ENPM 667
Project-I

Technical Report

Observer-Based Adaptive Tracking Control of Wheeled Mobile Robots with Unknown Slippage Parameters

&

Leader-Follower Formation Control of Multiple Non-holonomic Mobile Robots Incorporating a Receding-Horizon Scheme

Pin-Hao Huang (117517842)

Po-Lun Chen (118307435)

ABSTRACT

Abstract— When it comes to robots which are used in rescue scene tracked mobile robots (TMRs) can be considered as the most important type of rescue mobile robots. Because of its large contact area of tracks with the ground provides superior advantages for TMRs such as better mobility in unstructured environments, though it may cause a higher risk of slippage including longitude slip and lateral slip which is an extremely important issue if you want to control your TMR smoothly. In this paper, an adaptive control approach is proposed for trajectory tracking of TMRs with unknown longitude slipping. A kinematic model of tracked TMRs is established in this paper, in which longitude slipping is considered and processed as a parameter (we ignore lateral slipping). Also, in some situation, we do not have equipment to obtain the orientation angle, which is a crucial variable in control. To solve this problem, we provide a observer to observe the value of orientation angle. Secondly, it is not efficiency that only send one robot to the rescue scene; therefore, in this report we also implement a leader-follower based formation control which is used to control several robots in the same time. In order to control these robots smoothly, a receding-horizon control is implemented to these robots. Using RH control scheme can improve the performance of the formation control.

In this technical report, we will develop a formation control, controlling three robots at the same time, also the leader of the team is suffering from the extremely large slippage, and also do not have sensor to obtain the orientation angle. The results are simulated for a tracked mobile robot using Simulink.

Keywords—Tracked Mobile Robot, Slippage, Trajectory Tracking, Adaptive Control, State Observer Feedback, Model Predict Control, Optimal Control

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1. INTRODUCTION

1.1 Introduction

In this paper, we will show our kinematic model which ignores the lateral slipping parameter. The main contributions of our work are the design of an adaptive control approach for tracking the desired trajectory of a class of tracked mobile robots in the presence of unknown longitude slipping at the kinematics level. The unknown slipping is compensated by the adaptive control technique. The proposed adaptive control system is designed by the backstepping idea in fixed coordinates that can achieve global trajectory tracking for reference trajectories in global regions. From the Lyapunov stability theorem, we prove that tracking errors and parameter estimation errors of the controlled closed-loop system are uniformly bounded, and the trajectory tracking errors can be made arbitrarily small by adjusting design parameters regardless of large initial tracking errors and unknown slipping. Also, design an observer to get the unknown orientation angle. The design and proof of the observer are provided in this paper.

This study also implements a leader-follower-based formation control. The formation control uses the receding horizon control to control our tracked mobile robot. Receding horizon control is very powerful, using this algorithm the convergence rate increase. It is very useful in the topic of multi-mobile robots. The goal of this paper is to control three tracked mobile robots at the same time. The first tracked mobile robot is the leader of the team. Also, the leader of the team is suffering from longitude slippage and cannot obtain the orientation angle directly. The other tracked mobile robots are the follower. The first follower uses SBOS formation which means that it only follows the leader. The second follower used SSOS which means that it follows two leaders at the same time. The second leader is the first follower. The position of the second follower is based on the relationship between these two tracked mobile robots which are followed by the second follower robot.

1.2 Literature review

A kinematic based on feedback and the feed-forward controller is designed by [1]. Another trajectory tracking controller design and stability analysis using the Lyapunov function is presented by [2]. Lots of knowledge about the control of mobile robot have been summarized in [3]. Because the lateral force and the resistance force is caused by soil mechanism, the Newton Raphson method and Least Square method are applied in [4] to estimate the soil parameters. In [5] different type of soils lateral and longitude slippage is estimated by using the extended Kalman filter (EKF). A nonlinear adaptive control law is derived by [6], the stability is also proved in this paper. Furthermore, the adaptive control law and the orientation angle observer are designed in [7]. In [8] a leader-follower-based formation control is shown, including SSOS and SBOS which are two different types of leader follower-based control, the first one, SBOS is a two-robot formation and the second one, SSOS, is a three-robots formation. This paper also uses the receding-horizon control to improve the performance and also solve the information slowly update problem. In [9] a different derivation of formation control is provided.

CHAPTER 2. DISCUSSION

In this technical report, we will mainly focus on the [6] Juliano G.; CAMINO, and Juan F (2011) and [8] CHEN, Jian (2010), however, the adaptive update law used the method in [7] Mingyue (2018) and the Receding-horizon control also referenced the method provided in [9] CHEN, Jian (2011). Furthermore, the derivation of SSOS formation control will use our own derivation.

2.1 Kinematic model for trajectory tracking

An ideal geometry relationship of tracked mobile robot Body Coordinate corresponding to Cartesian Coordinate is shown in Fig 1. global coordinate is presented as $\{X, Y\}$ and tracked mobile robot posture is presented by $\mathrm{p} = \{x,\ y,\ \theta\}$. Tracked mobile robot's gravity center is presented by C. Body Coordinate presented as $\{\mathrm{x}, \mathrm{y}\}$ which is fixed in the gravity center of tracked mobile robot. θ is the angle between Cartesian Coordinate and Body Coordinate. Velocity and angular velocity are presented as v and ω respectively. The rotation speed of right tracked and left tracked are presented as ω_R and ω_L. The width of the tracked mobile robot is presented as 2b.

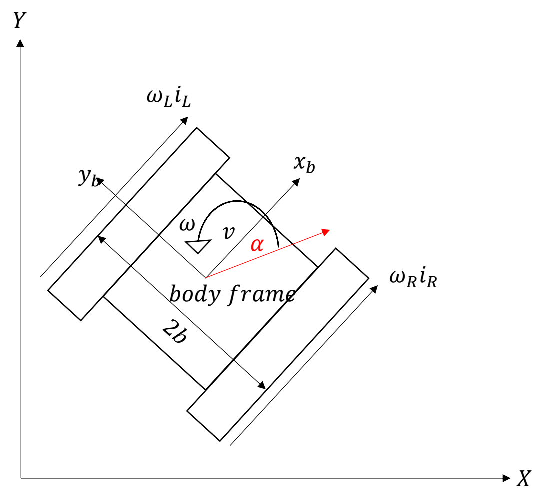


Figure 1. Geometry relationship of the tracked mobile robot without lateral slip

Tracked mobile robot with lateral and longitude slippage. The kinematic model can be obtained as follows:

$$\dot{x}_b = \frac{r}{2}(\omega_R i_R + \omega_L i_L) \tag{1a}$$

$$\dot{y}_b = \dot{x}_b \tan\alpha \tag{1b}$$

$$\omega = \frac{r}{2b}(\omega_L i_L - \omega_R i_R) \tag{1c}$$

From equation (1a)、(1b) and (1c) kinematic model can be deduced to Cartesian coordinate by rotate transformation as follows:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2}\tan\alpha & \frac{r}{2}\tan\alpha \\ \frac{r}{2b} & -\frac{r}{2b} \end{bmatrix} \begin{bmatrix} \omega_L i_L \\ \omega_R i_R \end{bmatrix} \tag{2}$$

Although the lateral and longitude are existing in the real plant, we can simplify our plant by some assumptions. Considering the low speed of our tracked mobile robot, we can ignore the lateral slip and the final kinematic model can be simplified as follow:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ 0 & 0 \\ \frac{r}{2b} & -\frac{r}{2b} \end{bmatrix} \begin{bmatrix} \omega_L(1-i_l) \\ \omega_R(1-i_r) \end{bmatrix} \tag{3a}$$

$$\mathrm{i_l} = \frac{r\omega_L - v_L^s}{r\omega_L} \tag{3b}$$

$$\mathrm{i_r} = \frac{r\omega_R - v_R^s}{r\omega_R} \tag{3c}$$

v_L^s and v_R^s are the true velocity that robot moving in the environment. $\mathrm{i_l}$ and $\mathrm{i_r}$ are the slip ratio.

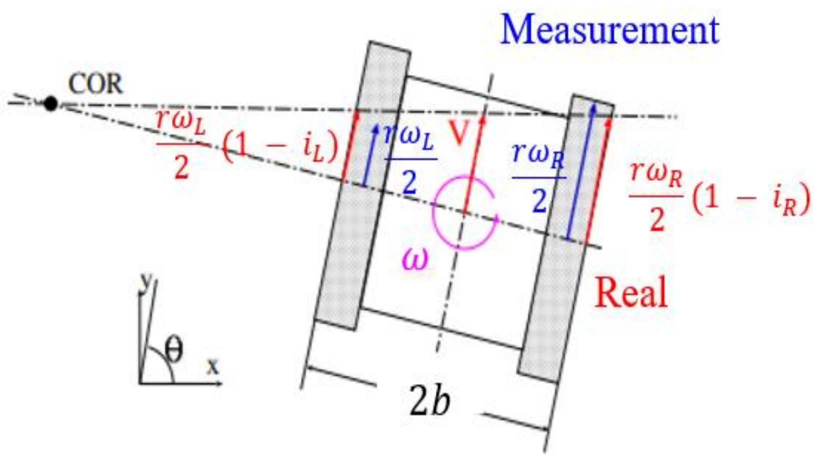


Figure 2. Concept of slippage influence on the tracked mobile robot

The effect of how do slip influences the tracked mobile robot is shown in figure 2. The value of slippage is based on the following assumption. The others assumptions made in these papers are also shown here.

Assumption 1: The range of tracked mobile robot longitude slipping ratio $\mathrm{i_r}$ and $\mathrm{i_l}$ are between $[0,1)$

Assumption 2: The reference velocities and their derivative are all available and bounded.

Assumption 3: This system has encoders and a laser range scanner which can measure (X, Y) and the speed of the left and right tracks.

From *Assumption 1* we know that if the slip ratio is equal to 1, tracked mobile robots are completely slipping, which means the robots would not move.

2.2 RH-LF FORMATION CONTROL FRAMEWORK

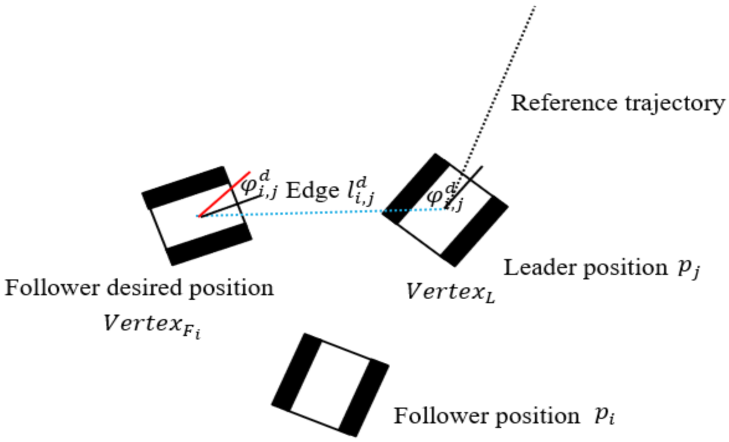


Figure 3. Concept of formation control

The Above figure shows the concept of a Leader-Follower based multi-robot system. To send a team of tracked mobile robots to work, we have to design the formation control framework. The formation control frameworks have two basic types one is SBOS and the other is SSOS. SBOS is the two robot's formation with a leader and a follower. The leader of the team of the tracked mobile robots will follow a reference trajectory and the followers will track the position which is derived from several parameters from the leader. SSOS is the three robot's formation, the followers will follow two leaders at the same time. The position of the followers is derived from the geometry relationship between two leaders.

Using the Leader-Follower based formation control, the leader of the team of tracked mobile robots is defined as $\mathrm{Vertex_L}$, and its position is defined as p_j. The followers of the team of the tracked mobile robots are defined as $\mathrm{Vertex_{F_i}}$ with index i = 2, 3, 4, …, n. n equal to the numbers in the group. The Leader-Follower pairs are defined as $\mathrm{G_1}, G_2, \dots, G_m$. m is the number of pairs. $G_i = \mathrm{LF}\left(\mathrm{Vertex_{F_i}} \rightarrow \mathrm{Vertex_L}\right)$ or $G_i = \mathrm{LF}\left(\mathrm{Vertex_{F_i}} \rightarrow (\mathrm{Vertex_{L_1}}, \mathrm{Vertex_{L_2}})\right)$. All of the tracked mobile robots are linked indirectly as a network. This network is defined previously. Note that the definition is different with the paper. The reason for the difference is that I want to show the similarities between formation and graph theory.

Formation control means that a group of tracked mobile robots form a graph or a previously defined relationship we designed in the beginning. The Leader of the group will track the reference trajectory and the Followers are controlled in order to track the Leader based on the parameters of the vertexes and the edges.

The kinematic model of the tracked mobile robot is derived in equation (3a) and the followers slip parameters are all assumed as 0. And then, we introduce the $\mathrm{i^{th}}$ numbers of a tracked mobile robot to the kinematics model. The equation will become

$$\begin{bmatrix} \dot{X}_i \\ \dot{Y}_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{r_i}{2} & \frac{r_i}{2} \\ 0 & 0 \\ \frac{r_i}{2b_i} & -\frac{r_i}{2b_i} \end{bmatrix} \begin{bmatrix} \omega_{L_i} \\ \omega_{R_i} \end{bmatrix} \tag{4}$$

Next, we will introduce the two types of Leader-Follower based formation control which are SBOS and SSOS. By using these two types of formation control we can track the reference trajectory and keep the formation at the same time.

2.2.1 SBOS

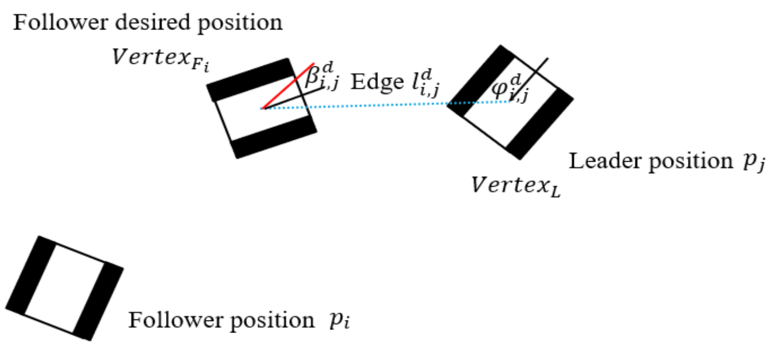


Figure 4. SBOS

The first type we will introduce is SBOS which is a two-robot formation with a leader and a follower. The relationship is defined as $G_i = \mathrm{LF}(\mathrm{Vertex}_{\mathrm{F_i}} \rightarrow \mathrm{Vertex}_{\mathrm{L}})$ which means the follower $\mathrm{Vertex}_{\mathrm{F_i}}$ will follow the Vertex_L. And we defined three parameters of Vertex_L which are separation $l_{i,j}^d$, bearing $\varphi_{i,j}^d$, orientation deviation $\beta_{i,j}^d$ and $Vertex_{F_i} = [x_i^d, y_i^d, \theta_i^d]$ is the position of the follower. $Vertex_L$ position is defined as $[x_j, y_j, \theta_j]$. The relationship between leader and follower is shown as follows.

$$Vertex_{F_i} = \begin{bmatrix} x_i^d \\ y_i^d \\ \theta_i^d \end{bmatrix} = \begin{bmatrix} x_j + l_{i,j}^d \cos \varphi_{i,j}^d \\ y_j + l_{i,j}^d \sin \varphi_{i,j}^d \\ \theta_j + \beta_{i,j}^d \end{bmatrix} \tag{5}$$

The relationship between the leader and the follower is described by the polar coordinate which $l_{i,j}^d$ is the desired distance and the $\varphi_{i,j}^d$ is the angle. From these two parameters, we can obtain the desire x position and y position. The desire theta θ_i^d can be defined by the combination of leader desired θ_j and the orientation deviation $\beta_{i,j}^d$.

