VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY

RBE3047 - REPORT
ELEMENTS OF INTELLIGENT ROBOT

# DESIGN THE NONLINEAR LYAPUNOV-BASED CONTROLLER AND EXTENDED KALMAN FILTER FOR THE TRICYCLE MOBILE ROBOT

Students group:

18020936 - Bui Duy Nam

18020944 - Duong Thi Thuy Ngan

Supervisor: PhD. Phung Manh Duong

HANOI - 2022

# Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Problems

A mobile robot, is a robot that is capable of moving in the surrounding. The mobile robotics is usually considered to be a subfield of robotics and information engineering. Mobile robots have the capability to move around in their environment and are not fixed to one physical location. Mobile robots can be "autonomous" which means they are capable of navigating an uncontrolled environment without the need for physical or guidance devices.

The main problem in the autonomous mobile robot is the navigation and localization capacity. In this report, we present the Tricycle Mobile Robot and the problems about this type robot model. The Tricycle mobile robot is the popular robot type in the world. Besides, we present the method to navigate the robot track to the desired trajectory and estimate the robot location in the actual.

## 1.2 Report layout

The report includes 4 parts:

**Chapter 1: Introduction.** This chapter presents the overview of the report, about the Trycicle model and the basic problems in control and localization for Robot. This chapter also present the techniques used in the report.

**Chapter 2: The Tricycle mobile robot modeling.** This chapter presents the Tricycle Mobile robot and the methods of controlling this model. Besides, this chapter also present the simulation of the model in MATLAB Simulink and some test case before control this robot model.

**Chapter 3: Trajectory tracking using Nonlinear Lyapunov-based controller.** This chapter presents the open loop controller and the closed-loop controller in control the Tricycle mobile robot. The main focus is the Nonlinear Lyapunov-base controller. This chapter also present the simulation result of both open loop and closed-loop control in the ideal condition and noise condition.

**Chapter 4: Robot localization using Extended Kalman filter.** This chapter presents Extended Kalman filter for the Tricycle in localization problems. In addition, the results simulation are shown to prove the performance of this localization method in the practical

# Chapter 2

# The Tricycle mobile robot modeling

In this chapter, we introduce the Tricycle-like mobile robot and the types of mechanisms used to control a tricycle. Besides, we present the kinematic model and simulate it by using MATLAB Simulink.

The tricycle model normally has one front wheel and two rear wheels. Tricycle mobile robots are of widely used due to their extremely simple design, structural sustainability of and simplicity of control actions.

## 2.1 The kinematic model

In this section, we present two different type of control a Tricycle mobile robot. In all of these cases, we assume that $\mathbf{x} = (x, y, \theta)^T$ is the position of the robot where $x$, $y$ present the midpoint of two rear wheels in the global frame and the heading angle $\theta$ is presented by the angle between wheelbase $d$ and the x-axis. The control signal includes two elements: the longitudinal velocity $v$ and the steering angle $\gamma$. In actual, the Tricycle robot has constraints, they are the limited constraints based on its geometry and defined as:

$$\begin{aligned} |v| &\leq v_{max} \\ |\gamma| &\leq \gamma_{max} \end{aligned} \tag{2.1}$$

where $a$ denotes the longitudinal acceleration and $\phi$ denotes the steering angle rate of a Tricycle.

### 2.1.1 The Type-1 method

The first type uses the front wheel for both the steering and driving actions. The kinematic model of Type-1 method is shown in Fig.2.1:

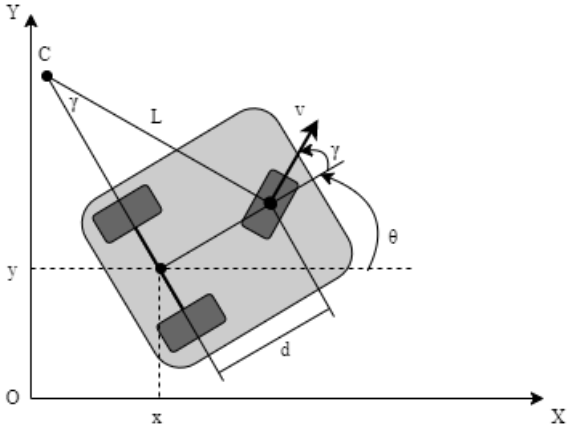



Figure 2.1: The Type-1 control method of a Tricycle

According to the Fig.2.1, $L$ denotes the distance from the Instantaneous Center of Curvature ($C$) to the front wheel. We defined the $\mathbf{x}_f = (x_f, y_f, \theta_f)^T$ presented the position of the front wheel as:

$$\begin{aligned} x_f &= x + d\cos\theta \\ y_f &= y + d\sin\theta \\ \theta_f &= \gamma + \theta \end{aligned} \tag{2.2}$$

The kinematic equations of the Type-1 are shown as follows:

$$\begin{aligned} \dot{x} &= \dot{x}_f + \dot{\theta} d\sin\theta = v\cos(\gamma+\theta) + \frac{v}{d}\sin\gamma.d\sin\theta = v\cos\gamma\cos\theta \\ \dot{y} &= \dot{y}_f - \dot{\theta} d\cos\theta = v\sin(\gamma+\theta) - \frac{v}{d}\sin\gamma.d\cos\theta = v\cos\gamma\sin\theta \\ \dot{\theta} &= \dot{\theta}_f = \frac{v}{L} = \frac{v}{d/\sin\gamma} = \frac{v}{d}\sin\gamma \end{aligned} \tag{2.3}$$

### 2.1.2 The Type-2 method

In the second type, the front wheel used for the steering angle driving and two rear wheel used for longitudinal control. The kinematic model of Type-2 method is shown in Fig.2.2:

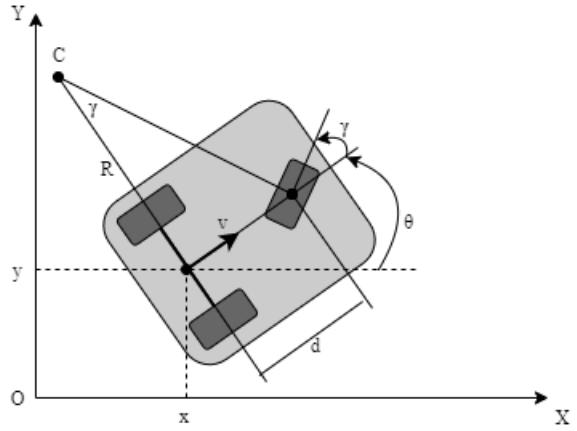



Figure 2.2: The Type-2 control method of a Tricycle

We denotes $R$ is the turning radius of a Tricycle. According to the Fig.2.2, the kinematic equations of the robot are shown as follows:

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \frac{v}{R} = \frac{v}{d/\tan\gamma} = \frac{v}{d}\tan\gamma \end{aligned} \tag{2.4}$$

## 2.2 The MATLAB Simulation

In this section, we design the MATLAB Simulink model to show the performance of all type control method with difference case study. The Tricycle mobile robot has the constraints:

$$\begin{aligned} |v| &\leq 1.68\ m/s \\ |\gamma| &\leq \pi/4\ rad \end{aligned} \tag{2.5}$$

The MATLAB Simulink model of the Tricycle robot are shown in the Fig.2.3. The system has a control input $\boldsymbol{u} = (v, \gamma)^T$ and output is the current position of the robot in the global frame. We assume that the initial position of the robot is $\boldsymbol{x}_0 = (0, 0, 0)^T$. The *TricycleModel* block is the kinematic of the model and the *Normalization* block always keep the heading angle of the robot between $[-\pi, \pi]$. For the simulation, the total time is $10s$ and the time step is $0.1s$.

