Toronto Metropolitan University

Ph.D. Interview Project

Trajectory Planning and Control of a Non-holonomic Platform

Author:
Ali Alaei

Prospective Supervisor:
Dr. Farrokh Janabi-Sharifi

Robotics, Mechatronics, and Automation Laboratory (RMAL)

May 18, 2024

Contents

List of Figures

Chapter 1

Trajectory Planning and Control of a Non-holonomic Differential Wheeled Robot

1.1 Abstract

This project focuses on the design and implementation of a control system for a non-holonomic autonomous platform, specifically a Differential Wheeled Robot. Initially, the kinematic model of the robot was derived to understand its movement constraints and capabilities. Subsequently, a Nonlinear Model Predictive Control (NMPC) algorithm was implemented to address both regulation and trajectory tracking problems in environments free of obstacles as well as those with obstacles. The control algorithm was designed to operate effectively under constraints on control inputs. The performance of the proposed control system was evaluated through simulations in MATLAB. The results demonstrate that the NMPC algorithm successfully manages the robot's trajectory and regulation tasks, exhibiting both effectiveness and accuracy in various scenarios.

1.2 Derivation of Kinematic Equations for the Differential Wheeled Robot [1, 2]

To determine the position of the robot on a plane, we need to establish a relationship between the global reference frame of the plane and the robot's local reference frame, as illustrated in Figure 1.1. The axes X_I and Y_I form an arbitrary inertial basis on the plane, defining the global reference frame from a given origin O. To specify the robot's position, we select a point P on the robot's chassis as its reference point. The local reference frame of the robot is defined by the axes X_R and Y_R, which are oriented relative to point P at the midpoint of the wheels' axle. The coordinates x and y represent the position of point P in the global reference frame, and the angle θ denotes the angular orientation between the global and local reference frames. The robot's pose can thus be described by a vector comprising these three elements. The subscript I is used to indicate that this pose is referenced to the global frame:

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{1.1}$$

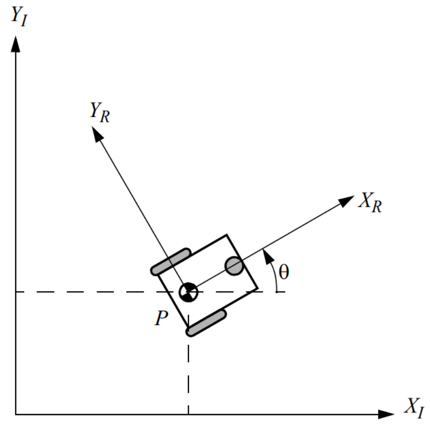

FIGURE 1.1: The global reference frame and the robot local reference frame

Moreover, we can write:

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{1.2}$$

where v is the velocity of the robot along X_R and ω is the angular velocity of the robot body. We can convert the equation (1.2) into matrix form:

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{1.3}$$

Now we can relate the linear and angular velocities of the robot body to the angular velocity of the wheels by the following equation:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{d} & -\frac{r}{d} \end{bmatrix} \begin{bmatrix} \dot{\phi}_r \\ \dot{\phi}_l \end{bmatrix} \tag{1.4}$$

where $\dot{\phi}_r$ and $\dot{\phi}_l$ represent the angular velocities of the right and left wheels, respectively. r denotes the radius of the robot wheels, and d signifies the distance between the robot wheels. By combining equation (1.3) and equation (1.4), we can obtain the kinematic equations of the robot as follows:

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\theta & \frac{r}{2}\cos\theta \\ \frac{r}{2}\sin\theta & \frac{r}{2}\sin\theta \\ \frac{r}{d} & -\frac{r}{d} \end{bmatrix} \begin{bmatrix} \dot{\phi}_r \\ \dot{\phi}_l \end{bmatrix} \tag{1.5}$$

Moreover, the non-holonomic constraint of the robot can be derived from the following equation:

$$\dot{Y}_R = 0 \rightarrow \dot{Y}_I\cos\theta - \dot{X}_I\sin\theta = 0$$

This equation implies that the robot cannot move sideways.

