Institute of Technology of Cambodia
Department of Electrical and Energy (GEE)

Student Project

Motion Planning for Mobile Robot

Group: I4-GEE-EA-B

Lecturer : Mr. IT Chivorn

Researched by : LANG Bandithvipho ID: e20200104

Academic Year: 2023 - 2024

ABSTRACT

In this report focuses on Motion Planning for Mobile robots to participat in ABU Robocon Contest. The goal is to study on the motion of these robots move precisely and efficiently using a technique called Non-linear Model Predictive Control (NMPC) with point stabilization. Mecanum wheel robots are known for their ability to move in any direction, but controlling them can be tricky due to their complex movements. We tackle this challenge by developing a smart control system that predicts the robot's future movements and adjusts its path accordingly. To test our method, we conduct simulations where the robot navigates through various scenarios. Our results show that our approach significantly improves the robot's ability to move accurately and quickly, which is crucial for success in Robocon. This research offers a practical solution to enhance Mecanum wheel robot performance in competitive settings, making them more competitive and reliable for real-world challenges.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1. Project Overview

The ABU Robocon 2024 contest hosted by Vietnam, has developed robot tasks that depict the stages of rice cultivation. These tasks include sowing, harvesting, and transporting the harvested grains to the warehouse. The underlying message is ’Efficient cultivation brings a warm and prosperous life for everyone”

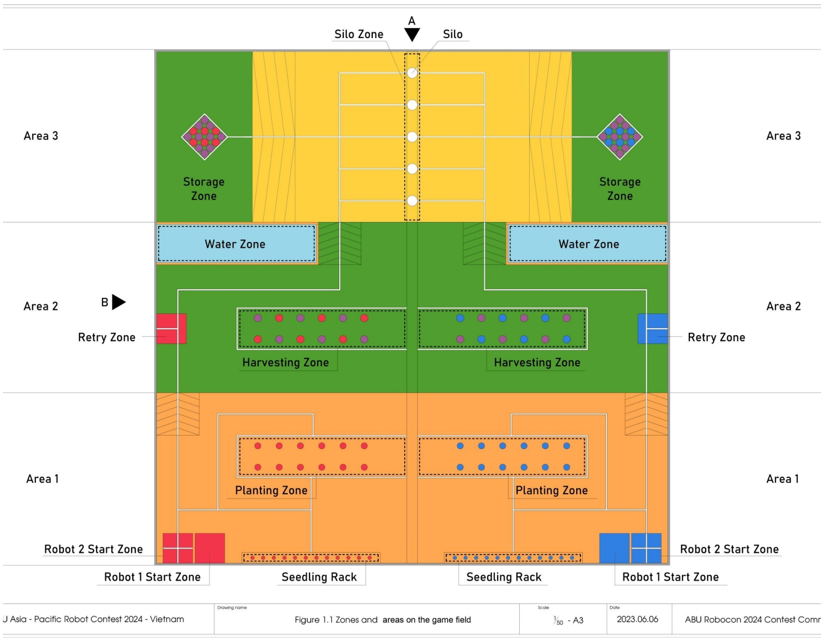

Figure 1.1. Robocon 2024 Game Field

In **Figure 1.1** represented the game field of ABU robot contest 2024. In the game field has given three area such as Planting Zone, Harvesting Zone and Storage Zone. The team have to build 2 robots:

- Robot 1: Represented farmer
- Robot 2: Represented buffalo (Autonomous Robot)

1.2. Objective

In this report study and research with the non linear model predictive control which know as the type of optimal control to apply in the Mecanum Wheel Robot for planning the motion of robot by giving the point or trajectory to the robot for moving from start zone in Area 1 to Storage Zone in Area 3 automatically.

1.3. Scope of work

The scope of works for this research contains four main parts:

- Introduce about Motion Planning
- System Model of Mecanum Wheel
- Optimal Control that we choose Model Predict Control (MPC) algorithm
- Point Stabilization to publish the Goal Point for the Robot
- Simulation Validation
- Future Update

1.4. Outline of Report

There are some flows which contains in this report:

- **Introduction**: Describe about the research overview of report. Especially, the objective and scope of work in this report.
- **Motion Planning**: Provide the methods and describe about Optimal control which is using Non-linear Model Predictive Control
- **Simulation Validation**: Show the result that get from navigation term and discuss about some problem then compare the simulation in gazebo with physical world.
- **Conclusion and Future work**: Telling the achieve progress and go with the future work planing.