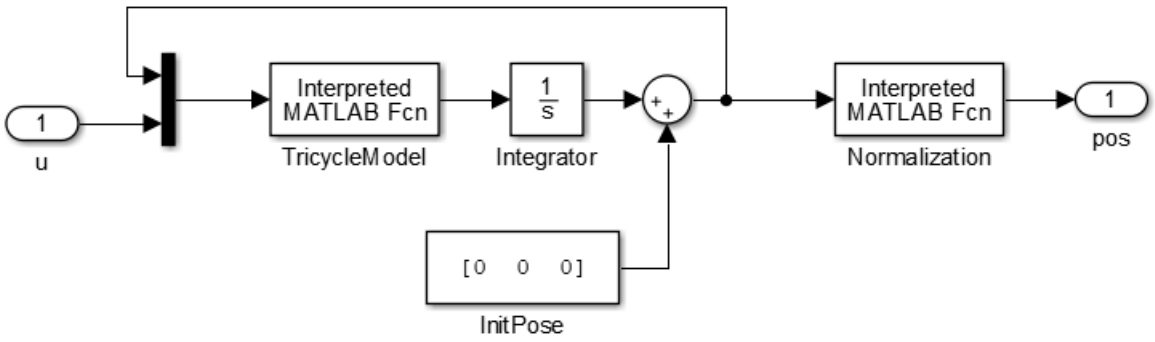



Figure 2.3: The Tricycle model in Simulink

We perform operation simulation of both types of robots as follows. Firstly, in Test1, we set fixed longitudinal velocity and steering angle as:

$$\begin{aligned} v &= 1.0\ m/s \\ \gamma &= \pi/6\ rad \end{aligned} \tag{2.6}$$

The simulation diagram in Test 1 is shown in the Fig.2.4

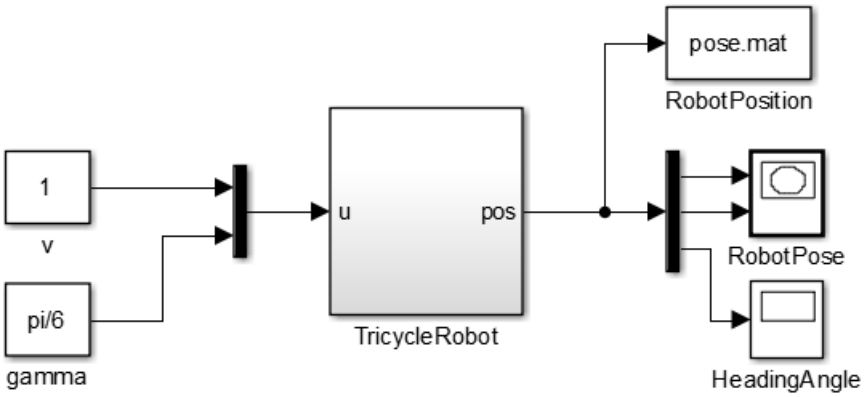



Figure 2.4: The simulation diagram in Test 1

The results satisfied of both types with the fixed velocity and steering angle are shown in the Fig.2.5 - 2.6.

According to Fig.2.5-2.6, we can see that the path of Type-2 robot is moving quickly than Type-1 one. And the heading angle is the same (Fig.2.6b). Because the heading angle are limited in range $[-\pi; \pi]$, the heading angle will be revalue when get outside the range.

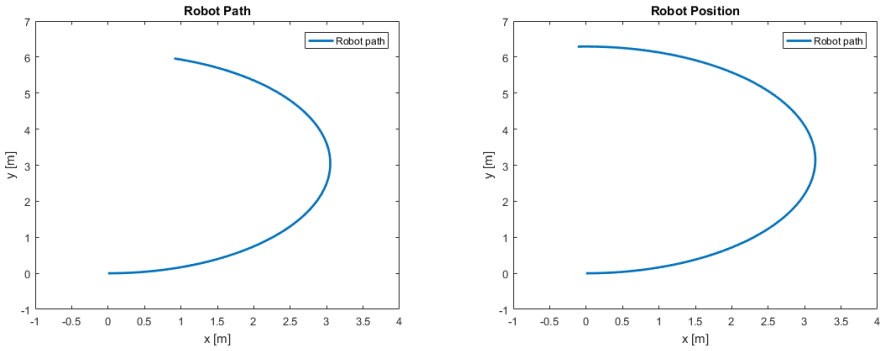
Next, in Test 2, we simulate the robot moved at a fixed speed but had a steering angle that varies linearly with time. In this test, we set:

$$\begin{aligned} v &= 1.0\ m/s \\ \gamma &= 0.5t\ rad/s \end{aligned} \tag{2.7}$$

The simulation diagram of Test 2 is shown in the Fig.2.7. where $v$ is the constant value and $\gamma$ is varies linearly with time, thus, we used *Ramp* block. Besides, because of the limited steering angle, we added the *Saturation* block to limit the signal.

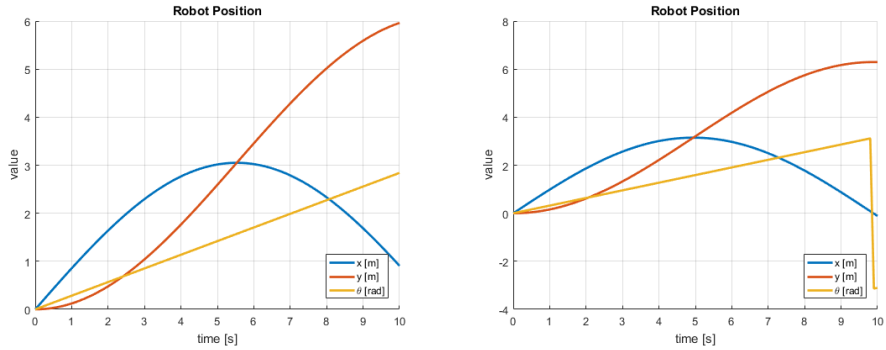
The control input and results in Test 2 are shown in Fig.2.8-2.9-2.10.

In Fig.2.8, steering angle varying linear time, when it's value reach the max steering angle, the angle will be no change at the max value. As well as Test 1, the heading angle of the robot using