1.3 Nonlinear Model Predictive Control (NMPC) Strategy

Nonlinear Model Predictive Control (NMPC) is a powerful control strategy utilized in various fields, including robotics, process control, and autonomous systems. Unlike traditional control methods, NMPC employs a predictive approach, where a model of the system is used to predict future states and optimize control actions over a finite prediction horizon.
NMPC is a sophisticated control strategy widely utilized in diverse fields due to several key advantages:
Firstly, it can effectively handle systems with nonlinear equations, making it suitable for complex systems where linear control methods may be inadequate. Secondly, NMPC inherently incorporates constraints on system states and control inputs, ensuring that the control actions remain within safe operating limits. This feature is particularly useful in applications where strict constraints must be adhered to, such as in robotics. Furthermore, it formulates control problems as optimization tasks, allowing for the optimization of various objectives, including trajectory tracking, energy efficiency, and obstacle avoidance. Moreover, NMPC is adaptable to changes in system dynamics or operating conditions, as the control strategy continuously updates predictions and control actions based on real-time feedback. Overall, this approach offers a versatile control strategy suitable for a wide range of dynamic systems, enabling precise control while satisfying constraints and optimizing system performance.
In this section, a MPC-based kinematic controller is designed to ensure that the mobile robot can be driven to the desired position accurately and smoothly [3]. Consider $X = [x, y, \theta]^T$ as the state of the mobile robot. We can discretize the equation (1.3) according to the following equation:

$$X(k+1) = X(k) + T_s B(k) u(k) \tag{1.6}$$

where T_s is the sampling time, $u(k) = [v, \omega]^T$ is the control input, and $B(k)$:

$$B(k) = \begin{bmatrix} \cos\theta(k) & 0 \\ \sin\theta(k) & 0 \\ 0 & 1 \end{bmatrix}$$

is determined dynamically by the orientation of the robot. Based on the state space model (1.6), the MPC algorithm is designed to control the system. To obtain the optimal control sequence over the prediction horizon, a cost function is required. The cost function for the MPC can be defined as:

$$\begin{aligned} J(t) = &\sum_{i=1}^{N_p} (X(t+i) - X_{ref}(t+i))^T Q (X(t+i) - X_{ref}(t+i)) + \\ &\sum_{i=0}^{N_c-1} u^T(t+i) R u(t+i) \end{aligned} \tag{1.7}$$

where N_p and N_c are the prediction horizon and control horizon respectively,
$Q \in \mathbb{R}^3$, and $R \in \mathbb{R}^2$ are appropriate weighting matrices, and X_{ref} is desired state vector. Through solving the following finite-horizon optimal control problem online, the optimal control sequence can be obtained as follows:

$$u^* = \arg\min_u \{J(t)\} \tag{1.8}$$

Since the linear and angular velocities of the robot are typically limited by the motors, the control input $u(k)$ should be constrained within an upper and lower bound. Consequently, the following constraint should be imposed on the system:

$$u_{min} \leq u(k) \leq u_{max} \tag{1.9}$$

Additionally, if the environment contains obstacles with positions denoted as X_{ob}, the following constraint should be added to the problem:

$$\|X(k) - X_{ob}\| \leq R_{min} \tag{1.10}$$

Where R_{min} is the minimum admissible distance between the robot and the obstacle.

1.4 Simulation Results

In this section, simulations are performed in MATLAB to verify the accuracy and effectiveness of the NMPC. In the following simulations, the range of control inputs is set to $[-5, 5]$ for both v and ω, and the initial points are considered to be the origin. The prediction horizon for regulation problems is $N_p = 10$ and for trajectory tracking problems is $N_p = 35$. Moreover, the weighting matrices Q and R are identity matrices. The 'fmincon()' function is chosen to solve the nonlinear optimization problem with the optimization variable $u(k)$. In all simulations, the prediction and control horizons are considered equal. At each step, the optimization solver calculates the optimal control sequence. However, based on the receding horizon strategy, only the first element of the control input sequence is fed to the system.