2.2.2 SSOS

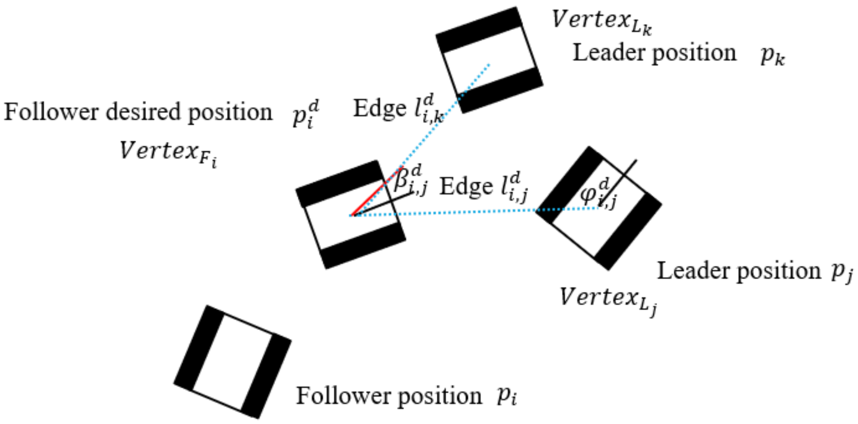


Figure 5. SSOS

In this part, we will introduce the second type of formation control which is called SSOS. SSOS is a three-robot formation, including two leaders and one follower, the concept of the SSOS formation control is shown in figure 5. The parameters are similar to the SBOS. The concept of SSOS formation control can be represented mathematically in the following equation.

$$G_i = \mathrm{LF}\left(\mathrm{Vertex}_{\mathrm{F_i}} \rightarrow (\mathrm{Vertex}_{\mathrm{L_j}}, \mathrm{Vertex}_{\mathrm{L_k}})\right) \tag{6}$$

$\mathrm{Vertex}_{\mathrm{L_j}}$ and $\mathrm{Vertex}_{\mathrm{L_k}}$ are the two leaders in this team. and $\mathrm{Vertex}_{\mathrm{F_i}}$ is the follower which track the position corresponding to $\mathrm{Vertex}_{\mathrm{L_j}}$ and $\mathrm{Vertex}_{\mathrm{L_k}}$.

$$Vertex_{F_i} = \begin{bmatrix} x_i^d \\ y_i^d \\ \theta_i^d \end{bmatrix} = \begin{bmatrix} \frac{\left(b_i \pm (b_i^2 - 4a_i c_i)\right)}{a_i} \\ \frac{\left(M_i - \Delta x_i x_i^d\right)}{\Delta y_i} \\ \theta_j + \beta_{i,j}^d \end{bmatrix} \tag{7a}$$

$$M_i = \frac{1}{2}\left[\left(x_j^2 - x_k^2\right) + \left(y_j^2 - y_k^2\right) - {l_{i,j}^d}^2 + {l_{i,k}^d}^2\right] \tag{7b}$$

$$\Delta \mathrm{x_i} = \mathrm{x_j} - x_k, \Delta y_i = y_j - y_k \tag{7c}$$

$$\begin{bmatrix} a_i \\ b_i \\ c_i \end{bmatrix} = \begin{bmatrix} 2(\Delta \mathrm{x_i^2} + \Delta \mathrm{y_i^2}) \\ 2\left(M_i \Delta x_i - \Delta y_i (\Delta x_i y_i - \Delta y_i x_i)\right) \\ \left(M_i - \Delta y_i y_j\right)^2 - \left(\Delta y_i x_j\right)^2 - \left(\Delta y_i l_{i,j}^d\right)^2 \end{bmatrix} \tag{7d}$$

In this technical report we will provide a different method of finding the desire position of SSOS follower, the derivation and the property of above equations are provided in **Appendix [1]**.

2.3 CONTROLLER DESIGN

2.3.1 FORMATION CONTROL

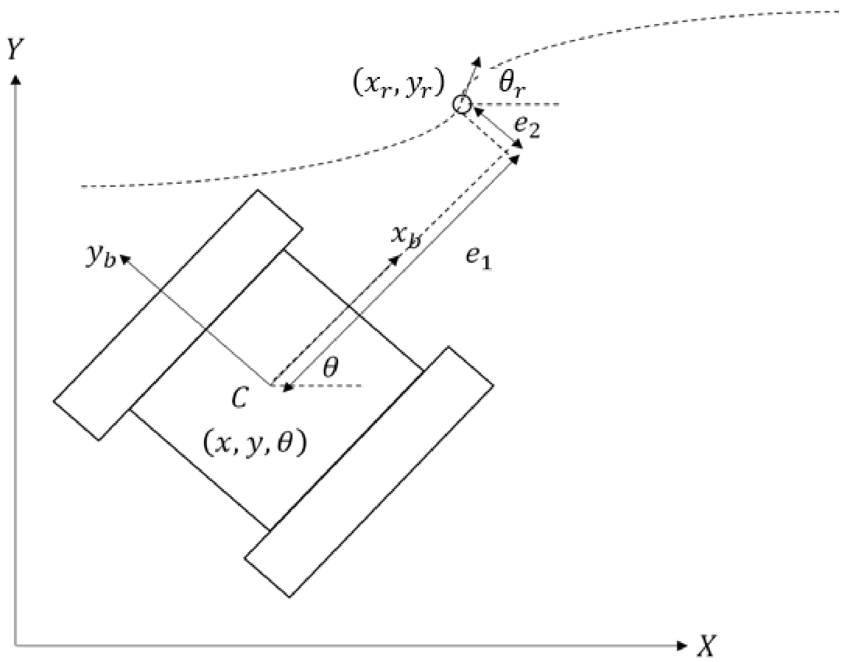


Figure 6. Trajectory tracking schematic diagram

First of all, we defined the desired position of the trajectory (x_r, y_r, θ_r). Then, we have to obtain the feedforward control input (v_r and ω_r) which is used to assist the tracking of trajectory. These terms are obtained by the differential of trajectory, and can be computed as follow:

$$v_r(t) = \pm\sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \tag{8a}$$

$$\omega_r(t) = \frac{\ddot{y}_r(t)\dot{x}_r(t) - \ddot{x}_r(t)\dot{y}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)} \tag{8b}$$

$$\theta_r(t) = \tan^{-1}\frac{\dot{y}_r}{\dot{x}_r} \tag{8c}$$

Equations (8a), and (8c) are very intuitive, (8a) just takes differential and combination of two vectors. Using the formula of differential of (8c), then the equation (8b) can be obtained. Therefore, we will not provide the derivation of equations (8a) and (8c) instead we will provide the controllability of a trajectory and (8b) in **Appendix [2].**

Where v_r and ω_r are the desired velocity and angular velocity. The goal of this adaptive trajectory tracking controller for a tracked mobile robot, even if the longitude slip occurs.

$$\lim_{t\to\infty}(p - p_r) = 0 \tag{9}$$

In the beginning, we introduce the control theory without adaptive control. The first step to designing the trajectory tracking algorithm is to get the tracking error e_1, e_2 and e_3 and transform tracking error from Body Coordinate into the Cartesian Coordinate for later calculation.

$$e = [e_1 \quad e_2 \quad e_3]^T = T_e(p_r - p) \tag{10a}$$

$$T_e = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10b}$$

We would like to derive the Leader-Followers based tracked mobile robot controller design which is just similar to above equations. e_{1_i}, e_{2_i} and e_{3_i} which means the difference between the reference position p_{r_i} and the actual position p_i. Then we can obtain the following equation.

$$e = [e_{1_i} \quad e_{2_i} \quad e_{3_i}]^T = T_{e_i}\left(p_{r_i} - p_i\right) \tag{11a}$$

$$T_{e_i} = \begin{bmatrix} \cos\theta_i & \sin\theta_i & 0 \\ -\sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11b}$$

From (3), (11a), (11b), taking time derivative to the tracking error model of the tracked mobile robot to get the system dynamic The equation is shown as follows and the derivation will be provided in **Appendix [3]**.

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \omega e_2 + v_r \cos e_3 - v \\ -\omega e_1 + v_r \sin e_3 \\ \omega_r - \omega \end{bmatrix} \tag{12}$$

From above equation (10) we can derive to the error dynamic of Leader-Follower based formation control.

$$\begin{bmatrix} \dot{e}_{1_i} \\ \dot{e}_{2_i} \\ \dot{e}_{3_i} \end{bmatrix} = \begin{bmatrix} \omega_i e_{2_i} + v_{r_i} \cos e_{3_i} - v_i \\ -\omega_i e_{1_i} + v_{r_i} \sin e_{3_i} \\ \omega_{r_i} - \omega_i \end{bmatrix} \tag{13}$$

It can also use to Leader-Follower based formation control. The control inputs are all identity with previous control inputs. The only difference is i^{th} which means the number of followers.

To track the desired trajectory, we have to design a control law of this system. The linear state feedback control law combined with feedforward control which is derived from [3] is shown as follows.

$$v = v_r \cos e_3 - u_1 \tag{14a}$$

$$\omega = \omega_r - u_2 \tag{14b}$$

$$\dot{e} = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e + \begin{bmatrix} 0 \\ \sin e_3 \\ 0 \end{bmatrix} v_r + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{14c}$$

Obtaining the linear time-varying system by linearizing (9) then designing the linear feedback law to converge all-state (e_1, e_2, e_3). The control structure of the system is shown in (14a), (14b), (14c). After designing this structure, we need to design the control input u_1, u_2 to make this system stable. The control input is shown as follow:

$$u_1 = -k_1 e_1 \tag{15a}$$

$$u_2 = -\frac{k_{2_i} v_{r_i}(t) \sin e_{3_i}}{e_{3_i}} e_{2_i} - k_{3_i} e_{3_i} \tag{15b}$$

Desired closed-loop polynomial is shown in (16)

$$(\lambda + 2\varsigma a)(\lambda^2 + 2\varsigma a\lambda + a^2) \qquad \varsigma, a > 0 \tag{16}$$

Choosing the feedback gain as (16) to let the eigenvalue always be negative constant.

$$k_1 = k_3 = 2\varsigma\sqrt{\omega_r^2(t) + b v_r^2(t)}, k_2 = b|v_r(t)| \tag{17}$$

Or we can choose the feedback gain k_1, k_2 and k_3 as negative constant, the control system will also be stable. In this paper we will choose the feedback gains k_1, k_2 and k_3 as negative constant.

Summarize equations (14a), (14b), (14c), (15a), (15b) above and we can get the control law shown in (18). We also get the control input of multi-robot control which is shown in (19).

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ \omega_r + \frac{k_2 v_r e_2 \sin e_3}{e_3} + k_3 \sin e_3 \end{bmatrix} \tag{18a}$$

$$\begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} v_{r_i} \cos e_{3_i} + k_{1_i} e_{1_i} \\ \omega_{r_i} + \frac{k_{2_i} v_{r_i} e_2 \sin e_{3_i}}{e_{3_i}} + k_{3_i} e_{3_i} \end{bmatrix} \tag{18b}$$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 v_r e_2 + k_3 \sin e_3 \end{bmatrix} \tag{19}$$

Note that both control law (18b) and (19) can make the system stable, and the control law (18b) is used in formation control and the control law (19) is used in adaptive control. All of the control gains k_1, k_2, k_3 and $\mathrm{k}_{1_\mathrm{i}}, k_{2_i}, k_{3_i}$ are positive so that the system has negative eigenvalues and the system is stable. To move on we have to consider the slip in this system. After combining this system with unknown slip, we can get the following equation.

First, we want to derive the stability of the formation control system without considering the slippage issue. Without slip, we can candidate our Lyapunov function as follows. Note that our Lyapunov function candidate is different from the original paper Lyapunov function.

$$\mathrm{V(t)} = \frac{k_{2i}}{2}\left(e_{1_i}^2 + e_{2_i}^2\right) + \frac{1}{2} e_{3_i}^2 \tag{20}$$

In order to analyze this system, we take the time derivative of equation (20). Note that in this section, we are different from the original paper, we add the control gain k_{2_i} which is a constant, also because k_{2_i} is a constant, the derivative is similar. The introduction of the Lyapunov function will be shown in **Appendix [4]**.

$$\dot{V}(\mathrm{t}) = k_{2_i} e_{1_i} \dot{e}_{1_i} + k_{2_i} e_{2_i} \dot{e}_{2_i} + \dot{e}_{3_i} e_{3_i} \tag{21}$$

Combining equation (21) with error dynamic equation (13), then we will get the following equation. The derivation of the following equation will be shown in **Appendix [5].**

$$\dot{V}(\mathrm{t}) = -k_{1_i} k_{2_i} e_{1_i}^2 - k_{3_i} e_{3_i}^2 \leq 0 \tag{22}$$

Because the time derivative of the Lyapunov function will always be smaller or equal to 0, we can derive that the Lyapunov function $\mathrm{V(t)}$ is a non-increasing function. Because k_{1_i} and k_{3_i} are constant and based on *Assumption 2,* we know that the reference velocities, reference angular velocities, and their derivative are all available and bounded. Also, we know that the $||e_i(t)||_2$ is bounded. Based on the above derivation and the equation (13) we can know that the $\dot{e}_i$ is bounded. Because the time derivative of $\dot{e}_{1_i}$ and $\dot{e}_{3_i}$ are bounded, we can obtain that e_i is continuously uniform. Using Barbalat's lemma, we can obtain that $e_{1_i} \to 0$ *and* $e_{3_i} \to 0$ as time approach infinity. The introduction of Barbalat's Lemma will be shown in **Appendix [6]**.