(a) The Type-1 robot (b) The Type-2 robot

Figure 2.5: The actual path of both types of the Tricycle in Test 1




(a) The Type-1 robot (b) The Type-2 robot

Figure 2.6: The actual position value of both types of the Tricycle in Test 1

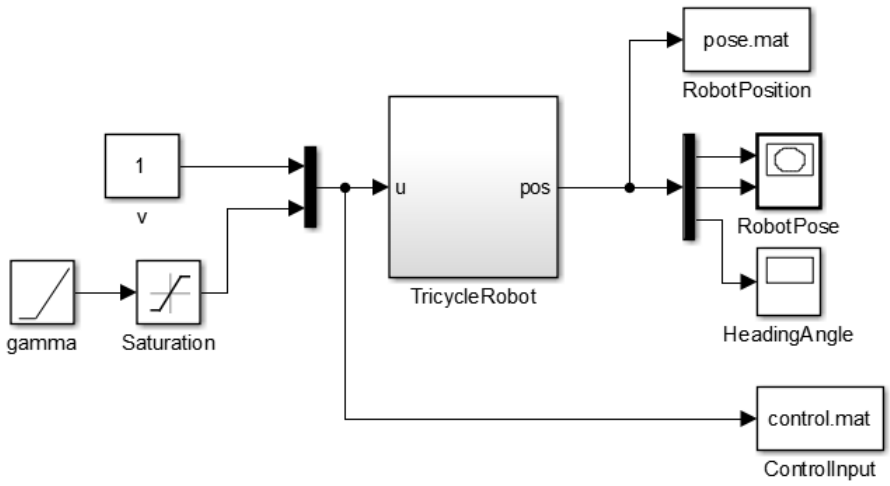



Figure 2.7: The simulation diagram in Test 2

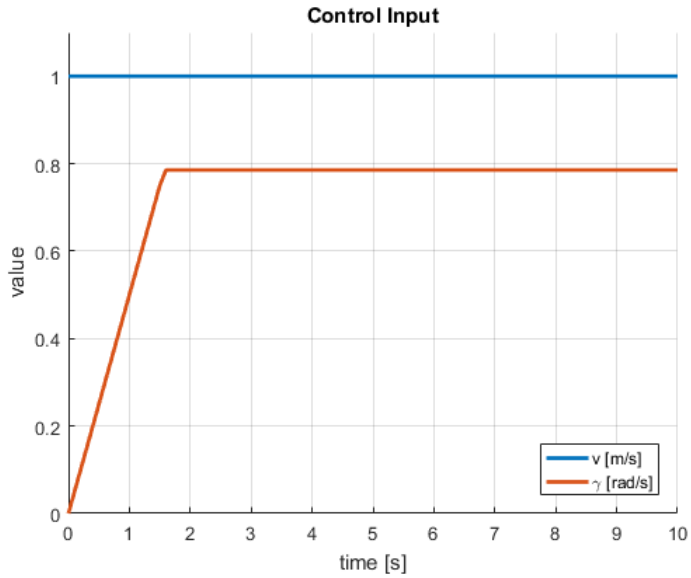
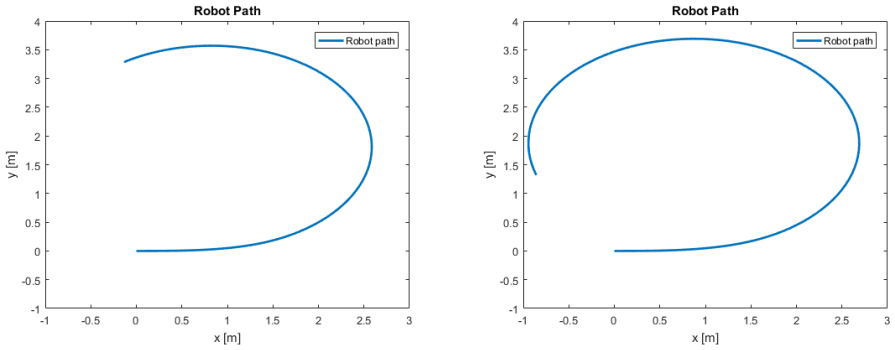
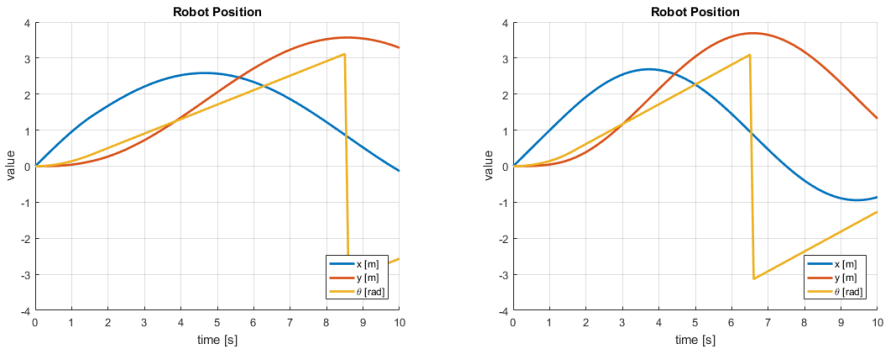



Figure 2.8: The control input of both types robot in Test 2




(a) The Type-1 robot (b) The Type-2 robot

Figure 2.9: The actual robot path of both types of the Tricycle in Test 2




(a) The Type-1 robot (b) The Type-2 robot

Figure 2.10: The actual position value of both types of the Tricycle in Test 2

Type-2 always turn quickly than Type-1. Although the shape of the path in both type of robot are the same.

Finally, in Test 3, we simulate the robot moved with both velocity and steering angle varying linearly with time. In this test, we set:

$$\begin{aligned} v &= 0.2t \ m/s \\ \gamma &= 0.5t \ rad/s \end{aligned} \tag{2.8}$$

The simulation diagram of Test 3 is shown in the Fig.2.11. Because of the linear varying time and limited velocity and steering angle, we add Two *Ramp* block for linear varying time and *Saturation* block to limit the value in both of them.

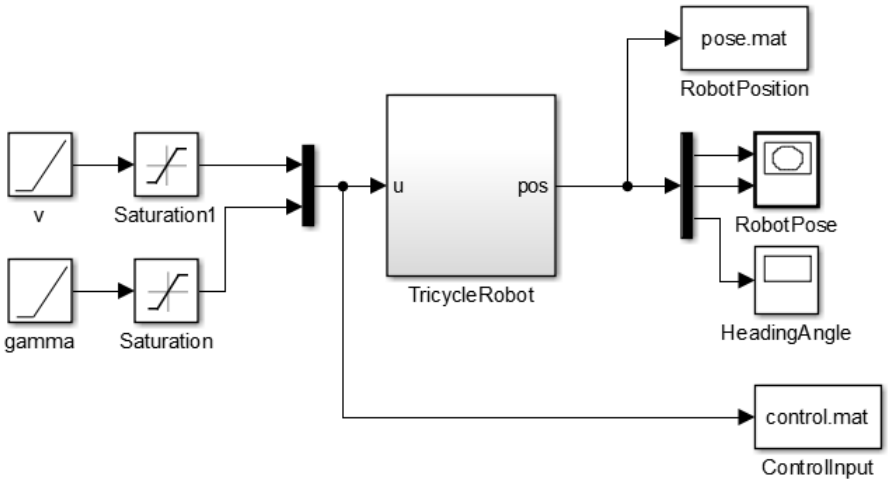



Figure 2.11: The simulation diagram in Test 3

The control input and results in Test 3 are shown in Fig.2.12-2.13-2.14.

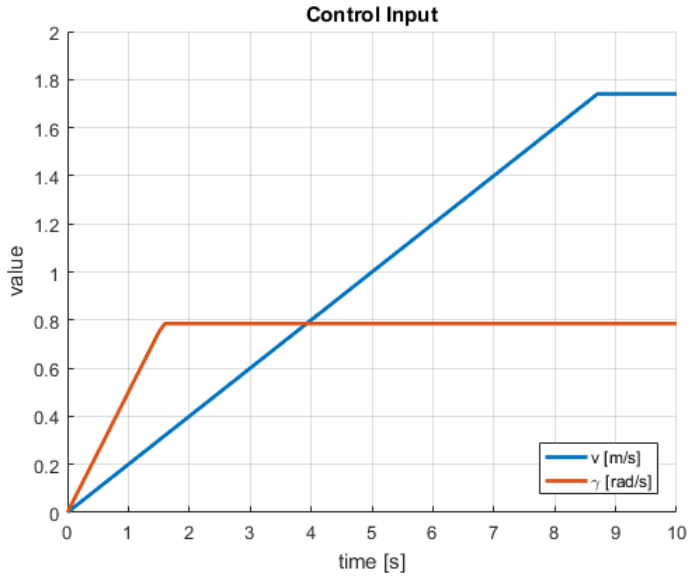
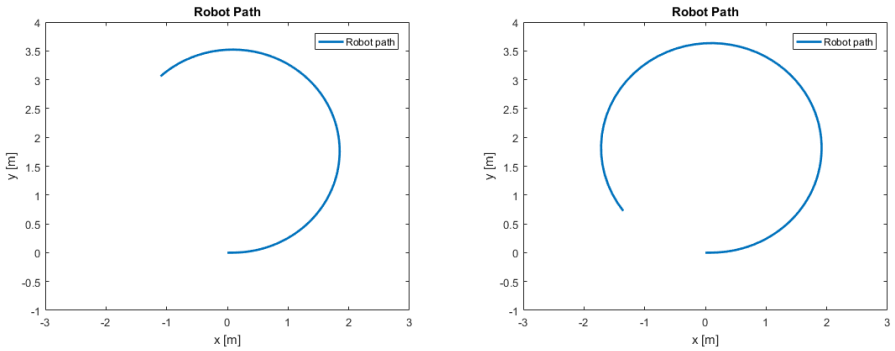