1.4.1 Obstacle-Free Environment

The first simulation corresponds to a regulation problem, where the goal is to drive the robot to a specific target point. The goal point is set to $X_{ref} = [3, 5]^T$. Figures 1.2 and 1.3 illustrate that the robot successfully reaches its goal point while adhering to the control input constraints. The results demonstrate good performance, showcasing the effectiveness of the NMPC in regulating the robot to the desired position. The second simulation addresses a trajectory tracking problem. The desired trajectory is a circle with a radius of 2 meters, defined by the following equations:

$$x_d(t) = 2\cos t$$
$$y_d(t) = 2\sin t$$

$$X_d(t) = \begin{bmatrix} x_d(t) \\ y_d(t) \end{bmatrix}$$

Figures 1.4 and 1.5 demonstrate that the mobile robot can accurately track its desired trajectory while adhering to the constraints on its control inputs. These results highlight the NMPC's capability to maintain precise trajectory tracking under specified input limitations.

1.4.2 Environment with Obstacles

The third and fourth simulations are similar to the previous ones but with the addition of the constraint (1.10) to the optimization problem to account for obstacles. The third simulation addresses a regulation problem. The goal is for the robot to

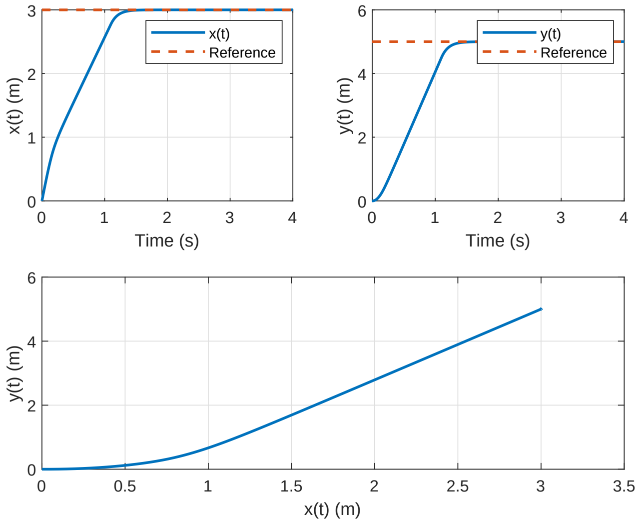

FIGURE 1.2: States and Trajectory of the Mobile Robot for the Regulation Problem in an Obstacle-Free Environment

reach the point $X_{ref} = [30, 30]^T$. Figures 1.6 and 1.7 show that the robot successfully reaches its reference point while adhering to control input constraints and avoiding collisions with two obstacles in its path. The fourth simulation focuses on a trajectory tracking problem. The desired trajectory is the same circular path as in the obstacle-free environment. Figures 1.8 and 1.9 demonstrate that the mobile robot can accurately track its desired circular trajectory while avoiding collisions with two obstacles placed on the trajectory. These results confirm the effectiveness of the NMPC in maintaining precise trajectory tracking and regulation in environments with obstacles.

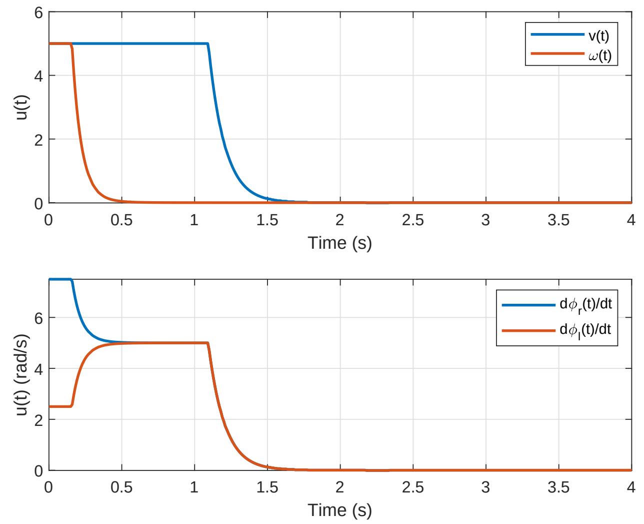

FIGURE 1.3: Control Inputs of the Mobile Robot for the Regulation Problem in an Obstacle-Free Environment