After $e_{1_i} \to 0$ *and* $e_{3_i} \to 0$, we can obtain the following equation from equation (10c) and equation (11b).

$$\dot{e}_{3_i} = -k_{2_i} sign\left(v_i^r(t)\right) e_{2_i} \tag{23}$$

From previous assumption 2, we know that $\dot{e}_{3_i} \to 0$ and k_2 *and* $sign\left(v_r(t)\right)$ is not equal to 0, then e_{2_i} is equal to 0.

Selecting appropriate gain which k_1, k_2, k_3 and $k_{1_i}, k_{2_i}, k_{3_i}$ are all positive, we can conclude that this system is stable. This derivative and control law will be used in the formation control and receding horizon control derivation after deriving the traditional control method, we will discuss the adaptive trajectory tracking control with unknown longitude slippage in the following section.

2.3.2 Adaptive Trajectory Tracking Control

In this section, we will derive the adaptive trajectory tracking control, which includes a control law with unknown parameters' slip and an adaptive update law to converge the error of estimate slip and unknown actual slip.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{\omega_L(1-i_L)r+\omega_R(1-i_R)r}{2} \\ \frac{-\omega_L(1-i_L)r+\omega_R(1-i_R)r}{b} \end{bmatrix} = \mathrm{T} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} \tag{24}$$

$$\mathrm{T} = \begin{bmatrix} \frac{r(1-i_L)}{2} & \frac{r(1-i_R)}{2} \\ \frac{-(1-i_L)}{b} & \frac{(1-i_R)}{b} \end{bmatrix} \tag{25}$$

After getting the forward kinematic of tracked mobile robot, we derive the inverse kinematic.

$$\begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix} = T^{-1} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{26a}$$

$$\mathrm{T}^{-1} = \begin{bmatrix} \frac{1}{r(1-i_L)} & \frac{b}{2(1-i_L)} \\ \frac{1}{r(1-i_R)} & \frac{b}{2(1-i_R)} \end{bmatrix} \tag{26b}$$

Where i_L and i_R are the real slipping parameters which are unknown. Because we cannot know the actual value of the slipping parameters, we defined new parameters $\hat{\imath}_R$, $\hat{\imath}_L$, $\tilde{\imath}_R$ and $\tilde{\imath}_L$. $\hat{\imath}_R$ and $\hat{\imath}_L$ represent the estimated value of slippage. $\tilde{\imath}_R$ and $\tilde{\imath}_L$ represent the error between estimate values and the real values. The equations of $\tilde{\imath}_R$ and $\tilde{\imath}_L$ are shown as following equations (27a) and (27b).

$$\tilde{\imath}_R = \hat{\imath}_R - i_R \tag{27a}$$

$$\tilde{\imath}_L = \hat{\imath}_L - i_L \tag{27b}$$

Our goal is to track the trajectory and converge $\tilde{\imath}_R$ and $\tilde{\imath}_L$ to 0. To make derivation earlier, we defined two new parameters which are the reciprocal of one minus slipping parameters.

$$\hat{a}_R = \frac{1}{1-\hat{\imath}_R} \tag{28a}$$

$$\hat{a}_L = \frac{1}{1-\hat{\imath}_L} \tag{28b}$$

$$a_R = \frac{1}{1-i_R} \tag{28c}$$

$$a_L = \frac{1}{1-i_L} \tag{28d}$$

Then we can rewrite the inverse kinematic shown in (26b) as following equation.

$$\mathrm{T}^{-1} = \begin{bmatrix} \frac{\hat{a}_L}{r} & -\frac{\hat{a}_L b}{2} \\ \frac{\hat{a}_R}{r} & \frac{\hat{a}_R b}{2} \end{bmatrix} \tag{29}$$

In order to derive the adaptive law of known slipping parameters, we differential the dynamic error equation (12), the result is shown as the following equation.

$$\dot{e}_1 = \frac{a_L+\tilde{a}_L}{a_L}\left[\left(\frac{e_2 v}{b}+\frac{v}{2}\right)-\left(\frac{e_2\omega}{2}+\frac{b\omega}{4}\right)\right]+\frac{a_R+\tilde{a}_R}{a_R}\left[\left(\frac{e_2 v}{b}-\frac{v}{2}\right)+\left(\frac{e_2\omega}{2}-\frac{b\omega}{4}\right)\right]+\mathrm{v_r}\cos e_3 \tag{30a}$$

$$\dot{e}_2 = \frac{a_L+\tilde{a}_L}{a_L}\left(\frac{e_1 v}{b}-\frac{e_1\omega}{2}\right)-\frac{a_R+\tilde{a}_R}{a_R}\left(\frac{e_1 v}{b}-\frac{e_1\omega}{2}\right)+\mathrm{v_r}\sin e_3 \tag{30b}$$

$$\dot{e}_3 = \frac{a_L+\tilde{a}_L}{a_L}\left(\frac{v}{b}-\frac{\omega}{2}\right)-\frac{a_R+\tilde{a}_R}{a_R}\left(\frac{v}{b}+\frac{\omega}{2}\right)+\omega_r \tag{30c}$$

To converge these two unknown slippage parameters we chosen a Lyapunov function as follows.

$$\mathrm{V(t)} = \frac{1}{2}e_1^2+\frac{1}{2}(e_2+k_3e_3)^2+\frac{(1-\cos e_3)}{k_2}+\frac{\tilde{a}_L^2}{2\rho_1 a_L}+\frac{\tilde{a}_R^2}{2\rho_2 a_R} \tag{31}$$

ρ_1, ρ_2 and $k_2$are all positive constants. Then we take the time derivative of the system, we can get. The derivation will be provided in **Appendix [7]**.

Combining equations (24), (30a), (30b) and (30c), then we obtain

$$\dot{V}(\mathrm{t}) = -k_1e_1^2-\frac{k_2k_3v_r}{2}(e_2+k_3e_3)^2-\frac{v_r}{2k_2k_3}\sin^2 e_3+\frac{\tilde{a}_L}{a_L}\left\{\frac{\dot{\hat{a}}_L}{\rho_1}-(v-b\omega)\left[\left(\frac{e_2}{2b}+\frac{1}{2}\right)e_1-\frac{(e_2+k_3e_3)e_1}{2b}-\frac{k_3(e_2+k_3e_3)}{2b}-\frac{\sin e_3}{2bk_2}\right]\right\}+\frac{\tilde{a}_R}{a_R}\left\{\frac{\dot{\hat{a}}_R}{\rho_2}-(v+b\omega)\left[-\left(\frac{e_2}{2b}-\frac{1}{2}\right)e_1+\frac{(e_2+k_3e_3)e_1}{2b}+\frac{k_3(e_2+k_3e_3)}{2b}-\frac{\sin e_3}{2bk_2}\right]\right\} \tag{32}$$

We can design the adaptive update law from (32)

$$\dot{\hat{a}}_R = \rho_1(v-b\omega)\left[(\frac{e_2}{2b}+\frac{1}{2})e_1-\frac{(e_2+k_3e_3)e_1}{2b}-\frac{k_3(e_2+k_3e_3)}{2b}-\frac{\sin e_3}{2bk_2}\right] \tag{33a}$$

$$\dot{\hat{a}}_L = \rho_2(v+b\omega)\left[\left(-\left(\frac{e_2}{2b}+\frac{1}{2}\right)e_1+\frac{(e_2+k_3e_3)e_1}{2b}+\frac{k_3(e_2+k_3e_3)}{2b}+\frac{\sin e_3}{2bk_2}\right)\right] \tag{33b}$$

Substitute (33a) and (33b) into (32) we can get

$$\dot{V}(\mathrm{t}) = -k_1e_1^2-\frac{v_r}{2}k_2k_3(e_2+k_3e_3)^2-\frac{v_r}{2k_2k_3}\sin^2 e_3 \leq 0 \tag{34}$$

From equation (34), we know that the system is asymptotically stable. The state error e will converge to 0 in the finite time.

2.3.3 Observer Design

In order to track the trajectory, we have to measurement all the state variables. However, based on *Assumption 3*, only (X, Y) can be measured. It is impossible to track the trajectory without the information of θ. Therefore, we design an observer to get reconstruct the state of θ. To design the state observer, we have to assume in the beginning.

Assumption 4: Linear velocity of the tracked mobile robot is bounded which means $0 < v_{min} \leq v < v_{max}$.

Because of the absence of $[e_1 \quad e_2 \quad e_3]^T$ we can not implement the control law shown in (14). To solve this problem, we design an angle observer which is used to estimate the orientation angle. The state variables of the observer are shown as follows.

$$z_1 = x \tag{35a}$$

$$z_2 = y \tag{35b}$$

$$z_3 = \cos\theta \tag{35c}$$

$$z_4 = \sin\theta \tag{35d}$$

The observer is design as follow

$$\begin{cases} \dot{\alpha} = -\hat{z}_4\omega - L\hat{z}_3 v \\ \dot{\beta} = \hat{z}_3\omega - L\hat{z}_4 v \\ \hat{z}_3 = \alpha + Lz_1 \\ \hat{z}_4 = \beta + Lz_2 \end{cases} \tag{36}$$

L is the observer gain which is a positive constant. α and β are the state variables of the observer. $\hat{z}_3$ and $\hat{z}_4$ are the estimated value of the state.

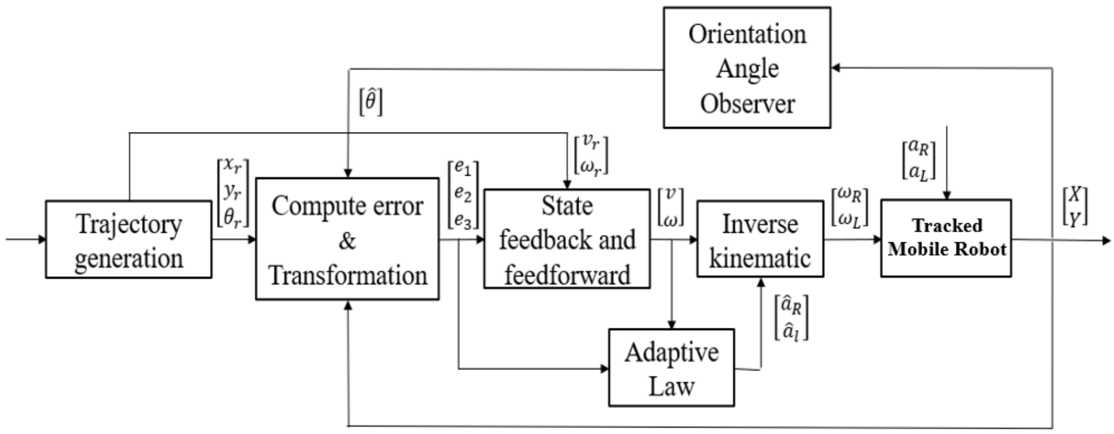


Figure 7. Control scheme

Figure 7 shows the control structure of our adaptive control trajectory tracking system.

2.3.4 Convergence Analysis of Observer

In this section, we will prove the stability of the observer using the Lyapunov function.

Theorem 1: Based on the *assumption 3* and *assumption 4*, the orientation's dynamic errors combining with the kinematic model of tracked mobile robot unknown parameter θ will asymptotically converge exponentially to 0 in finite time.

Proof: Based on the equation (2), (35a), (35b), (35c) and (35d), we can obtain the differential equations of the observer as follows

$$\dot{z}_1 = z_3 v \tag{37a}$$

$$\dot{z}_2 = z_4 v \tag{37b}$$

$$\dot{z}_3 = -\omega \sin\theta = -\omega z_4 \tag{37c}$$

$$\dot{z}_4 = \omega \cos\theta = \omega z_3 \tag{37d}$$

Defining observer error parameters $\tilde{z}_k$ which can be written as $\tilde{z}_k = z_k - \hat{z}_k$, k = 3 or 4. Introduce these new parameters to the differential equation we can get the new dynamic equation as follow

$$\begin{cases} \dot{\tilde{z}}_3 = \dot{z}_3 - \dot{\hat{z}}_3 = -\tilde{z}_4\omega - L\tilde{z}_3 v \\ \dot{\tilde{z}}_4 = \dot{z}_4 - \dot{\hat{z}}_4 = \tilde{z}_3\omega - L\tilde{z}_4 v \end{cases} \tag{38}$$

In order to analysis the observer we choose a Lyapunov function which is shown as follow

$$\mathrm{V_O} = \frac{1}{2}\tilde{z}_3^2 + \frac{1}{2}\tilde{z}_4^2 \tag{39}$$

After defining our Lyapunov function we take the differential of equation (39) and the result is shown as follow

$$\dot{V}_O = -\mathrm{L}(\tilde{z}_3^2 + \tilde{z}_4^2)v \leq -2v_{min}LV_O \tag{40}$$

$$V_O \leq V_O(0)e^{-2v_{min}Lt} \tag{41}$$

Using equations (38) and (40) we can obtain the following equation. Note that the will be provided in **Appendix [8].**

$$\left|\left|(\tilde{z}_3, \check{z}_4)\right|\right|_2 \leq \left|\left|\left(\tilde{z}_3(0), \check{z}_4(0)\right)\right|\right|_2 \; e^{-2v_{min}Lt} \tag{42}$$

we can get $\hat{\theta}$ by using $\hat{\theta} = \arctan\frac{\hat{z}_4}{\hat{z}_3}$. Also, because we know that L and v_{min} are positive, the observer dynamic error $\tilde{z}_3$ and $\check{z}_4$ will asymptotically converge exponentially to zeros in finite time. Converging $\tilde{z}_3$ and $\check{z}_4$ means that the estimated value is approaching the actual value. Combining above knowledge we know that $\hat{\theta}$ will converge to θin finite time. The trajectory tracking algorithm derived above can be implemented.

2.3.5 Analysis of adaptive control system

In this section, we will analyze the convergence of tracking errors.

First of all, we have to introduce Barbalat's lemma which means that if f(x) is uniformly continuous and also the (42) exists and is finite then we can get the equation (43).

$$\lim_{t\to\infty}\int_0^{\infty} f(x)dx \tag{42}$$

$$\lim_{t\to\infty} f(t) = 0 \tag{43}$$

Based on the separation principle which is derived in **Appendix [9]**, we can design the adaptive law and the control law of trajectory tracking and the observer design separately. And the observer errors do not affect the asymptotic stability.

Theorem 2: To be more specific, in this section we are trying to prove that we can converge the tracking error (10a) and while we implement the adaptive law (33a), (33b), and the control law (18), the unknown parameters longitude slippage will asymptotically converge to zero at the same time.

Proof : In order to prove theorem 2, we choose a Lyapunov function which is shown in (31). Then we define the domain D that $\mathrm{D} = \{e \in R^3 | -\pi < e_3 < \pi, e_3 \neq 0\}$. Then we know the Lyapunov we chosen $\mathrm{V(t)} > 0$ is positive definition.