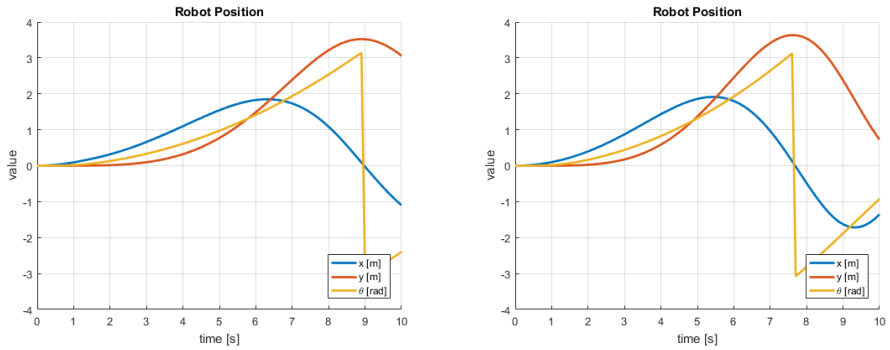

Figure 2.12: The control input of both types robot in Test 3

According to the Fig.2.12, at the begin of the motion, the velocity and steering angle are linear varying time until they reach the max value. In Fig.2.13-2.14, when the the steering angle are large, the heading angle of Type-2 robot are changed quickly.




(a) The Type-1 robot (b) The Type-2 robot

Figure 2.13: The actual robot path of both types of the Tricycle in Test 3




(a) The Type-1 robot (b) The Type-2 robot

Figure 2.14: The actual position value of both types of the Tricycle in Test 3

# Chapter 3

# Trajectory tracking using Nonlinear Lyapunov-based controller

In this chapter, we present the controllers both open loop controller and closed loop controller for a Tricycle mobile robot presented in Section 2.1.2. Besides, we present the simulation results to show the performance of our controller in ideal condition and noise condition.

## 3.1 Design the desired path

In this section, we design the reference path using for the controller presented bellow. The simple path is shown is the Fig.3.1

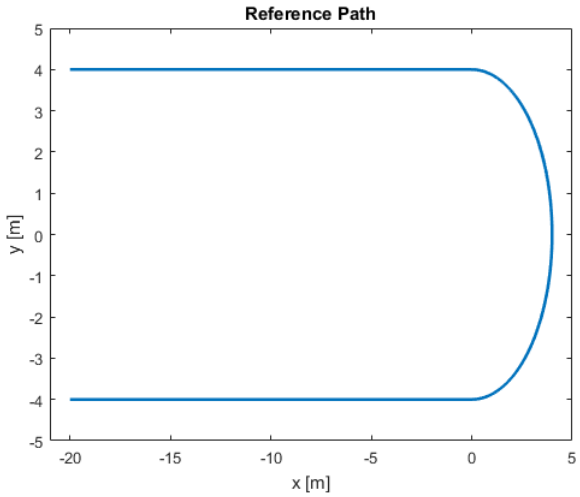



Figure 3.1: The desired trajectory

The reference path includes three parts. The first part is the straight line from $(-20, 4)$ to $(0, 4)$. The second part is the right half circle had center is $(0, 0)$ and the radius is 4. The last part is the straight line form $(0, -4)$ to $(-20, -4)$. We design three parts with different time $20 - 10 - 20$ s. Summary, the path can be written in the equations form as follows:
From $0 \rightarrow 20$ s:

$$\begin{aligned} x &= -20 + t \\ y &= 4 \\ \theta &= 0 \end{aligned} \tag{3.1}$$

From $20 \rightarrow 30$ s:

$$\begin{aligned} x &= 4\sin(\frac{2\pi}{20}(t-20)) \\ y &= 4\cos(\frac{2\pi}{20}(t-20)) \\ \theta &= -\frac{2\pi}{20}(t-20) \end{aligned} \tag{3.2}$$

From $30 \rightarrow 50$ s:

$$\begin{aligned} x &= 30 - t \\ y &= -4 \\ \theta &= -\pi \end{aligned} \tag{3.3}$$

The total time is 50 s. The reference velocity are shown in the Fig.3.2

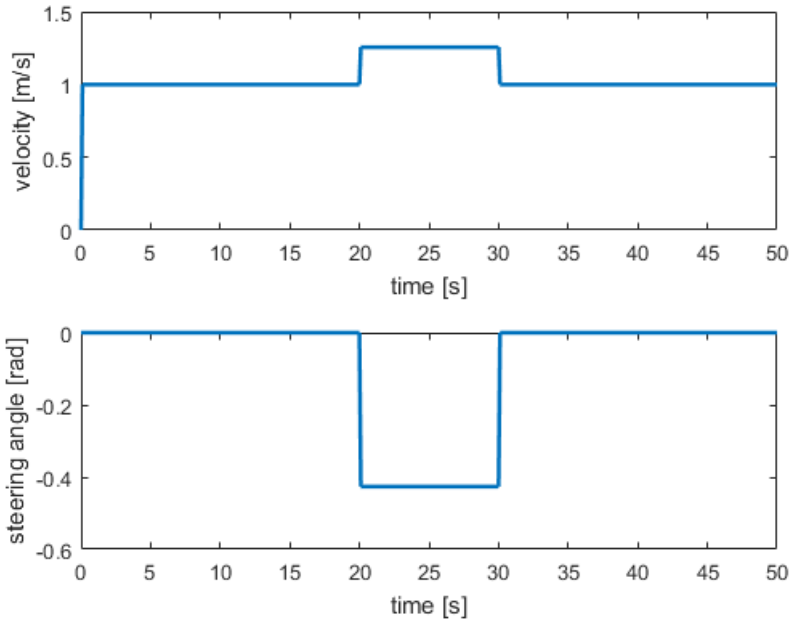



Figure 3.2: The reference control signals

## 3.2 Design the open loop controller

In this subsection, we present the open loop control to control the Tricycle tracked to the desired path. In this type of control, we compute directly the control signal based on the reference path. Therefore, the initial position of the robot is the first position of the reference path.

Assume the reference position is $\mathbf{x}_r = (x_r, y_r, \theta_r)^T$, according to the kinematic equations of Type-2 robot, we get:

$$\begin{aligned} \dot{x}_r &= v_r \cos\theta \\ \dot{y}_r &= v_r \sin\theta \\ \dot{\theta}_r &= \frac{v_r}{d}\tan\gamma_r \end{aligned} \tag{3.4}$$

where $v_r$ and $\gamma_r$ is the reference longitudinal velocity and the reference steering angle, respectively and they can be satisfied by:

$$\begin{aligned} v_r &= \sqrt{\dot{x}_r^2 + \dot{y}_r^2} \\ \gamma_r &= \arctan\frac{d\dot{\theta}_r}{v_r} \end{aligned} \tag{3.5}$$

Thus, we can get the control signals from the reference path are follows:

$$\begin{aligned} v &= v_r \\ \gamma &= \gamma_r \end{aligned} \tag{3.6}$$

## 3.3 Design the closed loop controller - Nonlinear Lyapunov-based controller

According to the Subsection 2.1.2, we have the kinematic model:

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \frac{v}{d}\tan\gamma \end{aligned} \tag{3.7}$$