2. MOTION PLANNING

2.1. What is Motion Planning?

Motion planning in robotics is the strategy of guiding and determining how a robot should move to achieve a specific goal, considering obstacles, constraints, and the overall optimization of its motion.

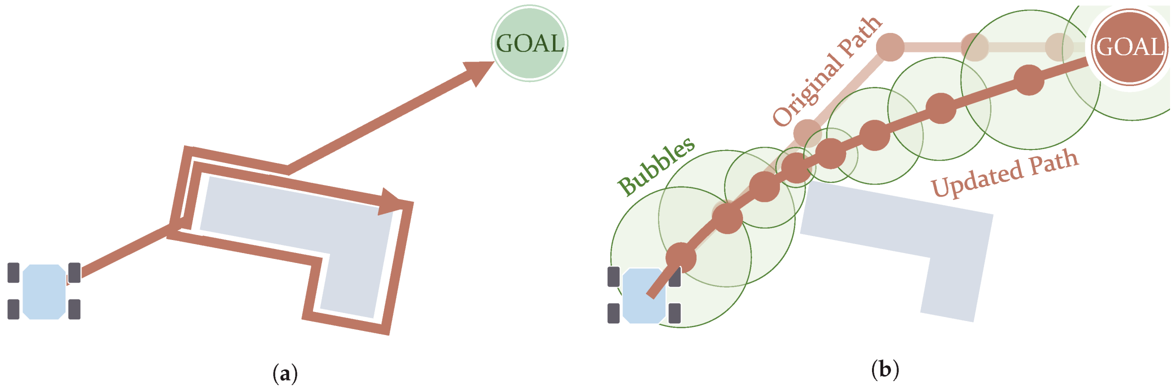

Figure 2.1. Motion Planning for Robot

In **Figure 2.1** Motion planning is like a GPS for robots. Just as a GPS helps you find the best route to your destination, In **Figure 2.1 (b)** the motion planning helps robots figure out the best path to move from one point to another without bumping into obstacles. It's about making smart decisions to navigate safely and efficiently through a complex environment.

2.2. Model Predictive Control

Model predictive control is a method that uses a mathematical model of the system to predict the future behavior of the output and optimize the control input. Unlike traditional control methods that operate based on current states, MPC considers future states and system dynamics over a specified time horizon. By continuously updating predictions and optimizing control actions, MPC enables proactive decision-making to achieve desired system performance while satisfying constraints.

In **Figure 2.2** has showm the Basic concept of Model Predictive Control (MPC) that involved predicting the future behavior of a system based on its current state and using this prediction to optimize control actions.