We have already analyze in previous that satisfy the condition discussed above the error dynamic equation (30a), (30b), (30c) and the adaptive law (33a), (33b) at $\mathrm{t} \in [0, \infty), V(t)$ is the monotone and nonincreasing function, therefore we can get the following equation.

$$\mathrm{V(t)} \leq V(0), \forall t \geq 0 \tag{44}$$

Because $V(t)$ is the monotone and nonincreasing function, and $V(t)$ is positive definition, we can know that V is bounded. From the above derivation we can realize that $\tilde{a}_L^2$, $\tilde{a}_R^2$, e_1, and e_2 are bounded. To move on, combining with equation (18) we know that $[v, \omega]$ are bounded. Also from (30a), (30b) and (30c), we can know that $[\dot{e}_1, \dot{e}_2, \dot{e}_3]$ are bounded. To use Barbalat's Lemma we have to take the second derivative of V and the derivate is shown as follows. Because this derive of the following equations are very intuition, we will not provide the derivation of equation (45a) and equation (45b).

$$\dot{V}(\mathrm{t}) = -k_1 e_1^2 - \frac{k_3}{k_2}\sin^2 e_3 \leq 0 \tag{45a}$$

$$\ddot{V}(t) = -2k_1 e_1 \dot{e}_1 - \frac{2k_3}{k_2}\dot{e}_3 \sin e_3 \cos e_3 \tag{45b}$$

All of the parameters in equation (45b) are bounded, so we can know that the $\ddot{V}$ is bounded. Based on the previous result, we know that $\dot{V}$ is uniformly continuous. Using Barbalat's Lemma we know that $\dot{V} \to 0$ when $t \to \infty$. Combine with the differential Lyapunov function (45a), we can realize that e_1 and e_3 will converge to 0 under the previous domain we defined $\mathrm{D} = \{e \in R^3 | -\pi < e_3 < \pi, e_3 \neq 0\}$. To prove the theorem, we have to show that $e_2 \to 0$ when $t \to 0$. Using equation (13) and (18) we can derivative the following equation

$$\dot{e}_3 = -k_2 v_r e_2 - k_3 \sin e_3 \tag{46}$$

After getting the equation (46), we take the time derivative of equation (46)

$$\ddot{e}_3 = -k_2 v_r \dot{e}_2 - k_3 \dot{e}_3 \cos e_3 \tag{47}$$

We have already shown that the $[\dot{e}_1, \dot{e}_2, \dot{e}_3]$ are bounded. e_3 is also bounded. k_2 and k_3 are constant and v_r is the trajectory we designed, so we can assume that it is bounded. From this derivation we can know that $\ddot{e}_3$ is bounded, which means $\ddot{e}_3 \in \mathrm{L}_\infty$, therefore we can obtain that $\dot{e}_3$ is uniformly continuous. Using Barbalat's Lemma, we can obtain that $\dot{e}_3 \to 0$ when $\mathrm{t} \to \infty$. Using equation (46), because $\dot{e}_3$ and e_3 will converge to 0 when time approached infinity, we can derivate that $-k_2 v_r e_2$ will converge to 0 as time approach infinity. Therefore, e_2 is bounded. Combining previous knowledge, we know that e_1, e_2 and e_3 will converge to 0 as $\mathrm{t} \to \infty$. and we completed the proof of the theorem that the tracking error will converge to 0 when $\mathrm{t} \to \infty$.

In sum, using the above trajectory tracking control law and the longitude slippage adaptive update law and the angle orientation observer, the equilibrium point $[e_1, e_2, e_3]^T = [0\ 0\ 0]^T$ is asymptotically stable which means that we can track the trajectory p_r from any arbitrary pose p (0).

2.3.5 RH Control Scheme

In order to optimize the control gain selection, we implement the RH control scheme instead of the traditional controller, therefore, we can obtain a faster converge rate of the error. RH formulation for regulation is shown in **Appendix [10]**. Receding control is also very useful in formation control. Using RH control the converge rate become faster.

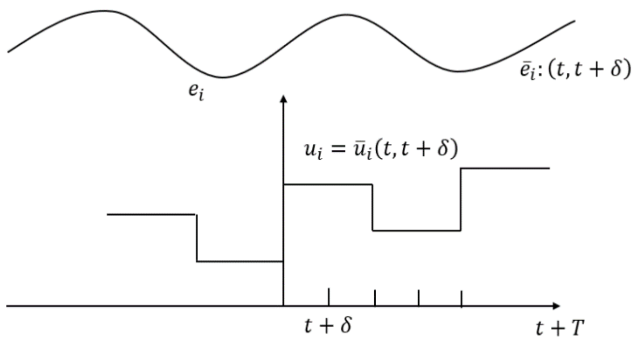


Figure 8 General RH scheme

Because we want to implement the RH control in our second and third tracked mobile robots, we have to define a control input $u_i(t)=[v_i(t) \quad \omega_i(t)]^T$ which is time-varying. In the figure general RH scheme the parameter δ means sampling time period and the other parameter T denotes the predictive horizon length. $\bar{\bar{u}}_i:(t,t+T),\bar{\bar{e}}_i:(t,t+T)$ are the control input and the predict error states respectively. The process of implementing RH control is shown as following figure 9.

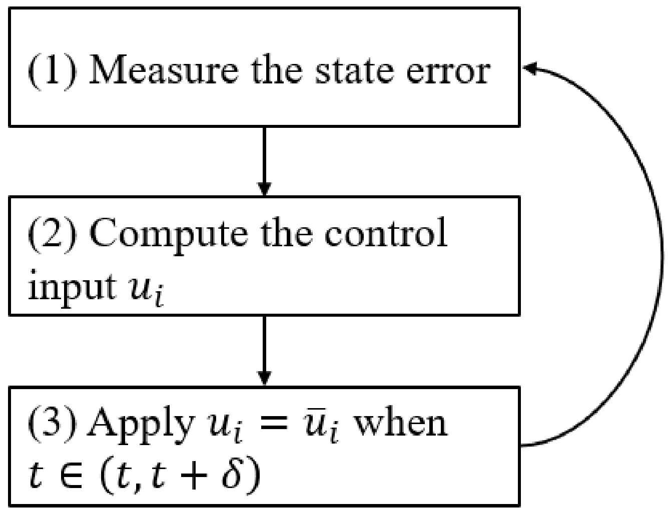


Figure 9. General process of RH scheme

To be more specific, the first step of RH control is that we have to observe the state error in the beginning. The control input is computed from the predictive horizontal $(t,t+T)$. After finding a control input, we can predict the future error based on the control input and the knowledge about the kinematic model. Applying this control input until current time equal to $t+\delta$. Note that δ is much smaller than T. Repeat the above process until finish tracking.

The process of RH scheme is trying to minimize the sum of a terminal state penalty function $g\big(e_i(t+T)\big)$ and cost function $L\big(e_i(t).u_i(t)\big)$ which is defined as follow

$$g\big(e_i(t+T)\big)=\frac{1}{2}\Big(e_{1_i}^2(t+T)+e_{2_i}^2(t+T)+e_{3_i}^2(t+T)\Big) \tag{48}$$

$$L\big(e_i(t),u_i(t)\big)=e_i(t)^T Q e_i(t)+u_i(t)^T R u_i(t) \tag{49}$$

First, $g\big(e_i(t+T)\big)$ should be continuous differentiable, and positive-definite. Second, $Q>0$ which means Q is a positive definite matrix, and R ≥ 0, which is semi-positive definite. To implement RH scheme to our system, we have to choose the symmetric weight matrices Q, R and $g\big(e_i(t+T)\big)$

$$Q=\begin{bmatrix} q_{x_i} & 0 & 0 \\ 0 & q_{y_i} & 0 \\ 0 & 0 & q_{\theta_i} \end{bmatrix} \tag{50a}$$

$$R=0 \tag{50b}$$

$$g\big(e_i(t+T)\big)=\frac{1}{2}\Big(e_{1_i}^2(t+T)+e_{2_i}^2(t+T)+e_{3_i}^2(t+T)\Big) \tag{50c}$$

Combining Q, R and L we can get the following equation

$$L\big(e_i(t),u_i(t)\big)=q_{x_i}(t)e_{x_i}^2(t)+q_{y_i}(t)e_{y_i}^2(t)+q_{\theta_i}(t)e_{\theta_i}^2(t) \tag{50d}$$

Define the terminal state region Ω

$$k_{1_i}(t)\geq q_{x_i} \tag{51a}$$

$$k_{3_i}(t)\geq q_{\theta_i} \tag{51b}$$

$$e_{2_i}(t+T)=0 \tag{51c}$$

The state error e_1 will converge with a exponential rate $k_{1_i}(t)$ and the state error e_3 will converge with a exponential rate $k_{3_i}(t)$.

CHAPTER 3. SIMULATION RESULT

This section is organized as the following description. First, we will show the assumption of this simulation. And then we will show the performance of the adaptive trajectory tracking control performance, also show the performance of the observer. After showing the adaptive control, we will show the simulation of formation control, the leader of the formation also suffers from the slip and cannot measure orientation angle directly. Therefore, the leader is using adaptive control, and we will also show its performance. And then we will provide the simulation result of formation control with the RH control scheme.

Note that because we implement two journal papers in our simulation, so our simulation will be different from the original paper, also we use the different control gain and trajectory and plant parameters. I asked the professor, and he said, I can be different with the original papers

The video of our formation control is shown in the Note.txt. Note that we use the free app, so we cannot record for more than 10 mins. But I suppose the video can successfully show the performance of our control system. The second video shows the transform from the triangle formation to linear formation. The line formation takes times to converge, so the third video shows the convergence of the line formation. Recommend that use 8-times speed to watch these videos.

3.1 Adaptive control and Observer performance simulation

In this section, we are going to verify the performance of the tracking control law and the longitude slippage adaptive update law and the angle orientation observer designed in previous chapter.

All of the parameters we used are based on the following tracked mobile robot figure 10a. This tracked mobile robot width is 0.3 m, and its wheel radius is 0.075 m. Its maximum velocity is 0.3 m/s and the minimum turning radius is 1.2 m. The angular velocity constraint is $-0.25 \leq \omega < 0.25\ (\mathrm{rad/s})$. Our control scheme is shown in figure 10b.

Figure 10a. Tracked Mobile Robot

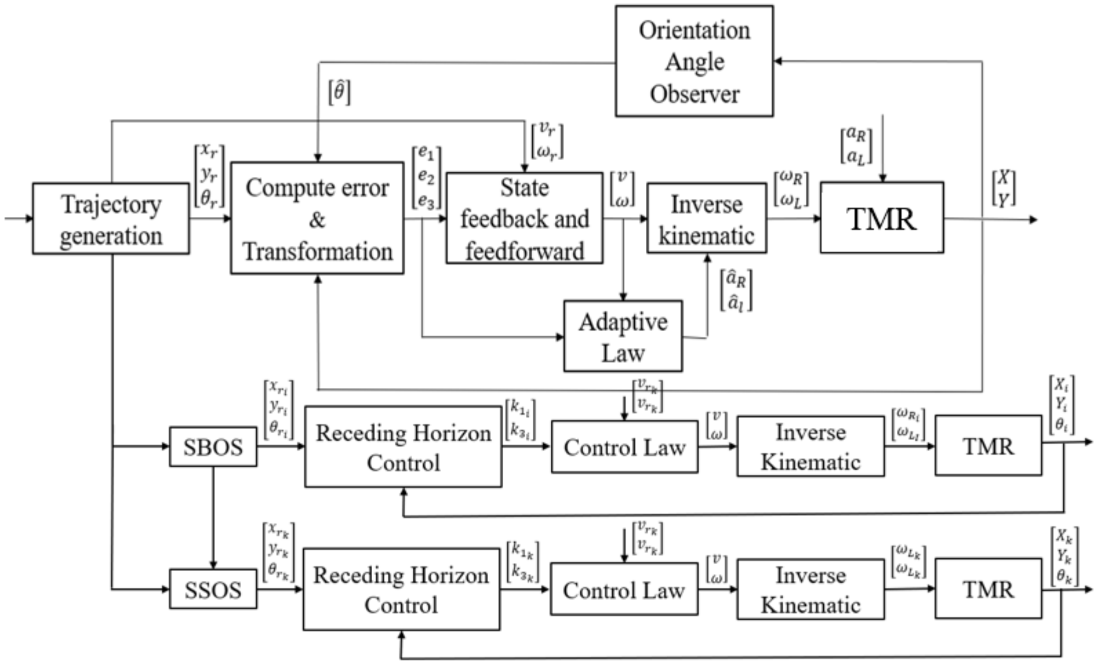


Figure 10b Control scheme of our formation control using RH scheme and adaptive trajectory tracking control

Two simple trajectories are shown in this part. One is circular trajectory and the other is linear trajectory.

The reference trajectory for line tracking is generated by substituting (53) into (52).

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$
$$y(t) = b_0 + b_1 t + b_2 t^2 + b_3 t^3 \tag{52}$$

$$x(0) = 0\,, \dot{x}(0) = 0, x(t) = 6, \dot{x}(t) = 0$$
$$y(0) = 0, \dot{y}(0) = 0, y(t) = 6, \dot{y}(t) = 0 \tag{53}$$

Here we applying a constant longitude slip $\mathrm{i_L} = 0.25, i_R = -0.35$.

The feedback gains $\mathrm{k_1}, \mathrm{k_2}$ and $\mathrm{k_3}$ are selected as 2, 10 and 2. The adaptive update gains ρ_1 and ρ_2 are chosen as 0.05 and 0.05. The angle orientation observer gain L is selected as 100. Observer gain is usually chosen ten times larger than feedback gains, because we have to reconstruct the true state as fast as possible.

To show this control law can track the trajectory successfully, we let the initial position (x_0, y_0, θ_0) equal to $(-0.15, -0.13, 0)$.

The linear trajectory is generated by equation (52) and parameters (53). The unit of x-axis in the X-Y plot is m, and in the y-axis is m. The unit of x-axis in the state error e_1, e_2 ,and e_3 are time (s) and the y-axis are m, m, rad. In the velocity and angular velocity plot the units are m/s – t and rad/s – t separately. In the figure of slip, the y-axis unit is time (s), however, in the x-axis because slip is a ration, it does not have units. In the observer plot, the unit in x-axis is time (s) and the y-axis unit is radian (rad).