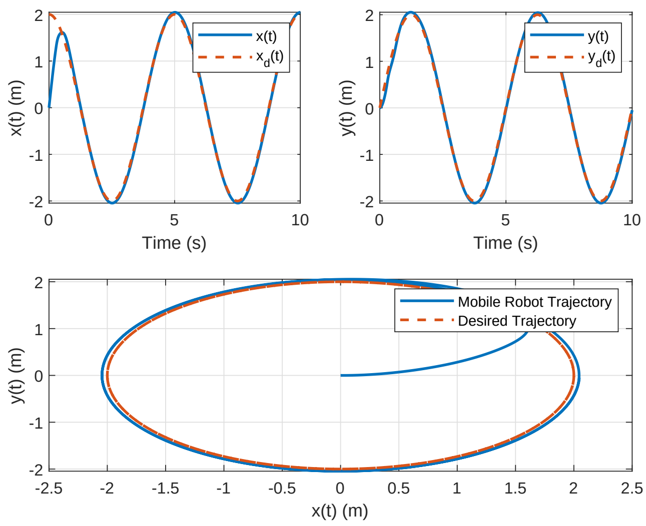

FIGURE 1.4: States and Trajectory of the Mobile Robot for the Trajectory Tracking Problem in an Obstacle-Free Environment

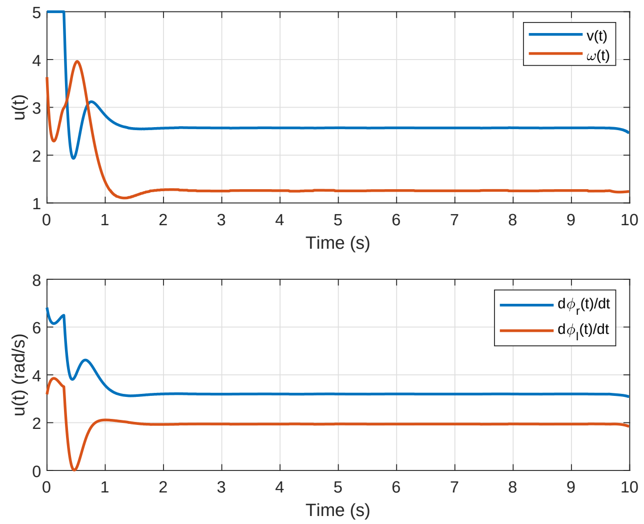

FIGURE 1.5: Control Inputs of the Mobile Robot for the Trajectory Tracking Problem in an Obstacle-Free Environment

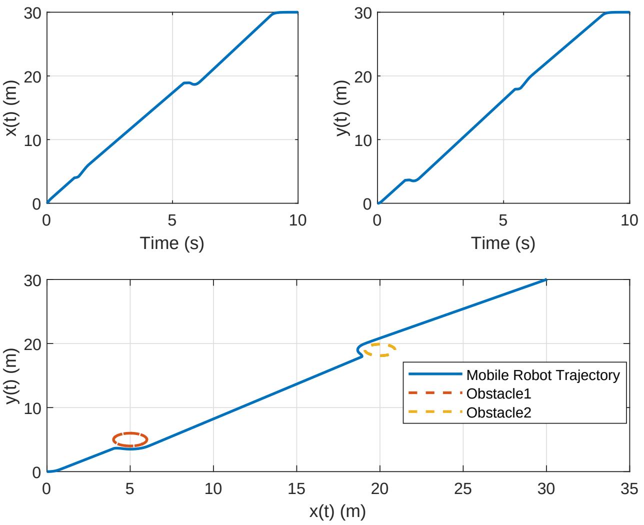

FIGURE 1.6: States and Trajectory of the Mobile Robot for the Regulation Problem in an Environment with Obstacles