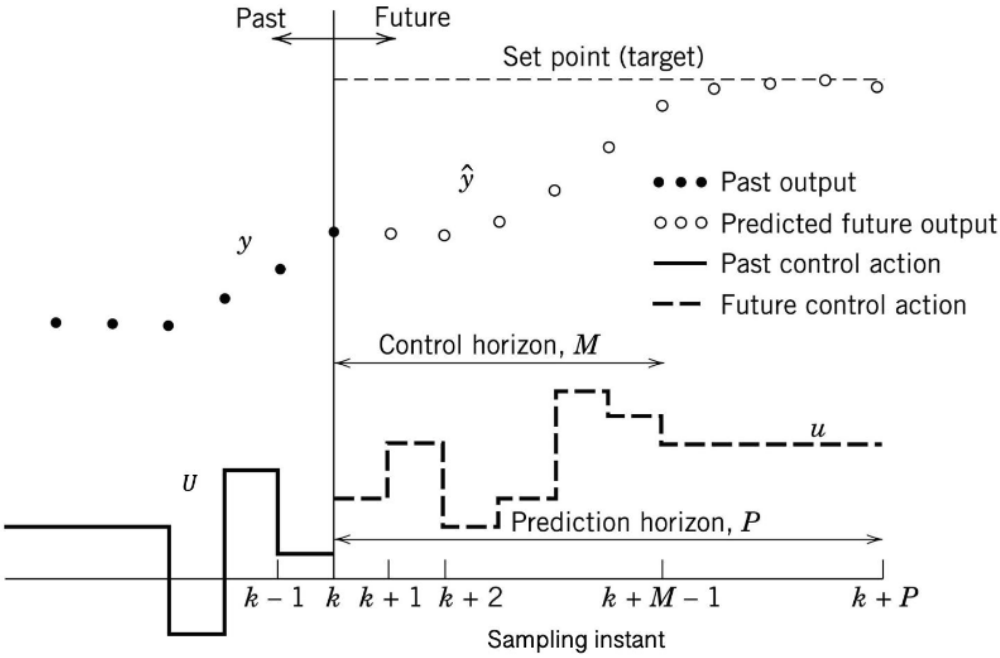

Figure 2.2. Basic Concept for Model Predictive Control Algorithm

Here some parameters that show in graph above:

- Prediction Horizon (P): This represents the number of future time steps for which the system's behavior is predicted. A longer prediction horizon allows MPC to anticipate how the system will evolve over time.
- Control Horizon (M): This indicates the number of future time steps over which control actions are optimized. By adjusting the control horizon, MPC can balance the trade-off between short-term responsiveness and long-term stability.

There are two Methods of Mathematics Model which can be use to represent the Model Predictive Control (MPC), such as Non-Linear MPC and Linear MPC:

- **Linear Model Predictive Control:** Linear MPC simplifies things by assuming the system follows straight-line rules. This makes it easier to crunch numbers and control systems with basic behaviors.
- **Non-linear Model Predictive Control:** Nonlinear MPC handles systems with complex behaviors, like those described by quadratic equations, without needing them to follow simple rules. It can use nonlinear models and constraints, accurately representing intricate system behaviors. Nonlinear MPC is more flexible, capturing a wider range of behaviors than linear MPC

3. OPTIMAL CONTROL

3.1. System Model

We decided to choose Mecanum Wheel to use in our system model. The Mecanum wheel, also known as an omni-directional wheel which can move in any direction and adapt to diverse configurations of the platform's frame.

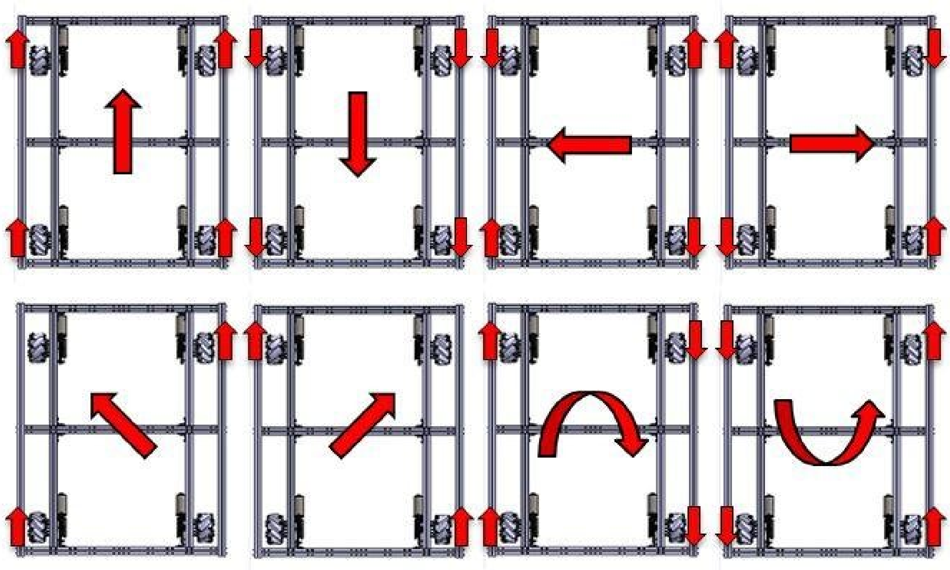

Figure 3.1. Basic driving directions of omnidirectional where the main wheels are rotating forwards or backwards