Defined the reference position $\mathbf{x}_r$ as shown in (3.4) and the reference velocity $(v_r, \gamma)^T$ as shown in (3.5). Based on the geometric relationship, the tracking error can be satisfied as follow:

$$\boldsymbol{x}_e = \begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{3.8}$$

where $x_e$, $y_e$ and $\theta_e$ is the longitudinal position, lateral position and angle errors between reference position and the actual position transformed into the robot frame. Taking the time derivative of (3.8), we get:

$$\begin{aligned} \dot{x}_e &= \dot{\theta}\left(-\sin\theta\left(x_r - x\right) + \cos\theta\left(y_r - y\right)\right) + \cos\theta\left(\dot{x}_r - \dot{x}\right) + \sin\theta\left(\dot{y}_r - \dot{y}\right) \\ &= y_e\dot{\theta} + \cos\theta\left(v_r\cos\theta_r - v\cos\theta\right) + \sin\theta\left(v_r\sin\theta_r - v\sin\theta\right) \\ &= y_e\dot{\theta} + v_r\cos\theta_e - v \\ \dot{y}_e &= \dot{\theta}\left(-\cos\theta\left(x_r - x\right) - \sin\theta\left(y_r - y\right)\right) - \sin\theta\left(\dot{x}_r - x\right) + \cos\theta\left(\dot{y}_r - \dot{y}\right) \\ &= -x_e\dot{\theta} - \sin\theta\left(v_r\cos\theta_r - v\cos\theta\right) + \cos\theta\left(v_r\sin\theta_r - v\sin\theta\right) \\ &= -x_e\dot{\theta} + v_r\sin\theta_e \\ \dot{\theta}_e &= \dot{\theta}_r - \dot{\theta} \end{aligned} \tag{3.9}$$

The (3.9) can be rewritten as:

$$\begin{aligned} \dot{x}_e &= y_e\dot{\theta} + v_r\cos\theta_e - v \\ \dot{y}_e &= -x_e\dot{\theta} + v_r\sin\theta_e \\ \dot{\theta}_e &= \dot{\theta}_r - \dot{\theta} \end{aligned} \tag{3.10}$$

For the tracking problem can be transformed to find the control signal $\mathbf{u} = (u_1, u_2)^T = \left(v, \dot{\theta}\right)^T$ such that $\mathbf{x}_e \to 0$ when $t \to \infty$.

We choose the Lyapunov function candidate as:

$$V = \frac{1}{2}\left(x_e^2 + y_e^2\right) + \frac{1 - \cos\theta_e}{k_2} \tag{3.11}$$

The derivative of $V$ as the follow:

$$\begin{aligned} \dot{V} &= x_e\dot{x}_e + y_e\dot{y}_e + \dot{\theta}_e\frac{\sin\theta_e}{k_2} \\ &= x_e\left(-u_1 + v_r\cos\theta_e + y_e u_2\right) + y_e\left(v_r\sin\theta_e - x_e u_2\right) + \dot{\theta}_e\frac{\sin\theta_e}{k_2} \\ &= x_e\left(-u_1 + v_r\cos\theta_e\right) + y_e v_r\sin\theta_e + \dot{\theta}_e\frac{\sin\theta_e}{k_2} \end{aligned} \tag{3.12}$$

Choose the control signal:

$$\begin{aligned} u_1 &= v_r\cos\theta_e + k_1 x_e \\ u_2 &= \dot{\theta}_r + v_r\left(k_2 y_e + k_3\sin\theta_e\right) \end{aligned} \tag{3.13}$$

Replacing equations (3.13) to $\dot{V}$ in (3.12), we satisfied:

$$\begin{aligned}\dot{V} &= -k_1 x_e^2 + \left(\frac{u_2 - \dot{\theta}_r}{k_2 v_r} - \frac{k_3 \sin\theta_e}{k_2}\right) v_r \sin\theta_e + \dot{\theta}_e \frac{\sin\theta_e}{k_2} \\ &= -k_1 x_e^2 + \frac{u_2 - \dot{\theta}_r}{k_2} \sin\theta_e - \frac{k_3}{k_2} v_r \sin^2\theta_e + \dot{\theta}_e \frac{\sin\theta_e}{k_2} \\ &= -k_1 x_e^2 - \frac{k_3}{k_2} v_r \sin^2\theta_e \leq 0\end{aligned} \tag{3.14}$$

Thus, the stability of the Lyapunov function is satisfied. That lead to the tracking error converge to zero. Therefore, the desired path is tracked.

Because the control input signal of the Tricycle mobile robot is $(v, \gamma)^T$. Therefore, the control signal are follows:

$$\begin{aligned} v &= u_1 \\ \gamma &= \text{atan2}\,(u_2 d, u_1) \end{aligned} \tag{3.15}$$

## 3.4 The simulation results

In this section, we present the tracking simulation results using both open loop and closed loop controller. In addition, an external disturbance is added into the steering angle to evaluate the stability of controllers. Therefore, we can see the performance of the Closed-loop controller. The noise has zero mean and have the variance is 0.09 $rad^2$. The disturbance signal is shown in the Fig.3.3

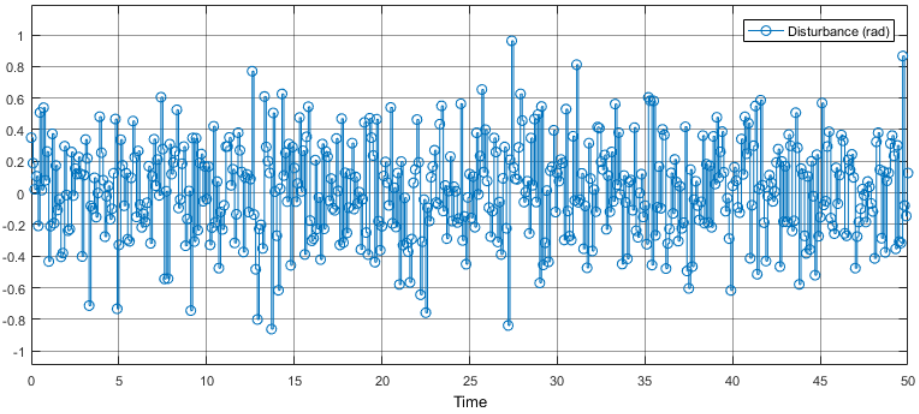



Figure 3.3: The added disturbance

The limited control signals of the system are:

$$\begin{aligned} |v| &\leq 1.74\ m/s \\ |\gamma| &\leq \frac{\pi}{4}\ rad \end{aligned} \tag{3.16}$$

### 3.4.1 The open loop control

In this subsection, we present the simulation results of the open loop controller. The initial position of the controller is $\mathbf{x}_0 = (-20, -4, 0)^T$. The simulation diagram of the controller is shown in Fig.3.4.

The simulation results are shown in Fig.3.5-3.6-3.7.

According to the Fig3.5, the we can that the trajectory of the Open loop control. In case without noise, the path is fit with the reference because the control signal is computed directly form the reference path. The tracked path is more clear in the Fig.3.5b, all the output signal of the robot are fit with the reference. But, when the noise added to the steering angle, the system can not tracked to the reference, the results are shown in the Fig.3.5c-3.5d. Because the control system did not have the feedback from the robot, the control signal can not process the noise. Therefore, the path are not tracked.

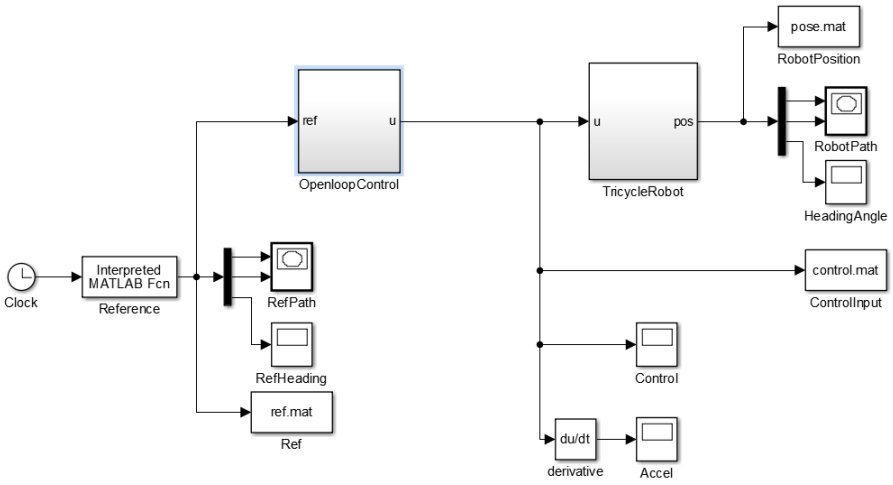
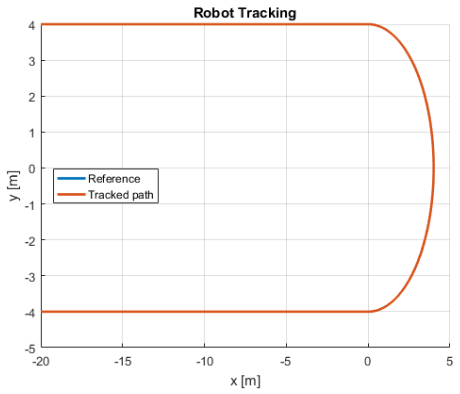
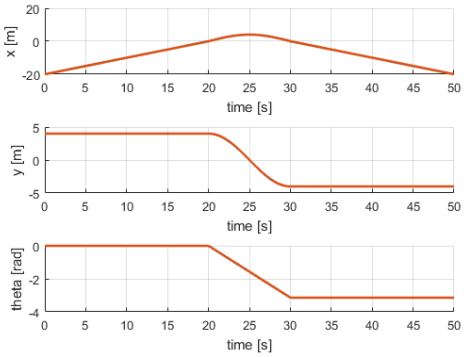