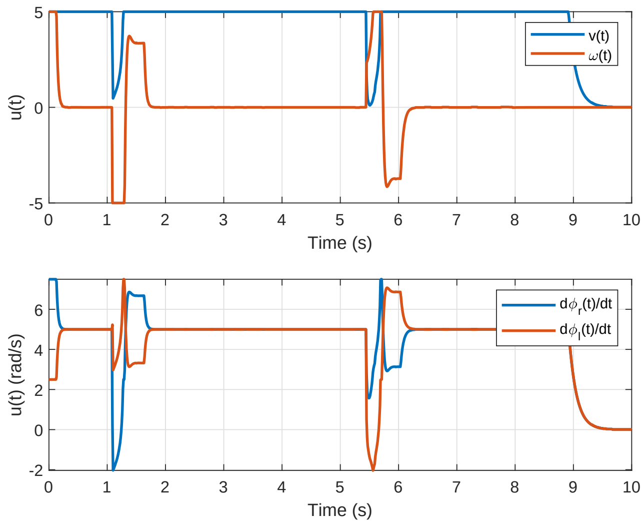

FIGURE 1.7: Control Inputs of the Mobile Robot for the Regulation Problem in an Environment with Obstacles

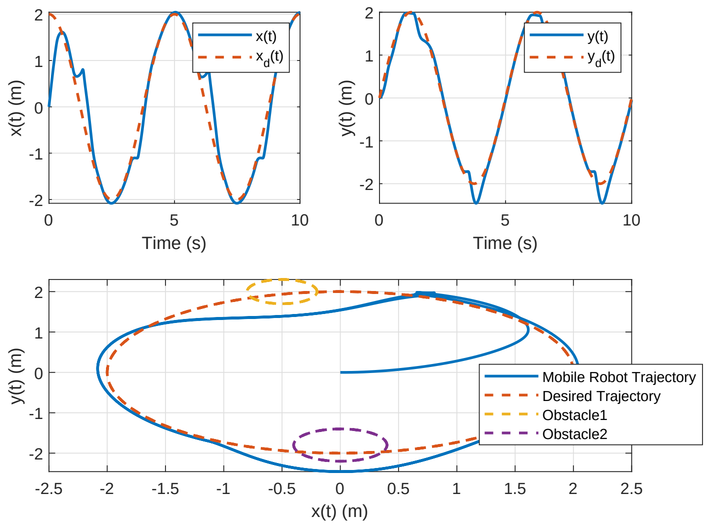

FIGURE 1.8: States and Trajectory of the Mobile Robot for the Trajectory Tracking Problem in an Environment with Obstacles

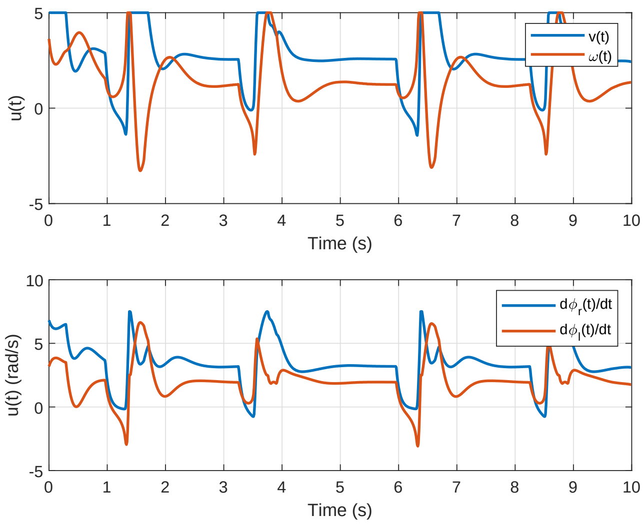

FIGURE 1.9: Control Inputs of the Mobile Robot for the Trajectory Tracking Problem in an Environment with Obstacles

Bibliography

[1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[2] N. Correll, B. Hayes, C. Heckman, and A. Roncone, *Introduction to autonomous robots: mechanisms, sensors, actuators, and algorithms*. Mit Press, 2022.

[3] C. Wang, X. Liu, X. Yang, F. Hu, A. Jiang, and C. Yang, "Trajectory tracking of an omni-directional wheeled mobile robot using a model predictive control strategy," *Applied Sciences*, vol. 8, no. 2, p. 231, 2018.