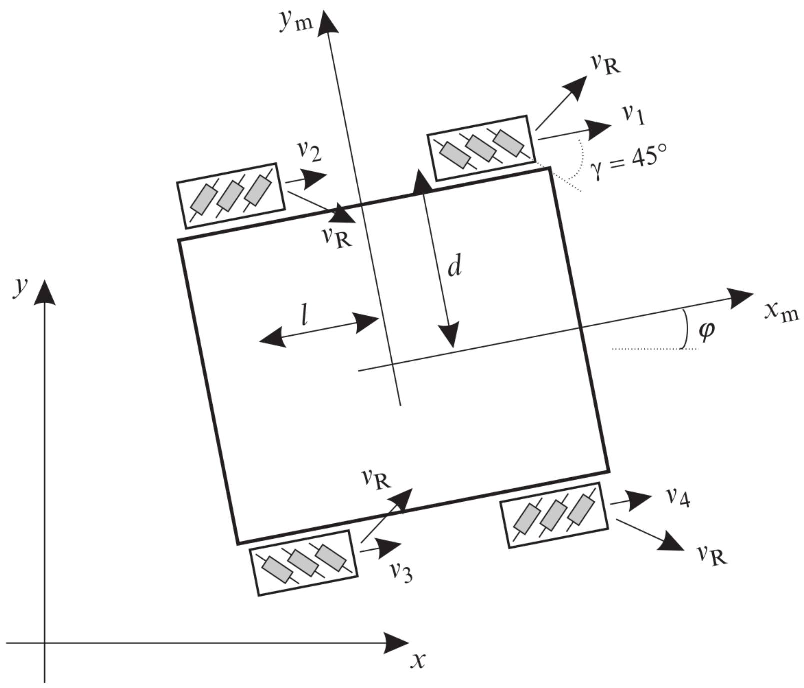

Figure 3.2. Four-wheel Mecanum Represented in Global Frame and Body Frame

From **Figure 3.2** we can derive the equation from Body Frame or Local Frame:

- $v_{m1x} = r \times v_1 + v_R cos(\frac{\pi}{4}) = r \times v_1 + \frac{v_R}{\sqrt{2}}$
- $v_{m1y} = v_R sin(\frac{\pi}{4}) = \frac{v_R}{\sqrt{2}}$

From where the main wheel velocity is obtained:

- $v_1 = v_{m1x} - v_{m1y}$

The first wheel velocity in the robot coordinate frame direction can also be expressed with the robot's translational velocity:

- $v_m = \sqrt{\dot{x_m}^2 + \dot{y_m}^2}$

For the angular velocity represented by $\dot{\phi}$, we can write:

- $v_{m1x} = \dot{x_m} - \dot{\phi} \times \frac{d}{r}$
- $v_{m1y} = \dot{y_m} - \dot{\phi} \times \frac{l}{r}$

From the relations the main wheel velocity can be expressed by the robot body velocity as:

- $v_1 = \dot{x_m} - \dot{y_m} - \frac{(l+d)}{r} \times \dot{\phi}$

Where:

- (l): represent the distance from the robot's center to the wheels projected on the x
- (d): represent the distance from the robot's center to the wheels projected on the y
- (r): represent the radius from the center of the wheel

After we get v_1 as similarly, we also can find the v_2, v_3 and v_4 which is obtained from the **Inverse Kinematic**in Local coordinates:

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -(l+d) & 1 & -1 \\ (l+d) & 1 & 1 \\ -(l+d) & 1 & 1 \\ (l+d) & 1 & -1 \end{bmatrix} \begin{bmatrix} \dot{x_m} \\ \dot{y_m} \\ \dot{\phi} \end{bmatrix} \quad \textbf{(Eq. 3.1.)}$$

Which is:

- $\dot{x_m} = v_x \implies$ Velocity on x-axis
- $\dot{y_m} = v_y \implies$ Velocity on y-axis
- $\dot{\phi} = \omega \implies$ Angular Velocity

The internal Inverse Kinematics in **Eq.3.1** in compact from reads $v = J\dot{q_m}$, where are:

- $v^T = [v_1, v_2, v_3, v_4]^T$

- $\dot{q_m}^T = [x_m, y_m, \phi]^T$

To calculate the Inverse Kinematic in **Global Coordinate**, we can use the Rotation Matrix to represented the orientation of the Local Coordinates with respect to the Global coordinates:

- $q_m = R_G^L \times q$

Since, we know that the R_G^L is a Rotational Matrix which is:

$$R_G^L = \begin{bmatrix} cos(\phi) & sin(\phi) & 0 \\ -sin(\phi) & cos(\phi) & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \textbf{(Eq. 3.2.)}$$

Need to considered as follow:

- $v = J \times R_G^L \times \dot{q}$

From the internal Inverse Kinematics $v = J \times \dot{q_m}$ from **Eq.3.1** the Forward internal Kineamtic is obtained by:

- $\dot{q_m} = J^+ v$

Since, we have:

$$v = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix}$$

And **J** we can define it by using **Pseudo-Inverse** of J:

- $J^+ = (J^T J)^{-1} J^T$

Therefore, we get the **Forward Internal Kinematics** for Four-Wheel Mecanum as follow:

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \frac{r}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{(l+d)} & \frac{1}{(l+d)} & -\frac{1}{(l+d)} & \frac{1}{(l+d)} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} \qquad \textbf{(Eq. 3.3.)}$$

The Forward Kinematic in Global coordinates is obtained by:

- $\dot{q} = (R_L^L)^T \times J^+ v$

3.2. Non-linear Model Predictive Control

In Non-linear Model Predictive Control we work by apply the Quadratic Equation to Solve the Quadratic Cost subject from Optimal Problem to Non-linear functions over the future finite horizon. To be able applying the Model Predictive Control with Mecanum Wheel Robot, we started with the following step:

- Define Non-linear Equation with Quadratic From
- Model Predictive Control Formulation
- Discretirization Method
- Transform Optimal Control Problem to Non-linear Problem

3.2.1. Quadratic Formula

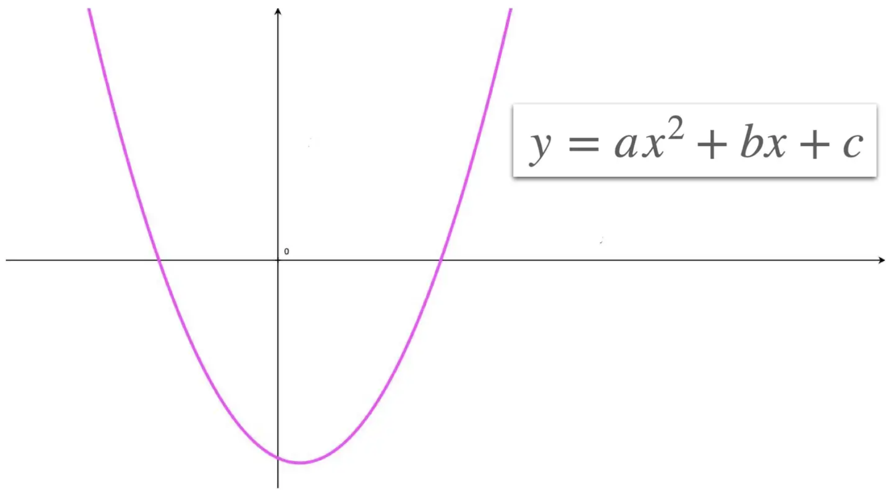

Figure 3.3. Quadratic Graph Represented

A quadratic form is a polynomial function of degree two in a number of variables. And, it can be represented as:

- $Q(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j$
- $x_1, x_2, \ldots, x_n$ are variables
- a_{ij} are constants.

As we can see in **Figure 3.3** the quadratic forms are considered non-linear due to the presence of terms involving the product of two variables, such as x_i, x_j. Where i and j can be different indices. This introduces a level of interaction between variables that doesn't exist in linear functions.

3.2.2. Model Predictive Control Formulation

The formula for MPC that the Quadractic Cost can be define as:

$$J(X,U) = \sum_{k=0}^{N-1} X_k^T Q X_k + U_k^T R U_k + \Phi(X_N) \qquad \textbf{(Eq. 3.4.)}$$

As we have known that the basic idea of Model Predictive control is to solve an optimization problem at each time step to find the control inputs that minimize a cost function while satisfying system constraints. Here, the equation use to minimize a cost function:

$$\min_{u_0,u_1,\dots,u_{N-1}} \sum_{k=0}^{N-1} X_k^T Q X_k + U_k^T R U_k + \Phi(X_N) \qquad \textbf{(Eq. 3.5.)}$$

Subject to $x_{k+1} = f(x_k, u_k)$ (System Dynamic)

$g(x_k, x_k) \leq 0$ (State Constraints)

$h(x_k, x_k) \leq 0$ (Input Constraints)

$p(x_k) \leq 0$ (Path Constraints)

$x_0 = x(0)$ (Initial Condition)

In this formulation, Non-linear Model Predictive Control solves an optimization problem to find the optimal sequence of control inputs u_k over a finite horizon while ensuring that the system dynamics and constraints are satisfied. The optimization is typically solved online at each time step based on current measurements of the system state x_k.

