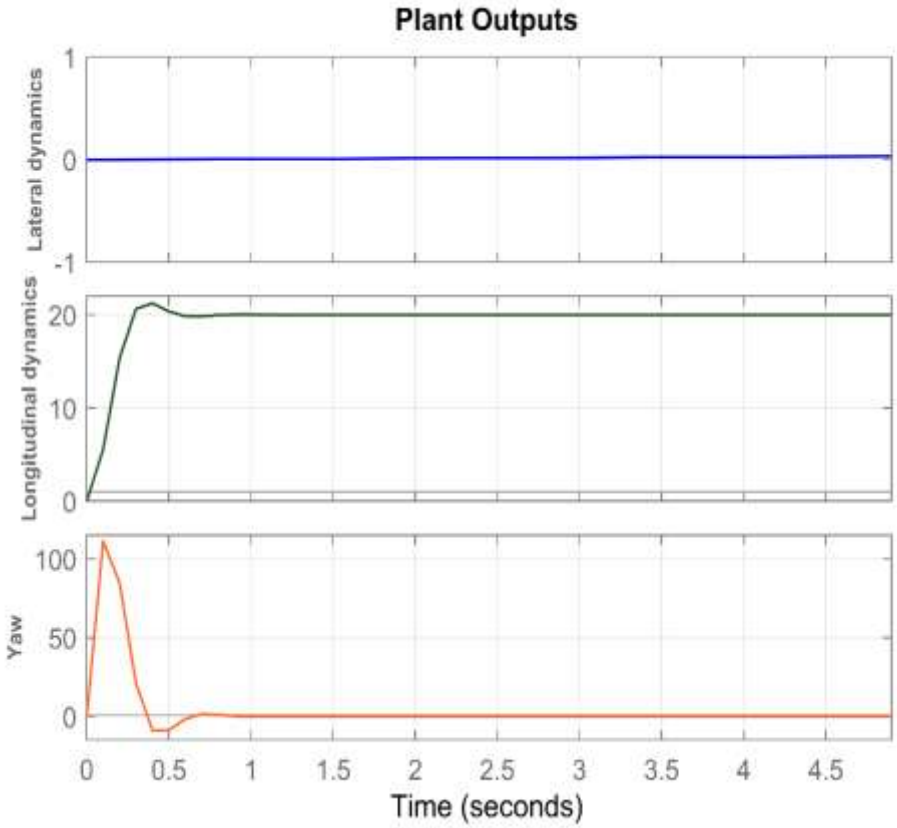

Figure 11.0: Output plot for set point stabilization task using dynamic model

The lateral dynamics are essentially zero throughout the process since the robot did not have a sideway drift confirming the control inputs effectively guided the robot along the desired path. Figure 11.0 shows the longitudinal dynamics plot indicates the robot's forward movement, quickly accelerating to reach the desired position. Once the robot achieves the target speed, the system holds the longitudinal dynamics steady, suggesting that the robot has reached a constant velocity and is maintaining it as it approaches the set point. The initial large yaw shows that there is a need for correction in the robot's heading to align it toward the set point. Once the heading has been corrected, the yaw angle stabilizes which means that the robot no longer rotate as it approaches its set point.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 09 Sep. 2024, pp: 723-737 www.ijaem.net ISSN: 2395-5252