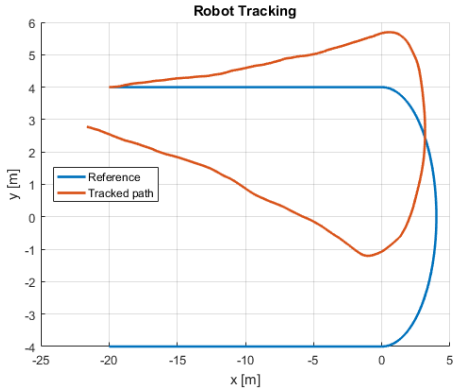

Figure 3.4: The simulation diagram of the Open loop controller



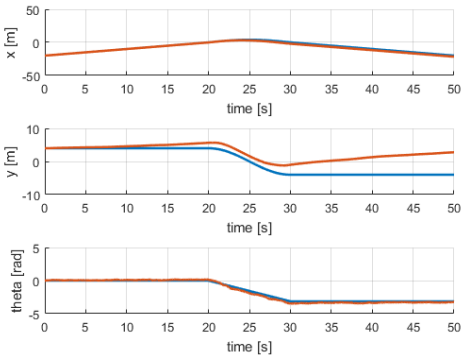

(a) The tracked trajectory without noise




(b) The tracked position without noise

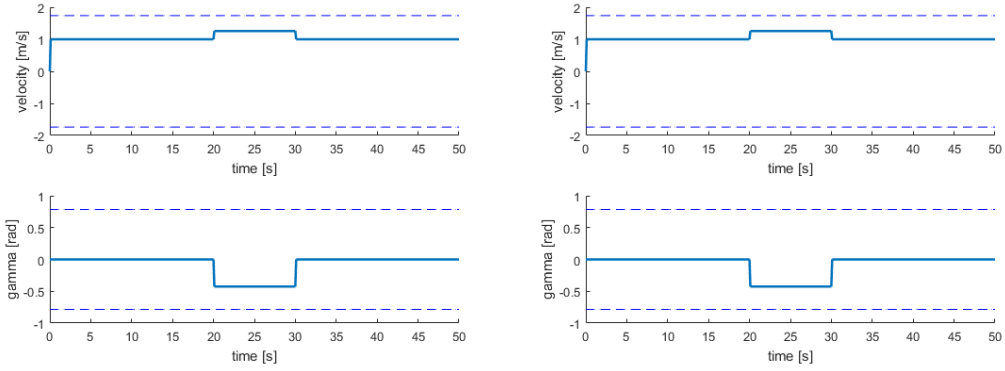



(c) The tracked trajectory with noise




(d) The tracked position with noise

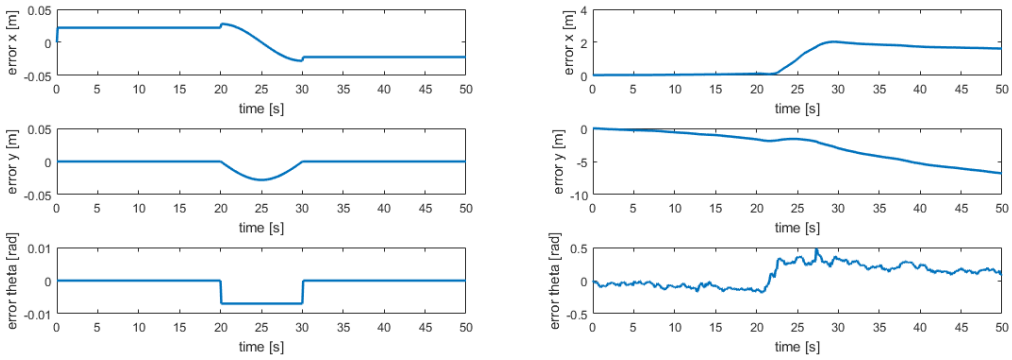
Figure 3.5: The Trajectory tracking using Open loop control




(a) Without noise (b) With noise

Figure 3.6: The control signals of Open loop controller

In the Fig.3.6, the control signal are computed from the reference path and had no change in both cases.




(a) Without noise (b) With noise

Figure 3.7: The tracked error of Open loop controller

The tracked errors can be seen in Fig.3.7. The case without noise, the tracked error is not equal 0 because the control signal is computed directly from the reference via derivative. Therefore the robot always lager than the reference. But in overall, the tracked path is fit with the reference. When the noise is added, we can see that the error increase vary time.

### 3.4.2 The closed-loop control

In this subsection, we present the performance of the Nonlinear Lyapunov-based control in trajectory tracking problem. First, according to Section 3.3, we design the simulation diagram as shown in the Fig.3.8:

According to the Fig.3.8, we group design the control signal in the *LyapunovController* block. This block has two inputs is the current position and reference position and the output signal is the limited control signal $\mathbf{u} = (v, \gamma)^T$.

We defined the initial position of the robot is $\mathbf{x}_0 = (-20, 3, 0)^T$ and the parameters of the control system as given:

$$k_1 = k_2 = k_3 = 0.5 \tag{3.17}$$

The simulation results are shown in Fig.3.9-3.10-3.11

The Fig.3.9 present the performance of the controller in tracking the reference path shown in Fig.3.1 of the Tricycle robot in both cases without noise and with noise. At the beginning of the motion, the robot moves to the desired path and tracks along the trajectory (Fig.3.9a-3.9c). The

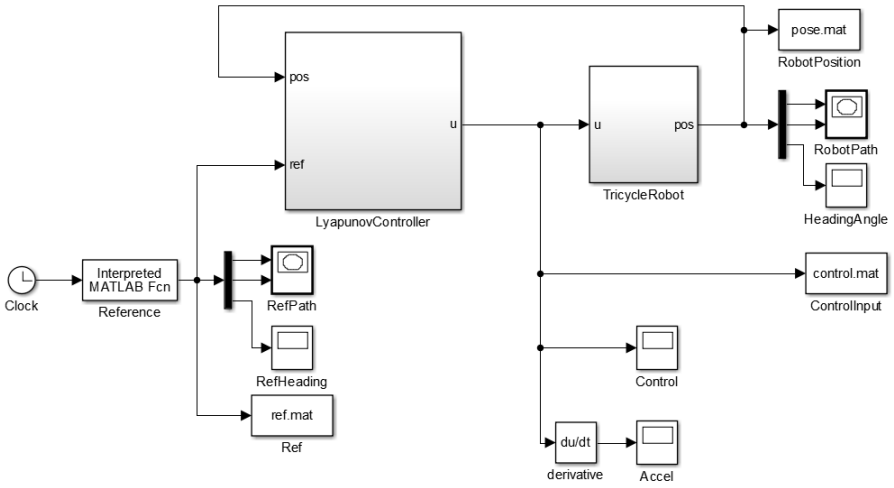
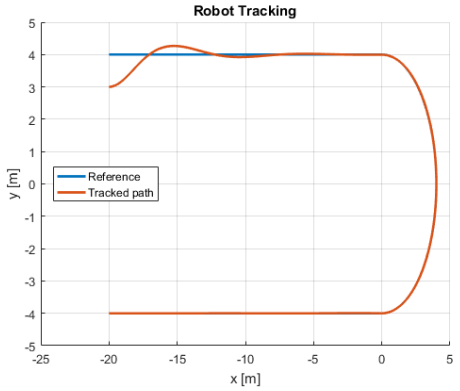
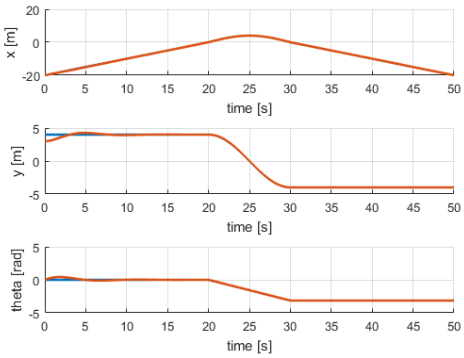