Name of parameters:

- (N) is the number of predict horizon over finite future
- (Q) is Control output weighting Matrix
- (R) is Control input weighting Matrix
- (g) is a state constraint of the system
- (h) is input constraint of the system
- (p) is path constraint (safety path for system)
- $(\phi(X_N))$ is terminal cost of the system

3.2.3. Discretirization Method

To discretirize the system, there are few methods such as:

- Euler-Lagrangian method
- Runge-Kutta 4th order
- Orthogonal collocation on finite element

Since, we have the system:

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

Euler-Lagrangian form:

- $x_{k+1} = x_k + f(x, u)dt$

We need to convert the Forward kinematic Model of Mecanum Wheel to the discrete form by using **Euler-Lagrangian**. From **Eq.3.3** we can write:

$$x_{k+1} = x_k + \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} dt$$

$$x_{k+1} = x_k + \frac{r}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{1}{(l+d)} & \frac{1}{(l+d)} & -\frac{1}{(l+d)} & \frac{1}{(l+d)} \end{bmatrix}\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} dt \qquad \textbf{(Eq. 3.6.)}$$

Noted that: x_{k+1} denote the next state and x_k denote the previous state of the system.

Runge-Kutta 4th order approximation:

$$\begin{aligned} k_1 &= f(x^*, u^*) \\ k_2 &= f(x^* + k_1\frac{dt}{2}, u^*) \\ k_3 &= f(x^* + k_2\frac{dt}{2}, u^*) \\ k_4 &= f(x^* + k_3 dt, u^*) \end{aligned}$$

Then, we get the dicretirization equation:

$$x_{k+1} = x_k + (k_1 + k_2 + k_3 + k_4)\frac{dt}{6} \qquad \textbf{(Eq. 3.7.)}$$

3.2.4. Optimal Transformation Method

To transform the optimal control problem into non-linear problem, there are 2 main methods:

- Single Shooting Method: we simplify the problem by transforming it into a nonlinear programming problem with just one decision variable. This variable represents the entire control trajectory over the prediction horizon.
- Multiple Shooting Method: solve the problem into smaller segments, each with its own decision variable, making it more efficient, especially for longer horizons or complex systems.

Single Shooting Method:

Since, we have the optimal control problem:

$$\min_{u_0,u_1,\ldots,u_{N-1}} \sum_{k=0}^{N-1} X_k^T Q X_k + U_k^T R U_k + \Phi(X_N)$$

$$\begin{aligned} \text{Subject to} \quad & x_{k+1} = f(x_k, u_k) \\ & g(x_k, x_k) \leq 0 \\ & h(x_k, x_k) \leq 0 \\ & p(x_k) \leq 0 \\ & x_0 = x(0) \end{aligned}$$

And, we need to transform it to Non-linear solution which is taking one decision variable:

- $\omega = [u_0, u_1, u_2, \ldots, u_{N-1}]$

Get x_u as a function of ω, x, t_k

- $x_u = F(\omega, x_0, t_k)$

Which, the system:

- $F(\omega, x_0, t_k) = x_0$

Then, we can solve the Non-linear problem as following:

$$\min_{\omega} \Phi(F(w, x_0, t_k), w) \qquad \textbf{(Eq. 3.8.)}$$

Subject to: $g_1(F(\omega, x_0, t_k), \omega) \leq 0$

Where:

- (g) is a state constraint of the system
- (Φ) is a cost function that typically evaluates the performance of the predicted state trajectory and control inputs.
- (ω) is the decision variable, the objective is to minimize this cost function, subject to constraints (g) imposed by the system dynamics (F) and other constraints, such as (p) is a path constraint.