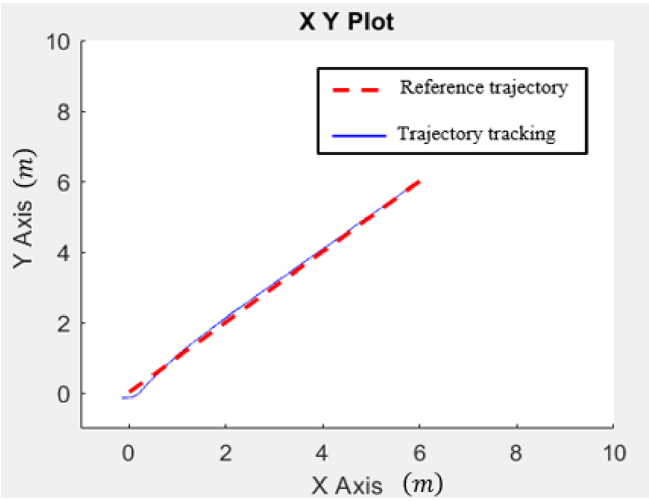


Figure 11a. Linear trajectory tracking X-Y graph

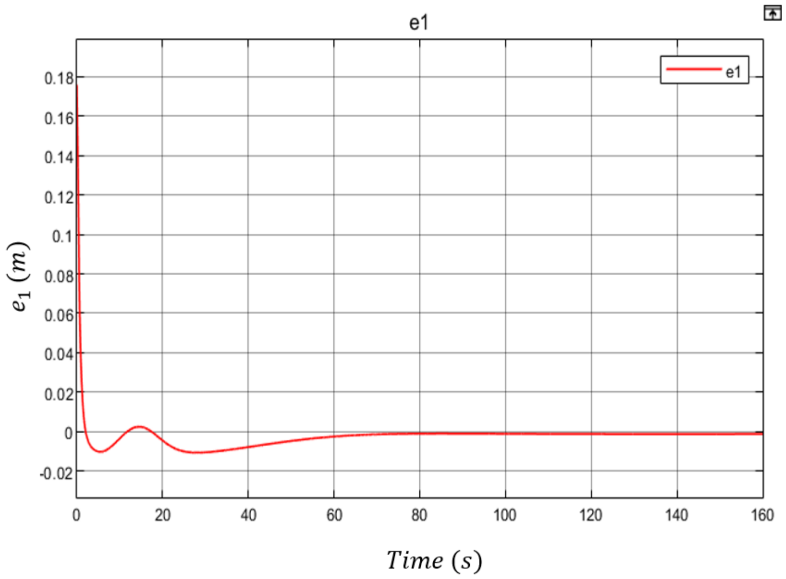


Figure 11b. Linear trajectory tracking state error e_1

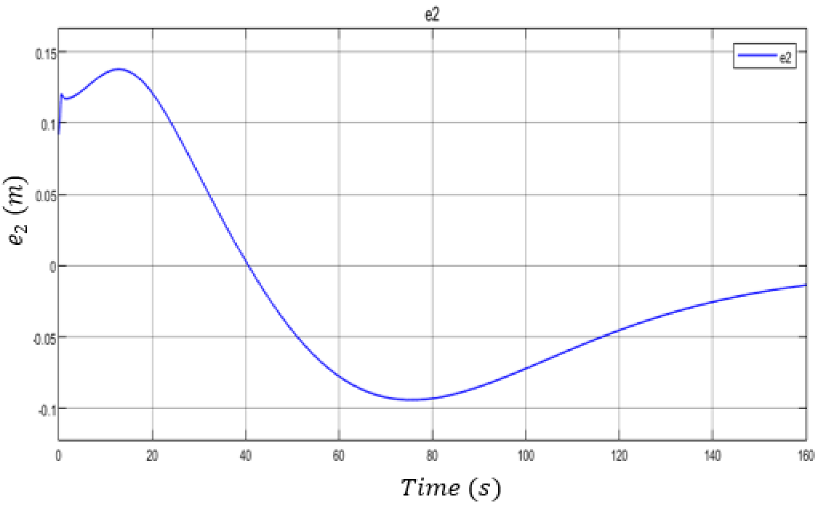


Figure 11c. Linear trajectory tracking state error e_2

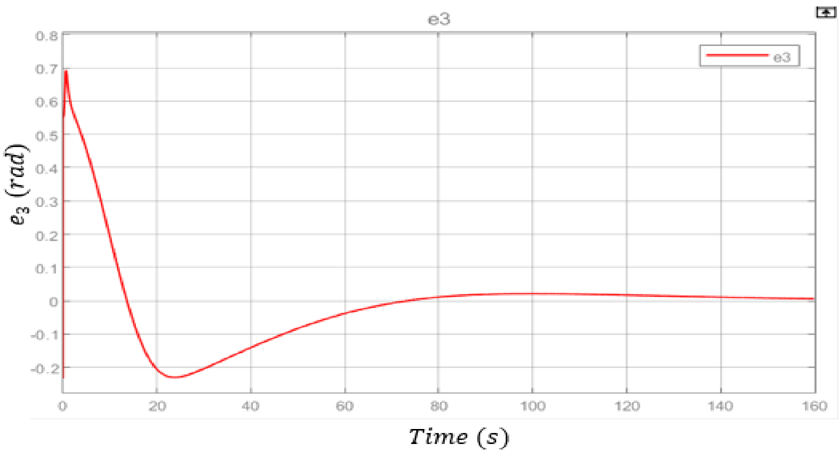


Figure 11d. Linear trajectory tracking state error e_3

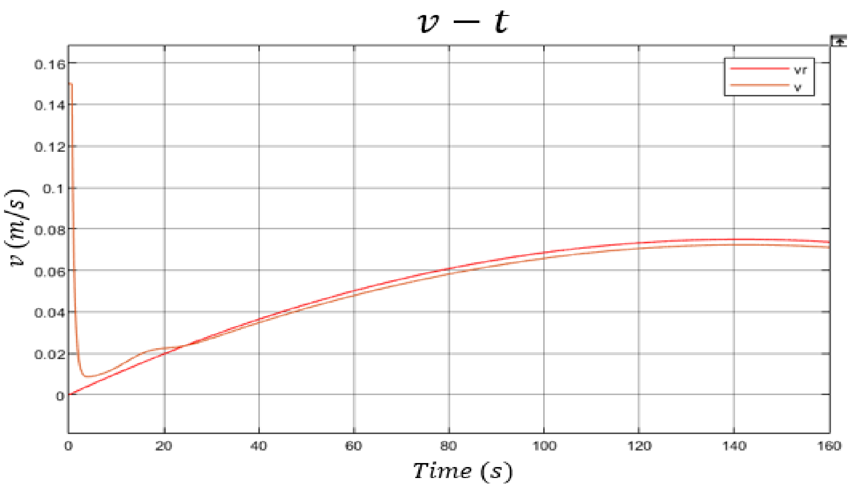


Figure 11e. Linear trajectory tracking reference velocity and velocity command

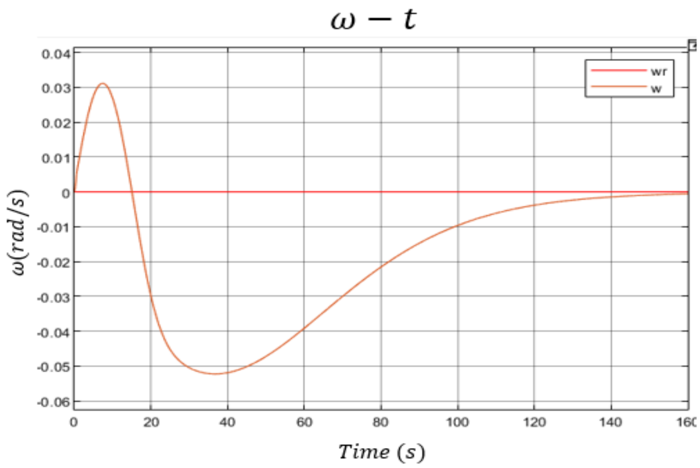


Figure 11f. Linear trajectory tracking reference angular velocity and angular velocity command

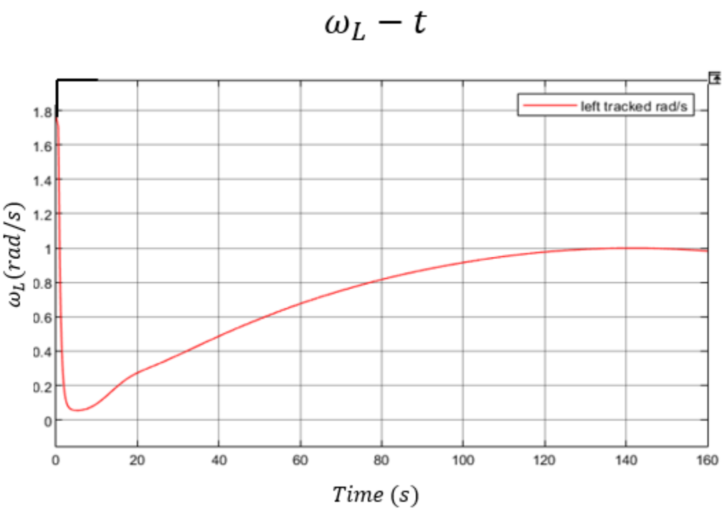


Figure 11g. Linear trajectory tracking reference left tracked speed and actual left wheel speed

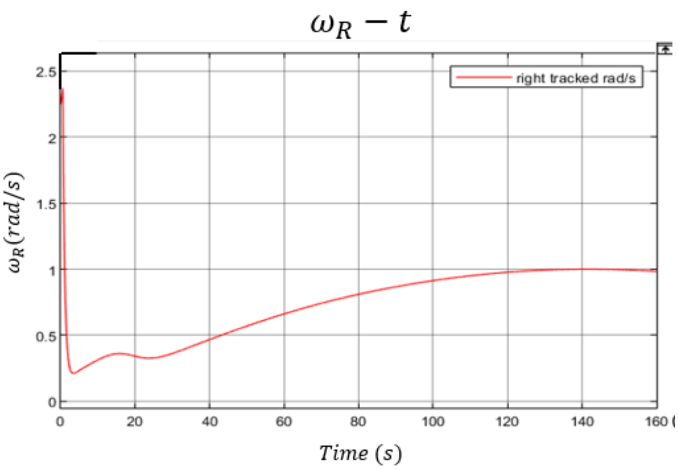


Figure 11h. Linear trajectory tracking reference right tracked speed and actual right wheel speed

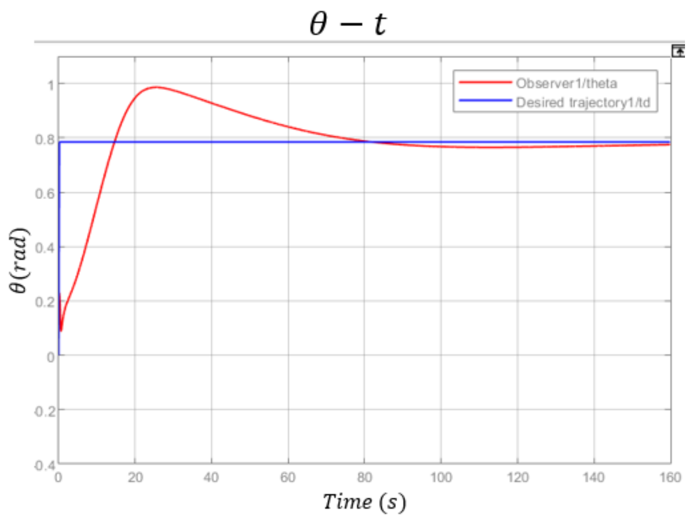


Figure 11i. Linear trajectory tracking Orientation angle observer

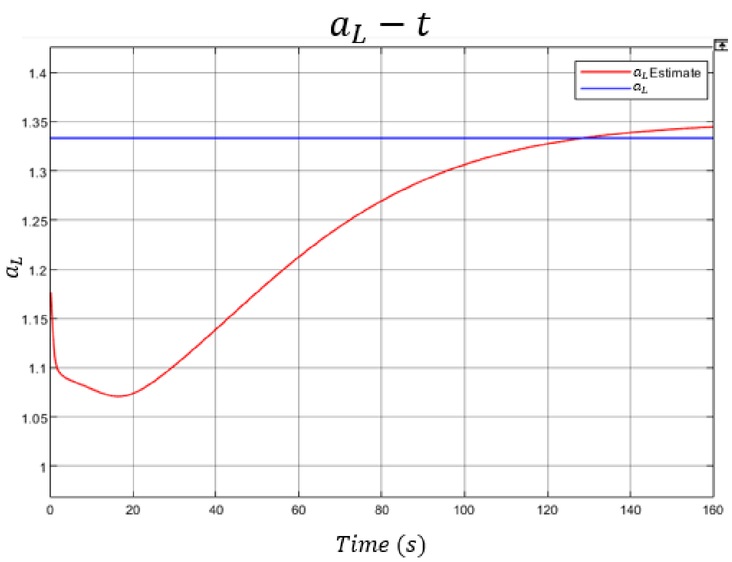


Figure 11j. Linear trajectory tracking Adaptive update law of left wheel slippage

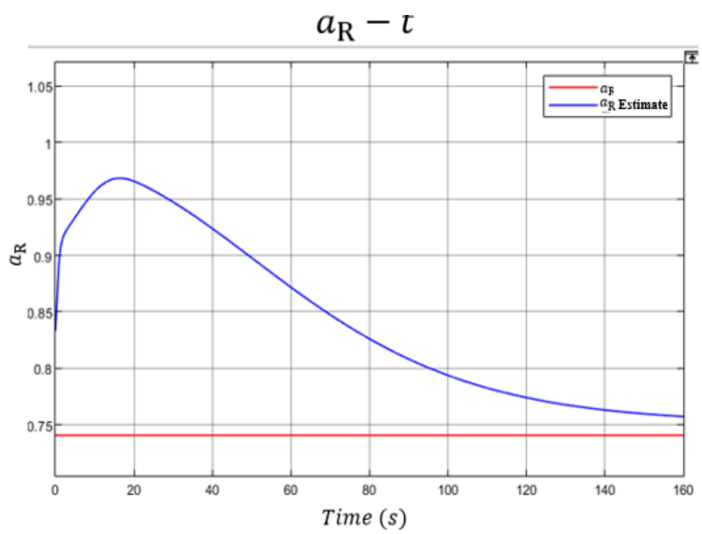


Figure 11k. Linear trajectory tracking Adaptive update law of right wheel slippage

We will show the performance of tracking a circular trajectory in the following part. The reference trajectory for circular tracking is generated by substituting (55) into (54).

$$\begin{cases} x(t) = Rsin(\varphi(t)) \\ y(t) = b' - Rcos(\varphi(t)) \\ \varphi(t) = \varphi_0 + \varphi_1 t + \varphi_2 t^2 + \varphi_3 t^3 \end{cases} \tag{54}$$

$$\varphi(0) = 0, \dot{\varphi}(0) = 0, \varphi(t) = \pi, \dot{\varphi}(t) = 0 \tag{55}$$

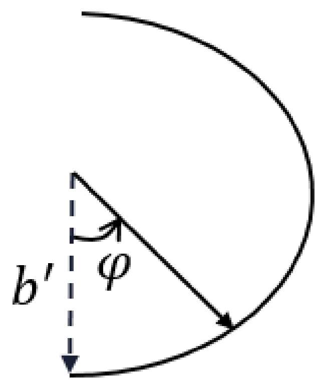


Figure 12. Circular trajectory schematic.