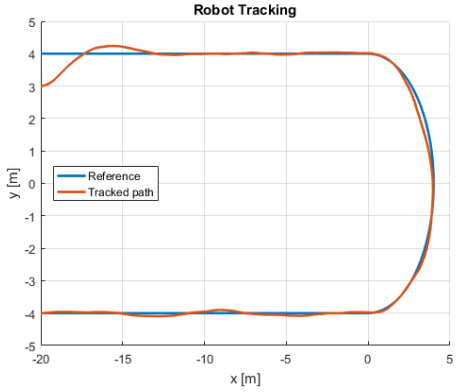

Figure 3.8: The simulation diagram of the Lyapunov controller



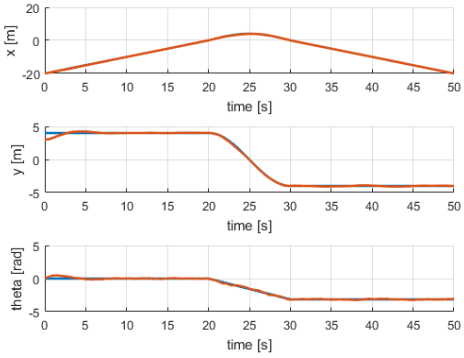

(a) The tracked trajectory without noise




(b) The tracked position without noise




(c) The tracked trajectory with noise

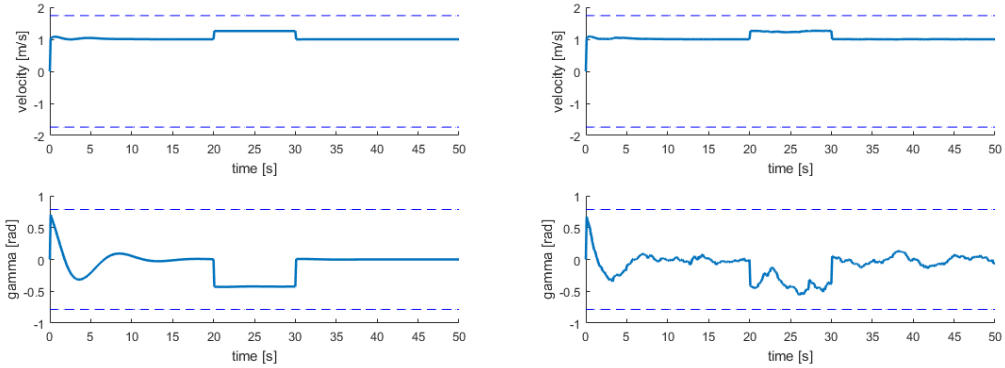



(d) The tracked position with noise

Figure 3.9: The Trajectory tracking using Nonlinear Lyapunov control

system response is fast and stable even when the robot is not on the desired path. When the noise added to the control signal (Fig.3.9c), the tracked path oscillate around the trajectory with the small error but the robot also keep the desired path in the general.

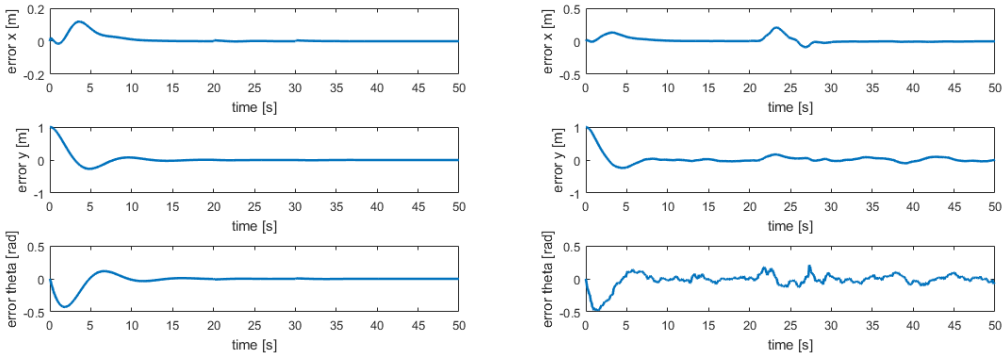
It can be seen more clearly in Fig.3.9b-3.9d, along each axis, we can see how well the controller track to the desired path. And we can see the affect of noise into the system.




(a) Without noise (b) With noise

Figure 3.10: The control signals of Nonlinear Lyapunov-based controller

According to Fig.3.10, the control signals of the controller are shown. The control signals is in the range of available values. Besides, we can see the affect of noise into the steering angle. although it is affected by noise on the system, the system is stable and the path is tracked.




(a) Without noise (b) With noise

Figure 3.11: The tracked error of Nonlinear Lyapunov-based controller

Additionally, the tracking errors in $x$, $y$ and $\theta$ of the reference trajectory are shown in the Fig.3.11. It shows that the error is converging to zero. Therefore, the desired trajectory is tracked, through the disturbance is added to the controller.

# Chapter 4

# Robot localization using Extended Kalman filter

## 4.1 Design the Extanded Kalman Filter for the Tricycle

In this section, we design the Extended Kalman filter for the Tricycle in localization problem. To design the Extended Kalman Filter, we linearise the plant and the measurement by deleting high-order terms from the Taylor expansion.

First, according to (2.4), we have the discretization kinematic model of the Type-2 Tricycle Mobile robot as follow:

$$\boldsymbol{x}_k = f\left(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}\right) = \begin{bmatrix} x_{k-1} + v_{k-1}\cos\theta_{k-1}\Delta T \\ y_{k-1} + v_{k-1}\sin\theta_{k-1}\Delta T \\ \theta_{k-1} + \frac{v_{k-1}}{d}\tan\gamma_{k-1}\Delta T \end{bmatrix} \tag{4.1}$$

where $\mathbf{x}_{k-1} = (x_{k-1}, y_{k-1}, \theta_{k-1})^T$ and $\mathbf{u}_{k-1} = (v_{k-1}, \gamma k - 1)^T$ are the position and control signal of the robot at the time $k-1$ and $\mathbf{x}_k = (x_k, y_k, \theta_k)^T$ is the robot position at the time $k$.