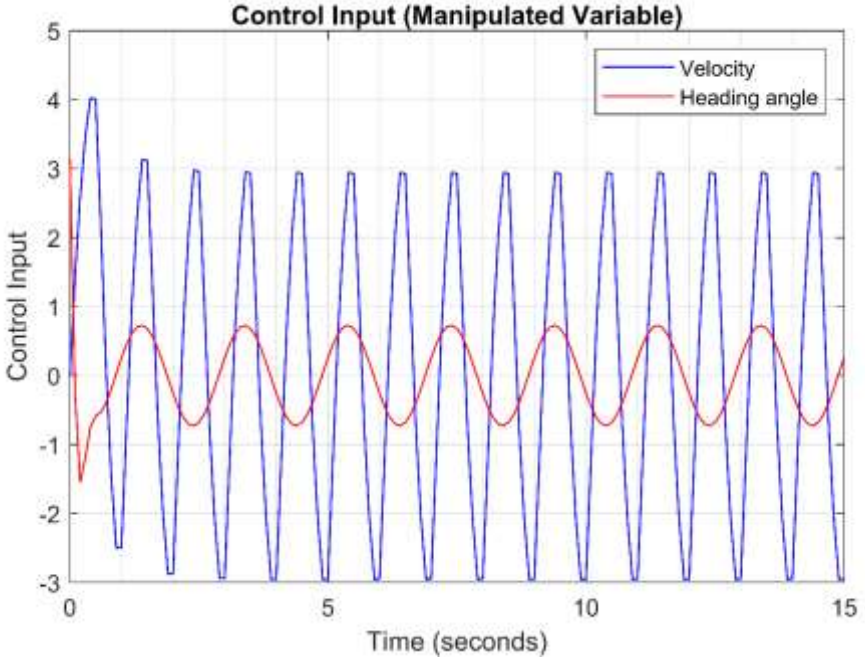

Figure 12.0: Input plot for sinusoidal trajectory for a mobile robot using the dynamic model

The oscillatory velocity input reflects the robot's need to continuously adjust its speed to follow the sinusoidal path. This is a typical response to a sinusoidal trajectory, as the robot needs to accelerate and decelerate to match the changing curvature of the path. The smoother changes in the heading angle indicate that the robot is making less frequent but still continuous steering adjustments to align with the trajectory.

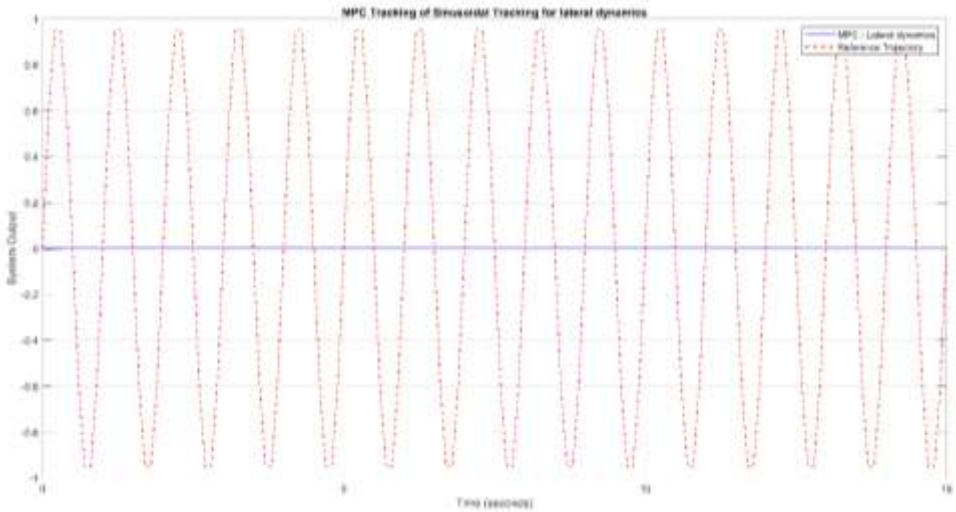

Figure 13.0: MPC tracking of sinusoidal trajectory for lateral dynamics

Figure 13.0 indicates that the control system performed well to keep the robot aligned with the path in such a way that lateral corrections are not needed due to the continuous heading adjustment.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 6, Issue 09 Sep. 2024, pp: 723-737 www.ijaem.net ISSN: 2395-5252