The feedback gains k_1, k_2 and k_3 are selected as 2, 10, and 2. The adaptive update gains ρ_1 and ρ_2 are chosen as 0.05 and 0.05. The angle orientation observer gain L is selected as 100. Observer gain is usually chosen ten times larger than feedback gain, because we have to reconstruct the true state as fast as possible.

To show this control law can track the trajectory successfully, we let the initial position (x_0, y_0, θ_0) equal to $(-0.05, 0.02, 2.9)$. The unit of x-axis in the X-Y plot is m, and in the y-axis is m. The unit of the x-axis in the state error e_1, e_2 ,and e_3 is time (s) and the y-axis are m, m, rad. In the velocity and angular velocity plot the units are m/s – t and rad/s – t separately. In the figure of slip, the y-axis unit is time (s), however, in the x-axis because slip is a ration, it does not have units. In the observer plot, the unit in the x-axis is time (s) and the y-axis unit is radian (rad).

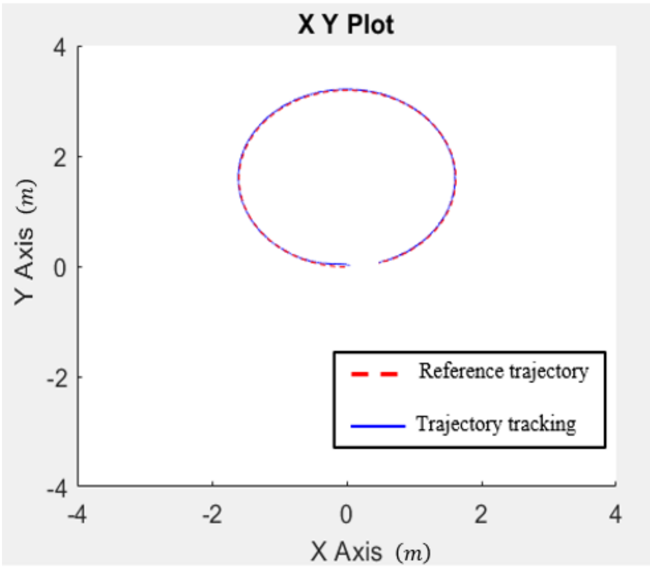


Figure 13a. Circular tracking trajectory X-Y graph

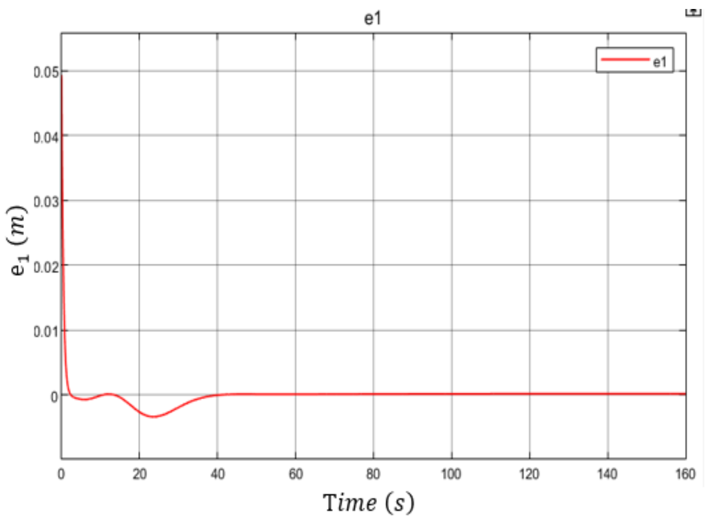


Figure 13b. Circular trajectory tracking state error e_1

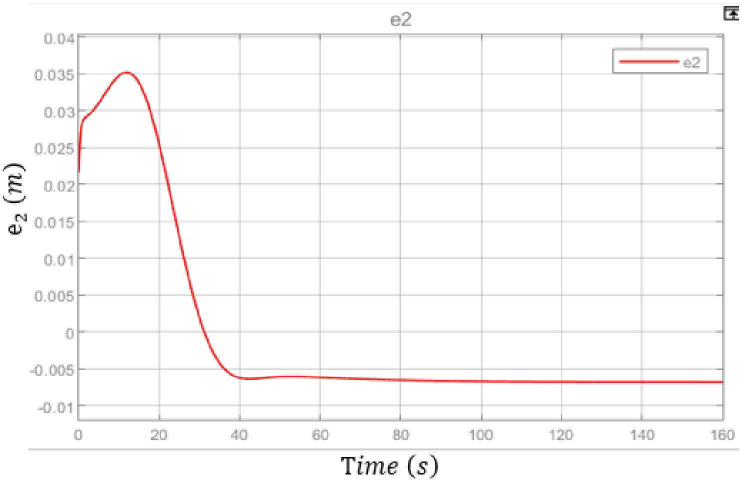


Figure 13c. Circular trajectory tracking state error e_2

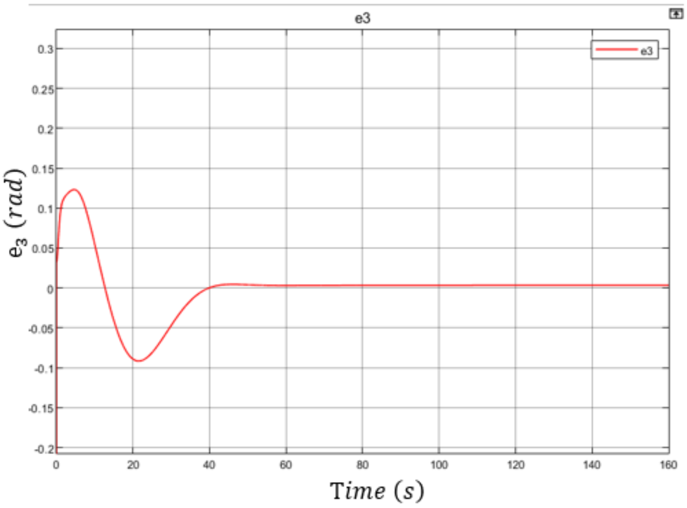


Figure 13d. Circular trajectory tracking state error e_3

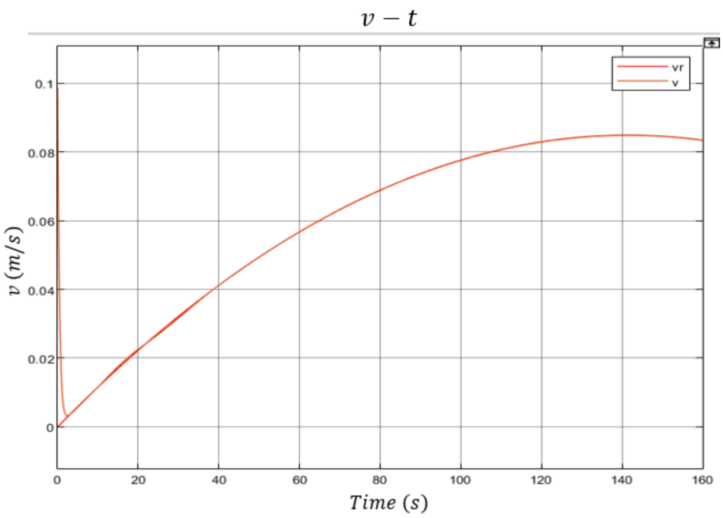


Figure 13e. Circular trajectory tracking reference velocity and velocity command

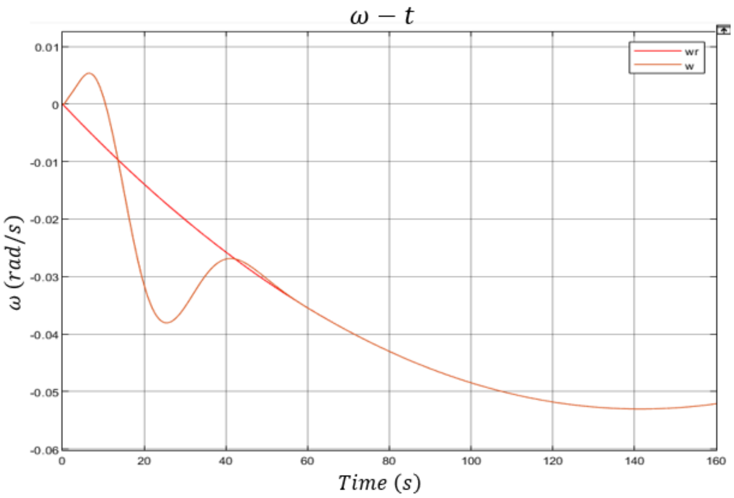


Figure 13f. Circular trajectory tracking reference angular velocity and angular velocity command

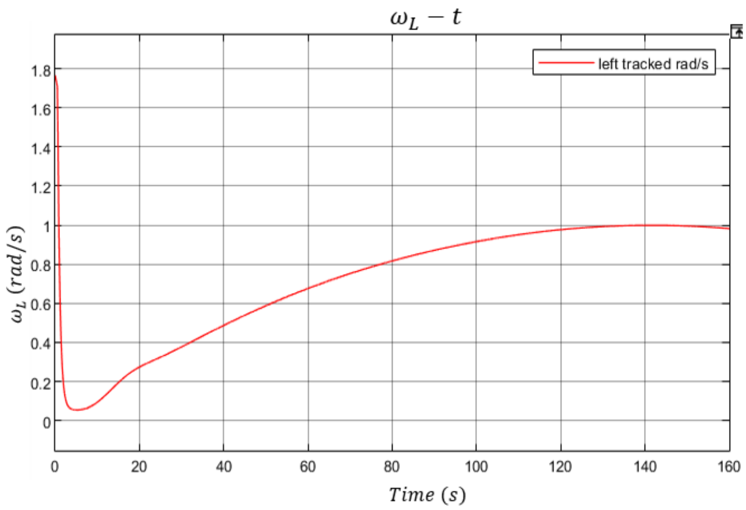


Figure 13g. Circular trajectory left tracked speed

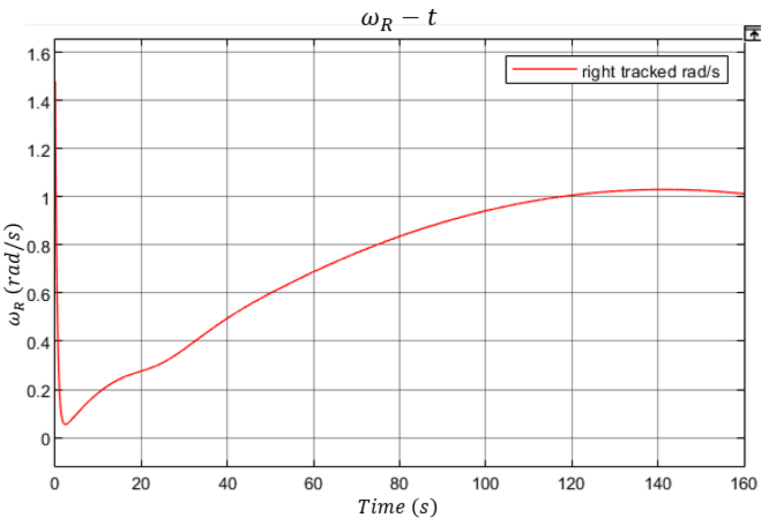


Figure 13h. Circular trajectory right tracked speed

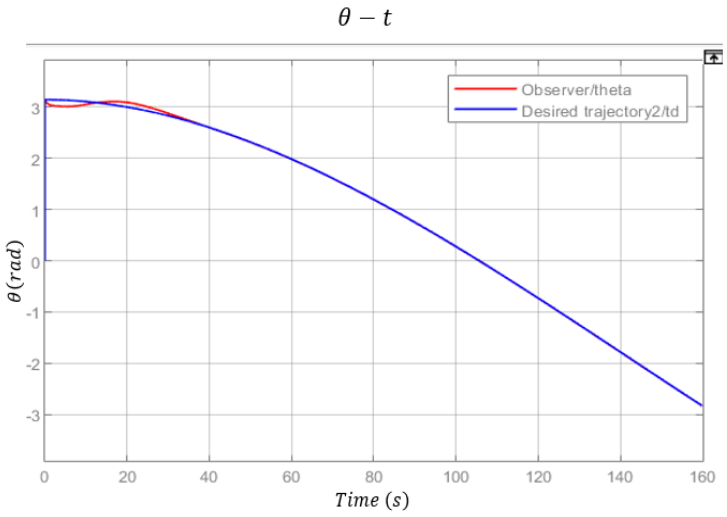


Figure 13i. Circular trajectory tracking Orientation angle observer

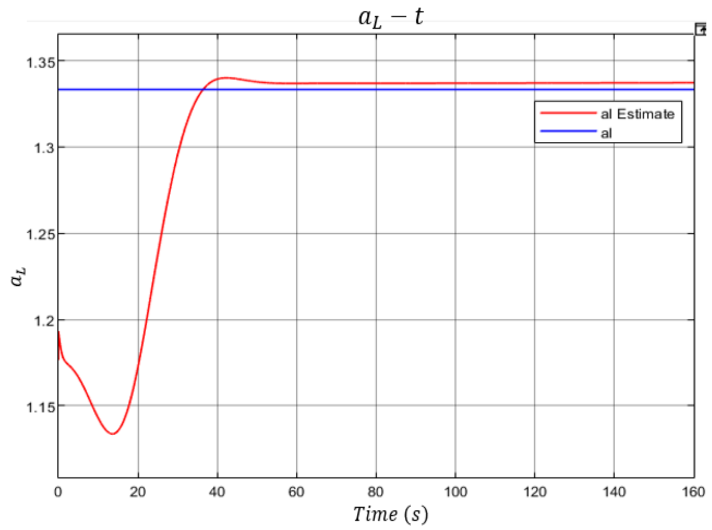


Figure 13j. Circular trajectory tracking Adaptive update law of left wheel slippage

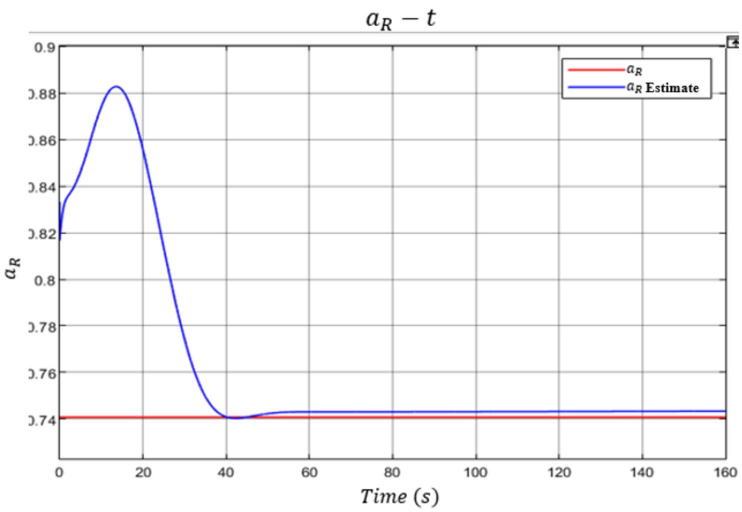


Figure 13k. Circular trajectory tracking Adaptive update law of right wheel slippage

Based on above simulation, we can know that the adaptive control algorithm can track the trajectory even with extremely large unknow slippage and unknown orientation angle.