Applying the first-order Taylor expansion for (4.1), we get:

$$\boldsymbol{x}_k = \boldsymbol{x}_k^- + \boldsymbol{A}\left(\boldsymbol{x} - \boldsymbol{x}_{k-1}\right) + \boldsymbol{W}\boldsymbol{w}_{k-1} \tag{4.2}$$

where $\mathbf{x}_k^-$ is the predicted position through control signal and:

$$\boldsymbol{A} = \frac{\partial f}{\partial \boldsymbol{x}_{k-1}} = \begin{bmatrix} 1 & 0 & -v_{k-1}\sin\theta_{k-1}\Delta T \\ 0 & 1 & v_{k-1}\cos\theta_{k-1}\Delta T \\ 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

$$\boldsymbol{W} = \frac{\partial f}{\partial \boldsymbol{u}_{k-1}} = \begin{bmatrix} \cos\theta_{k-1}\Delta T & 0 \\ \sin\theta_{k-1}\Delta T & 0 \\ \frac{\tan\gamma_{k-1}}{d}\Delta T & \frac{v_{k-1}}{d\cos^2\gamma_{k-1}}\Delta T \end{bmatrix} \tag{4.4}$$

$$\boldsymbol{Q} = \begin{bmatrix} \sigma_v & 0 \\ 0 & \sigma_\gamma \end{bmatrix} \tag{4.5}$$

where $\sigma_v$ and $\sigma_\gamma$ is the variance of the velocity and steering angle, respectively.

Secondly, in our localization problem, we have two landmarks and we can get the distance and angle of robot to each other. Therefore, the measurement of the system as follow:

$$h\left(\boldsymbol{x}\right) = \begin{bmatrix} r_1 = \sqrt{\left(x - x_1\right)^2 + \left(y - y_1\right)^2} \\ r_2 = \sqrt{\left(x - x_2\right)^2 + \left(y - y_2\right)^2} \\ \beta_1 = \arctan\frac{y_1 - y}{x_1 - x} - \theta \\ \beta_2 = \arctan\frac{y_2 - y}{x_2 - x} - \theta \end{bmatrix} \tag{4.6}$$

Applying the first-order Taylor expansion for the measurement, we get:

$$\boldsymbol{z}_k \approx \boldsymbol{z}_k^- + \boldsymbol{H}\left(\boldsymbol{x} - \boldsymbol{x}_k^-\right) + \boldsymbol{V}\boldsymbol{v}_k \tag{4.7}$$

where

$$\boldsymbol{H} = \frac{\partial h}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{x-x_1}{\sqrt{(x-x_1)^2+(y-y_1)^2}} & \frac{y-y_1}{\sqrt{(x-x_1)^2+(y-y_1)^2}} & 0 \\ \frac{x-x_2}{\sqrt{(x-x_2)^2+(y-y_2)^2}} & \frac{y-y_2}{\sqrt{(x-x_2)^2+(y-y_2)^2}} & 0 \\ -\frac{y-y_1}{(x-x_1)^2+(y-y_1)^2} & \frac{x-x_1}{(x-x_1)^2+(y-y_1)^2} & -1 \\ -\frac{y-y_2}{(x-x_2)^2+(y-y_2)^2} & \frac{x-x_2}{(x-x_2)^2+(y-y_2)^2} & -1 \end{bmatrix} \tag{4.8}$$

$$\boldsymbol{R} = \begin{bmatrix} \sigma_r & 0 & 0 & 0 \\ 0 & \sigma_r & 0 & 0 \\ 0 & 0 & \sigma_\beta & 0 \\ 0 & 0 & 0 & \sigma_\beta \end{bmatrix} \tag{4.9}$$

After get the required matrices, the estimated position of the robot can be satisfied follow steps:

1. Predict:

$$\boldsymbol{x}_k^- = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, 0) \tag{4.10}$$

$$\boldsymbol{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{W}_k \mathbf{Q}_{k-1} \mathbf{W}_k^T \tag{4.11}$$

2. Measurement update:

$$\boldsymbol{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{4.12}$$

$$\boldsymbol{x}_k = \mathbf{x}_k^- + \mathbf{K}(\mathbf{z}_k - h(\mathbf{x}_k^-)) \tag{4.13}$$

$$\boldsymbol{P}_k = (\mathbf{I}_{3\times 3} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \tag{4.14}$$

where $\mathbf{x}_k^-$ and $\mathbf{P}_k^-$ are the predict position and the predict covariance matrix of the system in the step $k$. And, $\mathbf{x}_k$ and $\mathbf{P}_k$ is the the estimated position and estimated covariance matrix.

At each value of the time variable $k$, the value of the state, the covariance matrix at predict and estimate, and Kalman gain $\mathbf{K}$ is calculated step by step from (4.10) to (4.14).

## 4.2 The simulation results

In this section, we show the performance of the Extended Kalman filter in Robot localization problems. The results of the simulation are shown in the Fig.4.1-4.2.

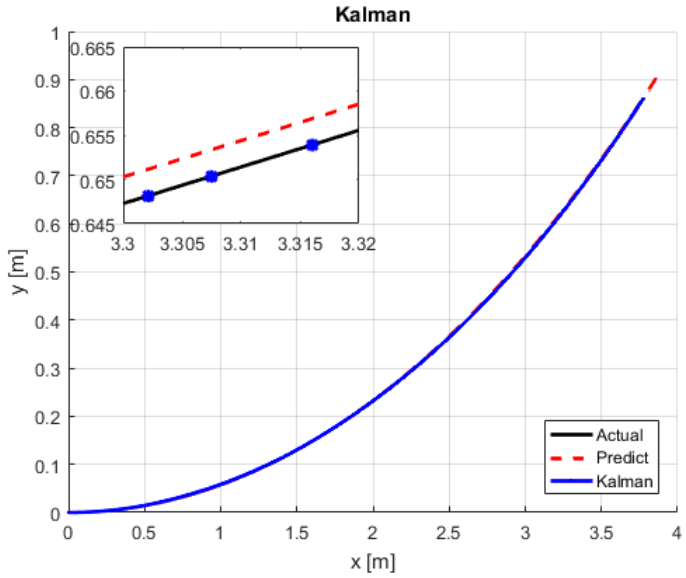



Figure 4.1: The comparison between predict path and Kalman path

According to Fig.4.1, there are three lines called: actual, predict, and Kalman. Where actual is the line get from 'contrlsig.mat' file. The predict is the line obtained by the kinematic model. And

Kalman line is obtained when applying the Extended Kalman filter to our system. We can see the Kalman estimated position is closer than the predicted one.

The error estimated position are shown in the Fig.4.2. It incldes the error in the state values $x$, $y$, $\theta$ between the prediction when using the kinematic model and the estimate by using Extended Kalman filter.

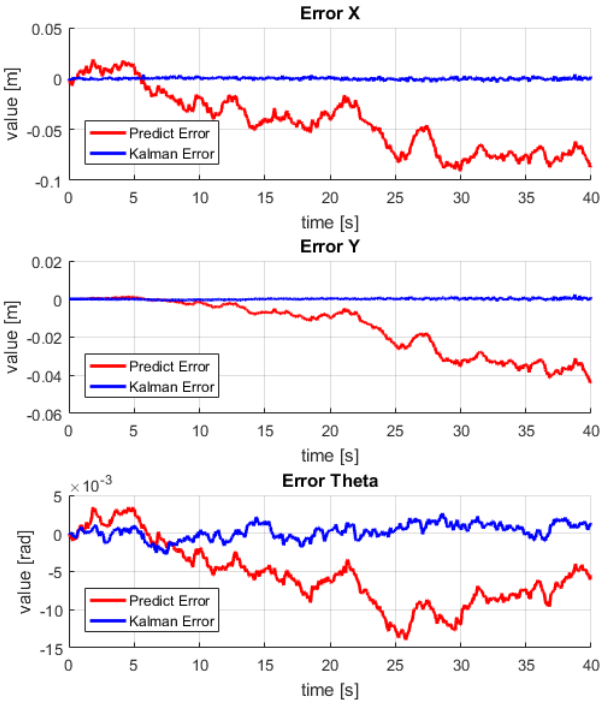



Figure 4.2: The localization error of predict path and Kalman path

The error in the prediction is increases time by time because the model errors or environment noising are not calculated in this case. But when using the Extended Kalman filter, those errors can affect the calculation state of the model by the measurement. So, as we can see, the estimated errors are nearer zero than the predicted error.

We can see that the Extended Kalman filter is a strong tool that estimate the robot position is near with the actual. It is very helpful in the localization of robots and many applications.