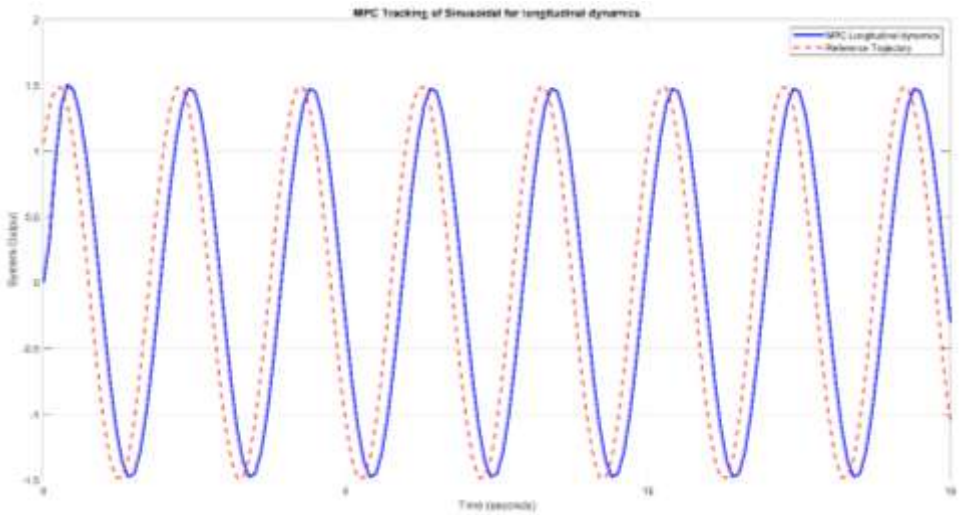

Figure 14.0: MPC tracking of sinusoidal trajectory for longitudinal dynamics

Although there are some minor phase lag which could be due to robot's physical constraints of the mobile robot model. Figure 14.0 still shows that the longitudinal dynamics of the robot closely follows the sinusoidal trajectory in the forward-backward direction.

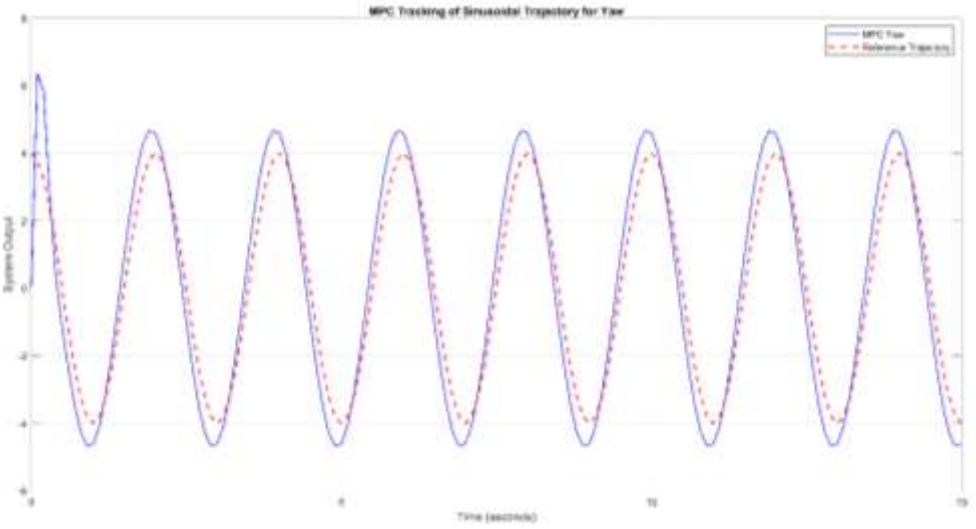

Figure 15.0: MPC tracking of sinusoidal trajectory for lateral dynamics

The yaw dynamics cloely match the sinusoidal reference showing that the robot successfully adjusted the orientation to follow the sinusoidal path. This is crucial for ensuring that the robot stays on track and doesn't deviate from the desired heading.

REFERENCES

[1]. N. Katevas, "Mobile Robotics in Healthcare: the past, the present and the future," 2001, pp. 3–16.

[2]. "Autonomous Mobile Robots in Manufacturing Operations." Accessed: Sep. 23, 2024. [Online]. Available: https://www.researchgate.net/publication/374279729_Autonomous_Mobile_Robots_in_Manufacturing_Operations

[3]. R. Barros, J. Filho, J. Neto, and T. Nascimento, An Open-Design Warehouse Mobile Robot. 2020, p. 6. doi: 10.1109/LARS/SBR/WRE51543.2020.9307137.