3.2 Formation Control simulation

In this part, we will show the result of our formation control simulation. First, in this simulation, we have one leader and two followers. The first follower, which forms a two-robot formation with SBOS, follows the leader with distance $l_{i,j}^{d}$ which is 1.5 m, and the bearing angle $\beta_{i,j}^{d} = \frac{2\pi}{3}$, $\varphi_{i,j}^{d} = 0$. The second robot follows the leader and the follower, which are three-robot formation called SSOS, with the distances which are $l_{i,j}^{d} = 1.85\ m$ and $l_{i,k}^{d} = 1.85$ m, and the bearing angle $\beta = 0$. The initial position of the first follower is $(-4 \quad -3 \quad \pi + 0.2)$ and the second follower's initial position is $(-3 \quad -1 \quad 0.2)$. In this simulation, after time > 60 second. The formation changes from triangle formation to the linear formation. The parameters of SBOS in line formation are $\beta_{i,j}^{d} = \frac{5\pi}{4}$, $\varphi_{i,j}^{d} = 0$, and $l_{i,j}^{d} = 0.3$. The parameters of SSOS in line formation are $\beta_{i,j}^{d} = 0$, $l_{i,k}^{d} = 0.3$, and $l_{i,j}^{d} = 0.6$.we design a simple trajectory which is followed by the leader the trajectory shows as follows.

$$x(t) = 0.05t \tag{56a}$$
$$y(t) = 0.05t \tag{56b}$$
$$\theta(t) = \frac{\pi}{4} \tag{56c}$$

From this trajectory, we can obtain that the desire velocity v_r and the desire angular velocity ω_r are 0.05 m/s and 0 rad/s respectively.

Secondly, we will discuss the parameters $q_1, q_2, and\ q_3$ in the weight matrix Q which is used in receding horizon control also called model predictive control. In the first follower, the weight matrix Q and R are formed as follows

$$Q = \begin{bmatrix} q_1 & 0 & 0 \\ 0 & q_2 & 0 \\ 0 & 0 & q_3 \end{bmatrix} \tag{57a}$$

$$R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{57b}$$

In the first follower the weighting matrix's parameters $q_1, q_2, and\ q_3$ are 0.02, 0.02 and 0.015 respectively. In the second follower which follows the leader and the first follower at the same time, the parameters $q_1, q_2, and\ q_3$ in this robot are 0.01, 0.1 and 0.5 respectively. The weighting matrix R in the first follower and second follower are 0. Then, the parameters in receding horizon control are predict horizon = 0.1 s and the sampling time = 0.01 s. After introducing all of the parameters in formation control, we will show our result.

First, because we combine the adaptive control and state observer in our leader robot, and the leader robot initial position is $(-0.15 \quad -0.13 \quad 0)$ the units are meter, meter and radian separately. The slip of tracked mobile robot is assumed as 0.25 and -0.35. The adaptive update gains ρ_1 and ρ_2 which are used to solve the slip issue are chosen as 0.05 and 0.05. The control gains k_1, k_2, and k_3 are selected as 2, 10, and 2. The angle orientation observer gain L is selected as 100. We will show the result of the leader as the following figure.

Note that the paper has two experiment, and we combine the two experiment to one, therefore, it is the same with the paper.

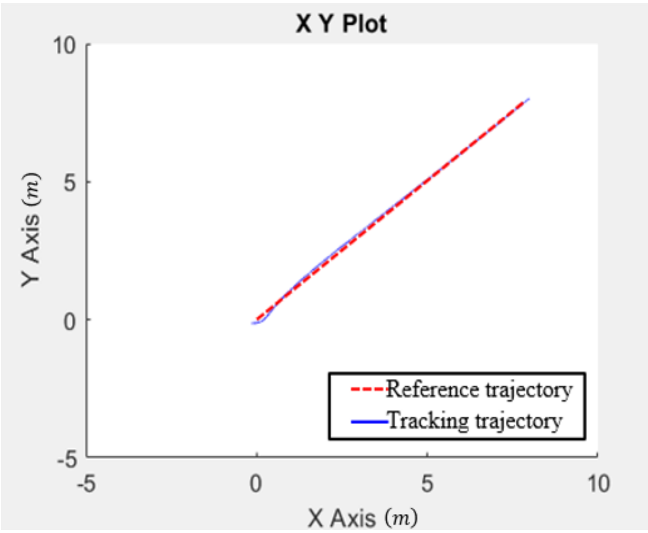
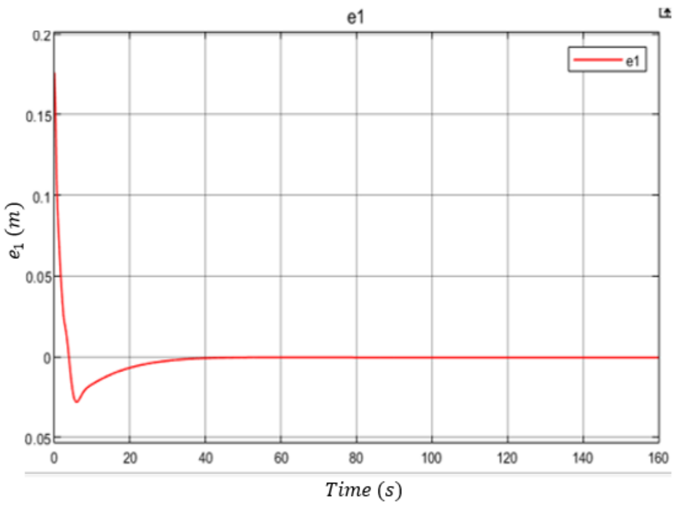


Figure 14a. Leader of formation control trajectory tracking



c

Figure 14b. Leader of formation control trajectory tracking state error e_1

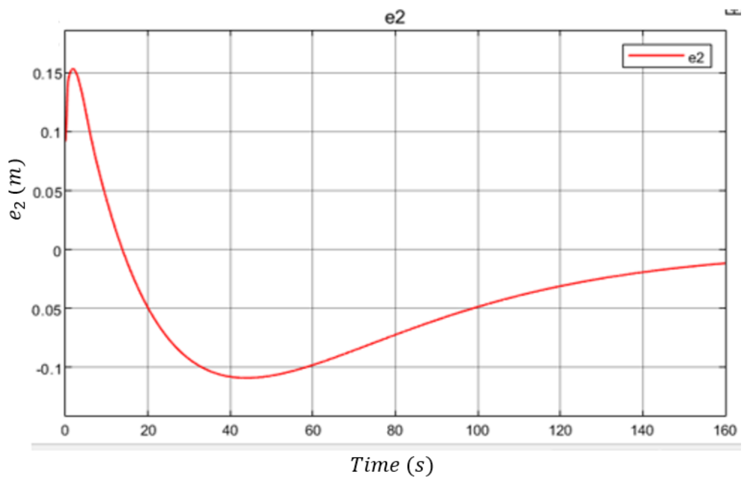


Figure 14c. Leader of formation control trajectory tracking state error e_2

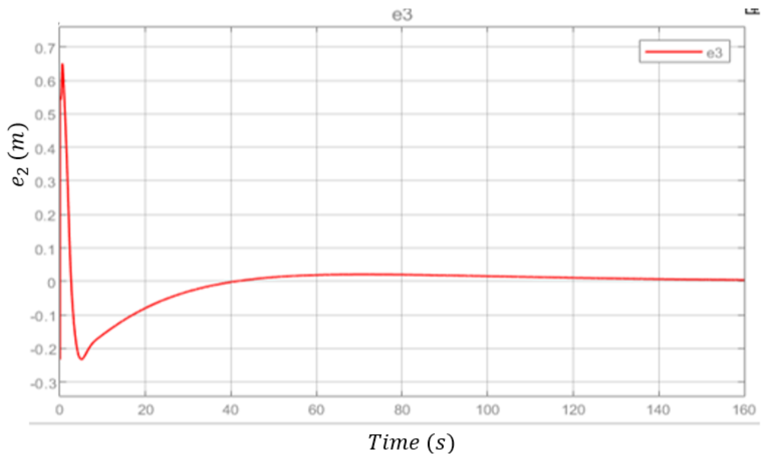


Figure 14d. Leader of formation control trajectory tracking state error e_3

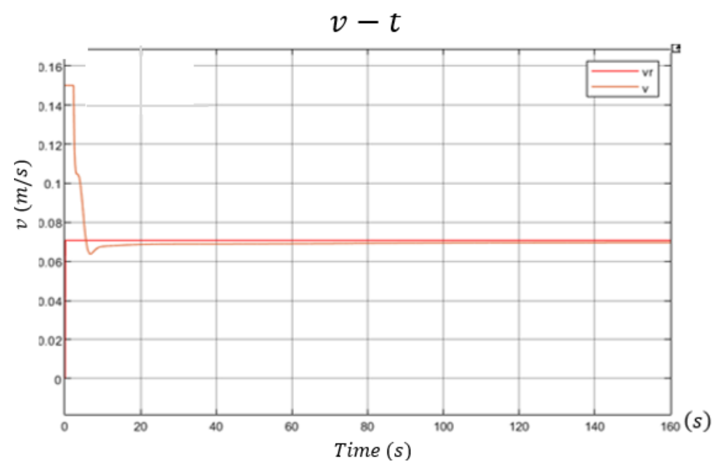


Figure 14e. Leader of formation control reference velocity and velocity

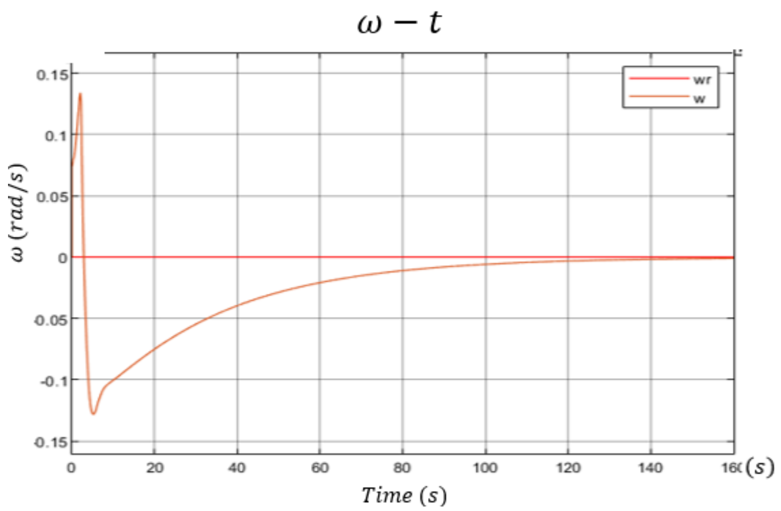


Figure 14f. Leader of formation control reference angular velocity and angular velocity

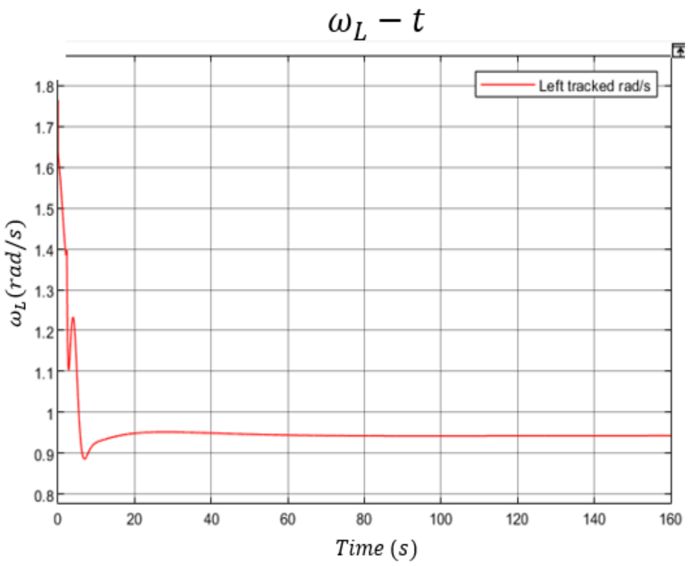


Figure 14g. Leader of formation control left tracked speed

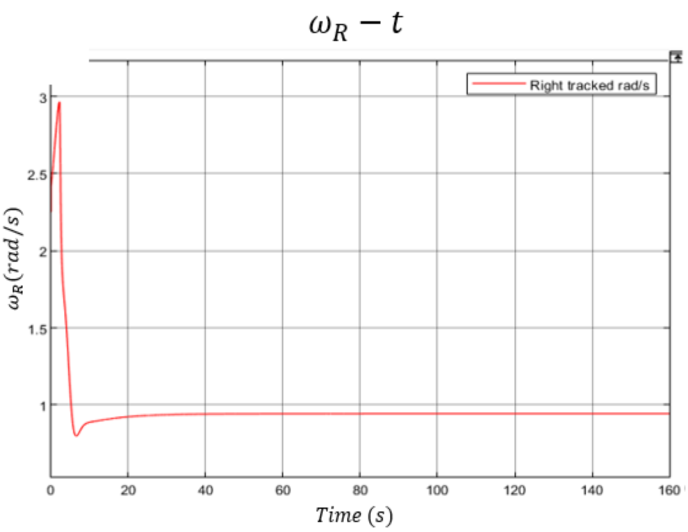


Figure 14h. Leader of formation control right tracked speed

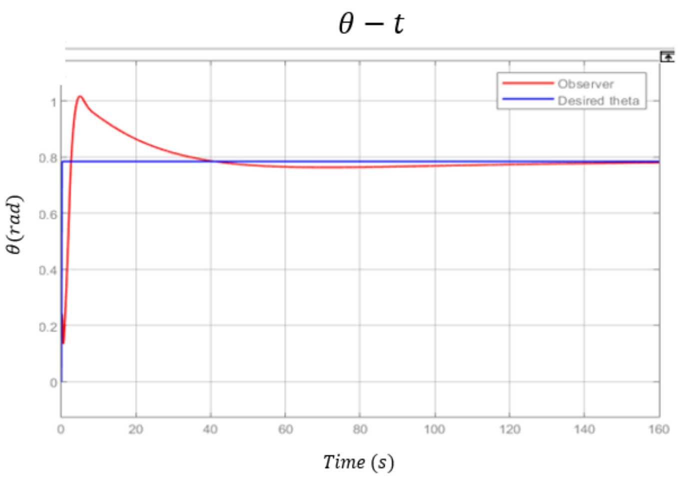


Figure 14i. Leader of formation control orientation angle and observer orientation angle