Multiple Shooting Method

Since, we have the optimal control problem:

$$\min_{u_0, u_1, \ldots, u_{N-1}} \sum_{k=0}^{N-1} X_k^T Q X_k + U_k^T R U_k + \Phi(X_N)$$

$$\begin{aligned} \text{Subject to} \quad & x_{k+1} = f(x_k, u_k) \\ & g(x_k, x_k) \leq 0 \\ & h(x_k, x_k) \leq 0 \\ & p(x_k) \leq 0 \\ & x_0 = x(0) \end{aligned}$$

And, we need to transform it to Non-linear solution which is taking two decision variables:

- $\omega = [u_0, u_1, u_2, \ldots, u_{N-1}; \quad x_0, x_1, x_2, \ldots, x_N]$

Then, we can apply the subject constraint at each iteration step as following:

- $x_{k+1} = f(x_k, u_k) = 0$

And, solve the Non-linear problem as following:

$$\min_{\omega} \Phi(F(w, x_0, t_k), w) \qquad \textbf{(Eq. 3.9.)}$$

Subject to: $g_1(F(\omega, x_0, t_k), \omega) \leq 0$

$$\begin{bmatrix} \bar{x} - x_0 \\ f(x_0, u_0) - x_1 \\ \ldots \\ f(x_{N-1}, u_{N-1} - x_N) \end{bmatrix} = 0 \qquad \textbf{(Eq. 3.10.)}$$

3.3. Goal Point Stabilization

In **Figure 3.4** has shown the Point stabilization which is referred to the control objective of driving the system's state to a specific point (Start Point or often called the equilibrium or desired state) and maintaining it there despite disturbances or changes in the system. In this context, our controller (NMPC) above is designed to achieve the point that we set or the desired state.

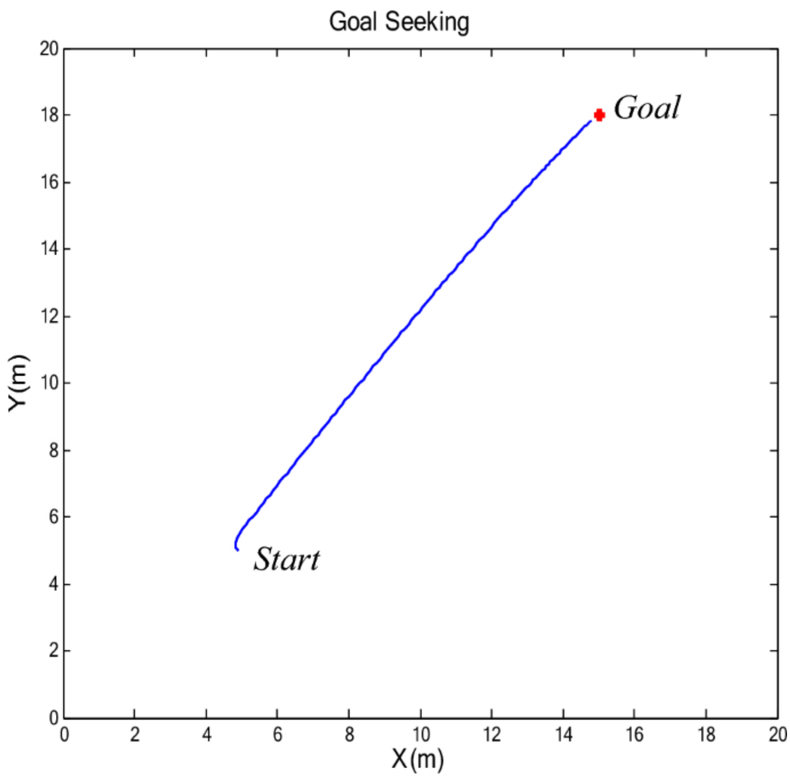

Figure 3.4. Goal Point Following for Mecanum Wheel

To implement it into our controller we can following the function:

$$\begin{bmatrix} x_r(t) \\ y_r(t) \\ \theta_r(t) \end{bmatrix} = \begin{bmatrix} x_c(t) \\ y_c(t) \\ \theta_c(t) \end{bmatrix} \quad \textbf{(Eq. 3.11.)}$$

- $x_r(t), y_r(t), \theta_r(t)$ represent the reference trajectory or set-point for the system state variables x, y, θ respectively.
- $x_c(t), y_c(t), \theta_c(t)$ which is generated by the controller or specified by the system designer based on the desired behavior of the system.

4. SIMULATION

4.1. Block Control System

This block representation of a Nonlinear Model Predictive Control (NMPC) system, various components work together to achieve control objectives. In the other hand, we can implement the System Model with NMPC in real application by following to this block control system.