[4]. "Robot-assisted for medical surgery: A literature review." Accessed: Sep. 23, 2024. [Online]. Available: https://www.researchgate.net/publication/3

69633254_Robot-assisted_for_medical_surgery_A_literature_review

[5]. "Mobile Robotics in Agricultural Operations: A Narrative Review on Planning Aspects." Accessed: Sep. 23, 2024. [Online]. Available: https://www.researchgate.net/publication/341456084_Mobile_Robotics_in_Agricultural_Operations_A_Narrative_Review_on_Planning_Aspects

[6]. "Autonomous Robotic Self-Driving Cars 2021." Accessed: Sep. 23, 2024. [Online]. Available: https://www.researchgate.net/publication/359340229_Autonomous_Robotic_Self-Driving_Cars_2021

[7]. W. F. Lages and J. A. Vasconcelos Alves, "REAL-TIME CONTROL OF A MOBILE ROBOT USING LINEARIZED MODEL PREDICTIVE CONTROL," IFAC Proc. Vol., vol. 39, no. 16, pp. 968–973, Jan. 2006, doi: 10.3182/20060912-3-DE-2911.00166.

[8]. M. Hamza et al., "PID Based Design and Development of a Mobile Robot Using Microcontroller," 2018, pp. 699–716. doi: 10.1007/978-3-319-95165-2_49.

[9]. A. Abbasi and A. J. Moshayedi, "Trajectory Tracking of Two-Wheeled Mobile Robots, Using LQR Optimal Control Method, Based On Computational Model of KHEPERA IV," vol. 3, pp. 42–47, Mar. 2017.

[10]. K. Kanjanawanishkul and A. Zell, "Path following for an omnidirectional mobile robot based on model predictive control," in 2009 IEEE International Conference on Robotics and Automation, Kobe: IEEE, May 2009, pp. 3341–3346. doi: 10.1109/ROBOT.2009.5152217.

[11]. Y. Han and Q. Zhu, "Robust Optimal Control of Omni-directional Mobile Robot using Model Predictive Control Method," in 2019 Chinese Control Conference (CCC), Jul. 2019, pp. 4679–4684. doi: 10.23919/ChiCC.2019.8865344.

[12]. A. Rezaee, "Model predictive Controller for Mobile Robot," Trans. Environ. Electr. Eng., vol. 2, p. 17, Jun. 2017, doi: 10.22149/teee.v2i2.96.

[13]. F. di Sciascio, "Dynamic nonlinear model based predictive control for mobile robots", Accessed: Sep. 21, 2024. [Online]. Available: https://www.academia.edu/98946162/Dynamic_nonlinear_model_based_predictive_control_for_mobile_robots

[14]. F. Bouani, "Comparative Study of PI Controller and Model Based Predictive Control for Mobile Robot," Univers. J. Control Autom., Jan. 2017, Accessed: Sep. 21, 2024. [Online]. Available: https://www.academia.edu/81549305/Comparative_Study_of_PI_Controller_and_Model_Based_Predictive_Control_for_Mobile_Robot

[15]. M. Shourian, "Comparative Application of Model Predictive Control Strategies to a Wheeled Mobile Robot," J. Intell. Amp Robot. Syst., Jan. 2017, Accessed: Sep. 21, 2024. [Online]. Available: https://www.academia.edu/105738979/Comparative_Application_of_Model_Predictive_Control_Strategies_to_a_Wheeled_Mobile_Robot

[16]. "A Nonlinear Model Predictive Control of an Omni-Directional Mobile Robot | IEEE Conference Publication | IEEE Xplore." Accessed: Sep. 18, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4374943

[17]. F. Kühne, "MOBILE ROBOT TRAJECTORY TRACKING USING MODEL PREDICTIVE CONTROL".

[18]. "Model predictive control," Wikipedia. May 15, 2024. Accessed: Sep. 24, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Model_predictive_control&oldid=1223992680

[19]. "What Is Model Predictive Control? - MATLAB & Simulink." Accessed: Sep. 24, 2024. [Online]. Available: https://www.mathworks.com/help/mpc/gs/what-is-mpc.html

[20]. M. T. Augustine, "Model Predictive Control using MATLAB," Sep. 01, 2023, arXiv: arXiv:2309.00293. doi: 10.48550/arXiv.2309.00293.