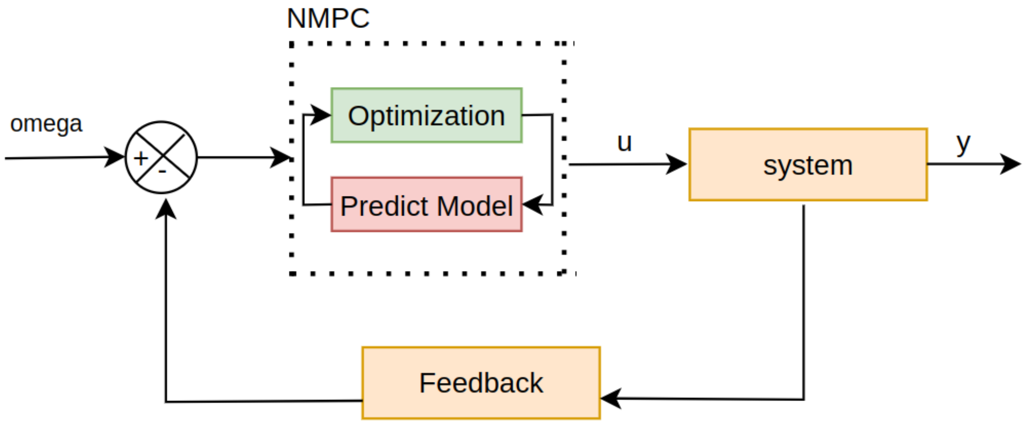

Figure 4.1. Block Control System

4.2. Simulation Result

In this simulation part, we are simulating by applying the Nonlinear Model Predictive Control with Mecanum Wheel which adjusted:

- N-Predict horizon = 30 at time-step = 0.1
- For Goal State x = 10, y = 0.0, $\theta = \dfrac{\pi}{2}$

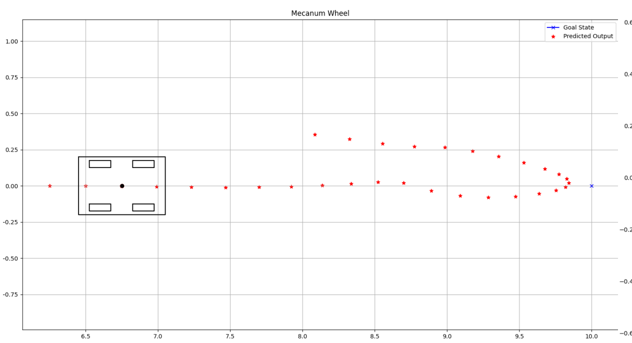

Figure 4.2. Robot in Start Position

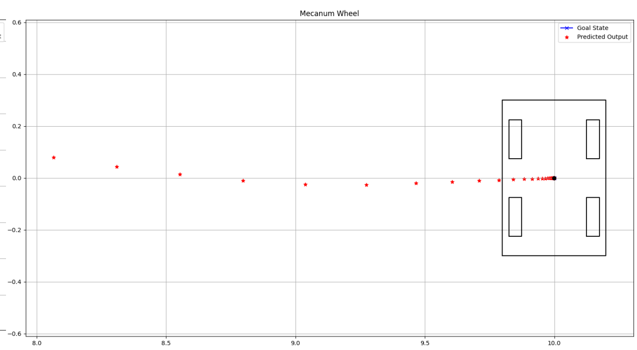

Figure 4.3. Robot reached the Goal-state

5. Conclusion and Future work

5.1. Conclusion

In conclusion, Non-linear Model Predictive Control (NMPC) is a powerful method for motion planning in mecanum wheeled robots. Its ability to optimize control actions in real-time, handle constraints, and adapt to changes makes it a valuable tool. After implement NMPC for mecanum wheel motion planning by using a kinematic model in python code, the simulation shows the robot precisely following a predefined point, showcasing the practicality and effectiveness of NMPC in robotics.

5.2. Future Work

For the future step we consider to improve the NMPC solution and implementing on real application through ROS2 frame work with Trajectory Tracking. Especially, we will research more about the state estimate algorithm then integrate in with the NMPC. In the rest of the goal we plan to integrate the realsense camera for capturing the color and determine the distance from robot to the ball and publish the data to the robot and the robot will move automatically.

REFERENCE

- Blog AuToBarge/model-predictive-control.
- https://www.do-mpc.com/en/latest/theory_mpc.html
- https://courses.cs.washington.edu/courses/cse571/16au/slides/Ferguson et al 2008-Journal of Field Robotics.pdf
- C.E. García et al. Model predictive control: theory and practice—a survey