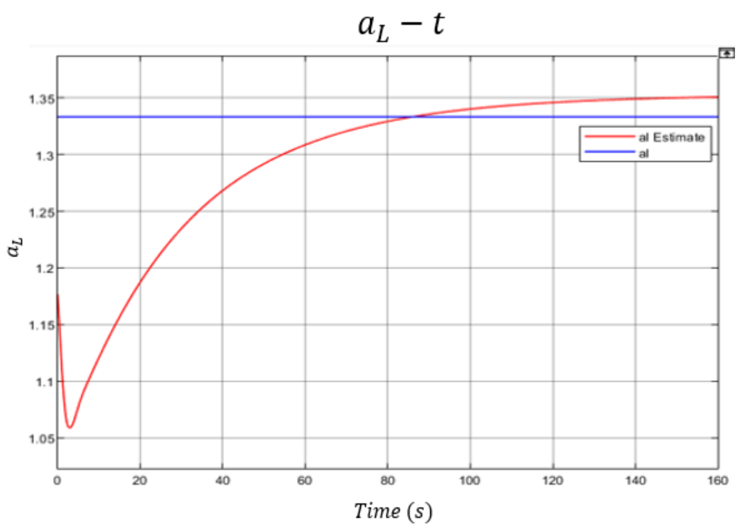


Figure 14j. Leader of formation control left tracked slip

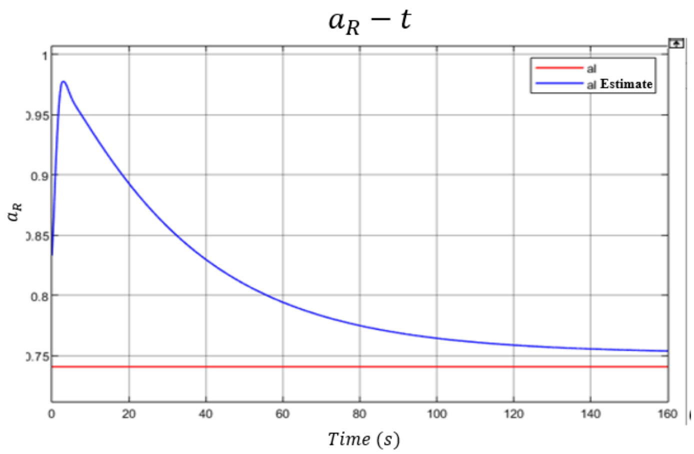


Figure 14k. Leader of formation control right tracked slip

After showing the leader of the tracked mobile robot, we will show the performance of the first follower which only follows the leader. The trajectory of the first follower is generated from the reference trajectory of the leader. Note that the initial condition is $(-4, -3\ ,\ 3.3)$.

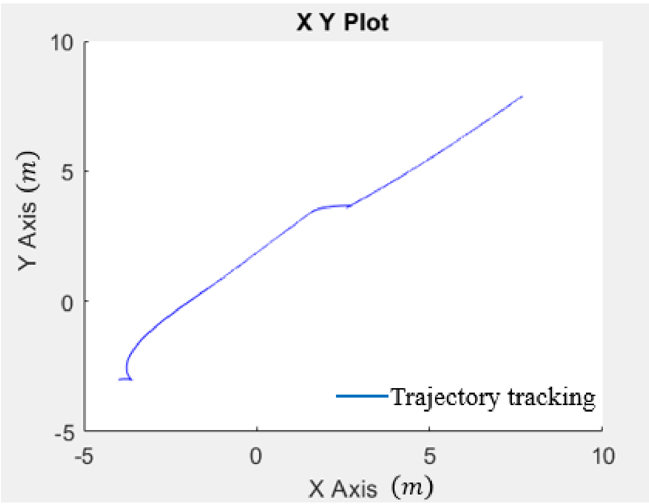


Figure 15a. SBOS follower of formation control trajectory tracking

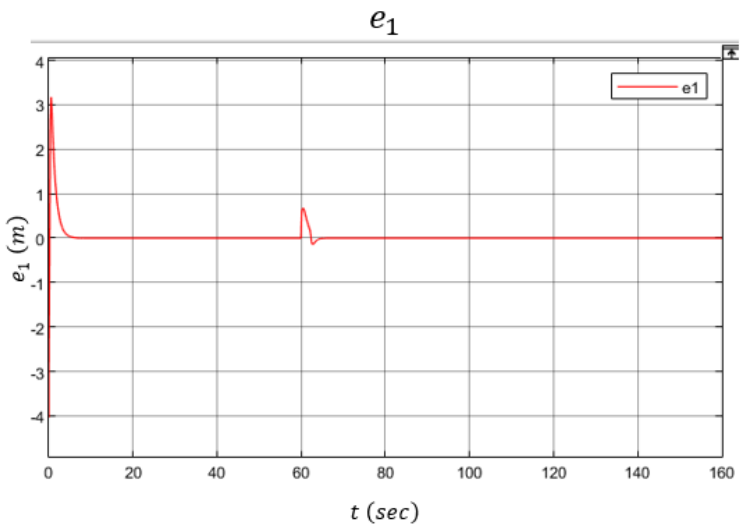


Figure 15b. SBOS follower of formation control state error e_1

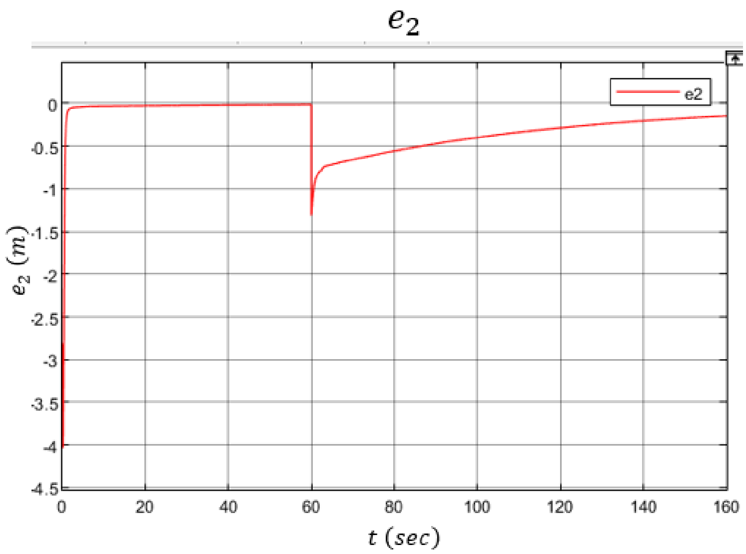


Figure 15c. SBOS follower of formation control state error e_2

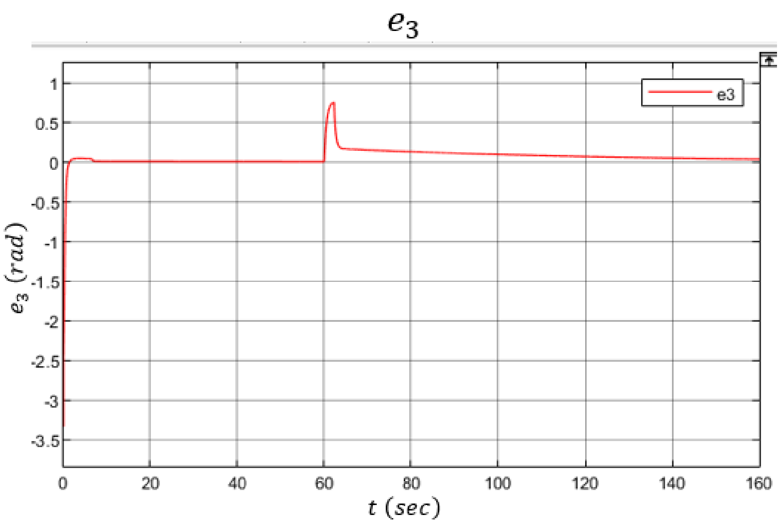


Figure 15d. SBOS follower of formation control state error e_3

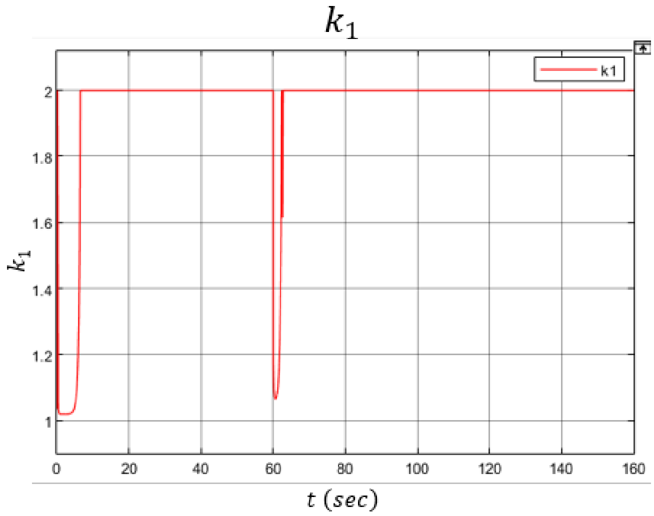


Figure 15e. SBOS follower of formation control gain k_1

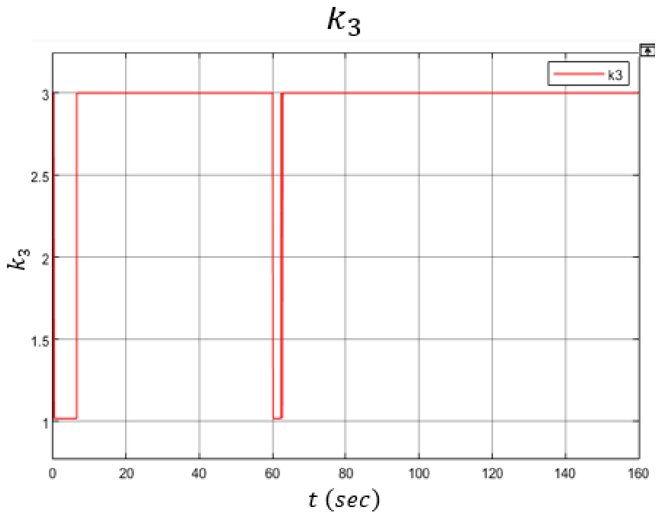


Figure 15f. SBOS follower of formation control gain k_3

The reference paper suggests that if the state error e_1 is much smaller than the state error e_2 the control gain will become extremely large; therefore, it suggests that if the k_1 is too large, we should change the receding control to the linear control feedback. From above figure, we can know that the state error k_1 converge much faster than the others state errors, so we use the linear state feedback when this situation happens.

In the following part, we will show the velocity and angular velocity performance. Because we set a very large initial error, we do not make any kinematic constraint of the followers.

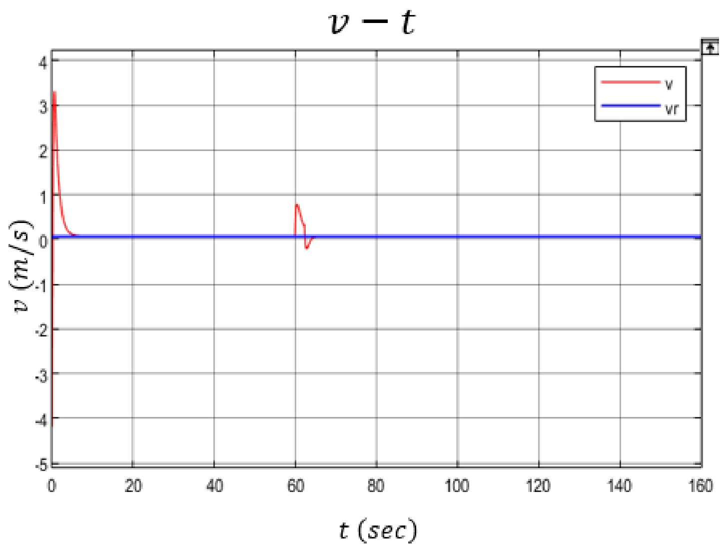


Figure 15g. SBOS follower of formation control velocity

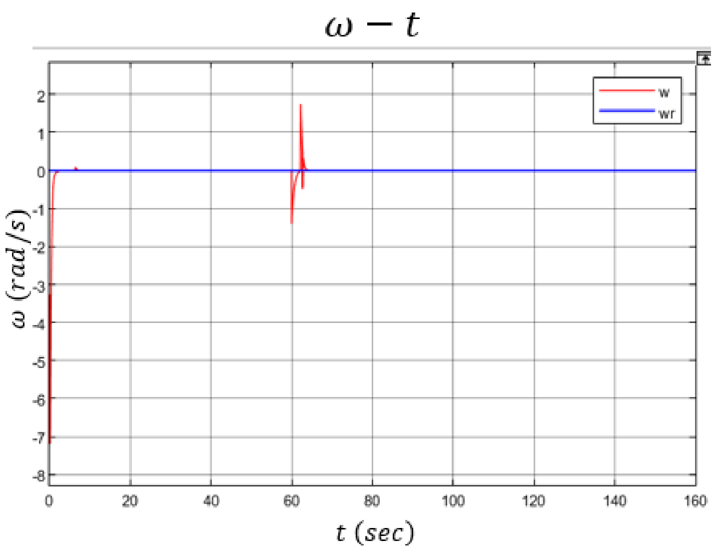


Figure 15h. SBOS follower of formation control angular velocity

In the next part, we will show the second robot, which is using SSOS formation, tracking performance, after showing the second follower performance, we will show the converge of the formation control. Note that the initial condition is $(-3,\ 0.5,\ 0.2)$.

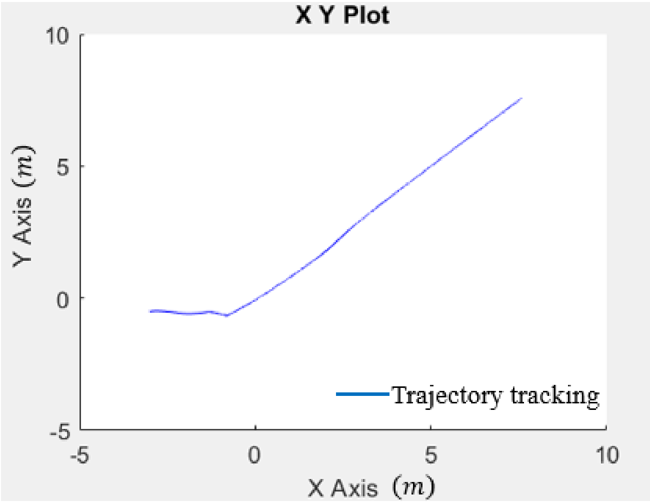


Figure 16a. SSOS follower of formation control trajectory tracking

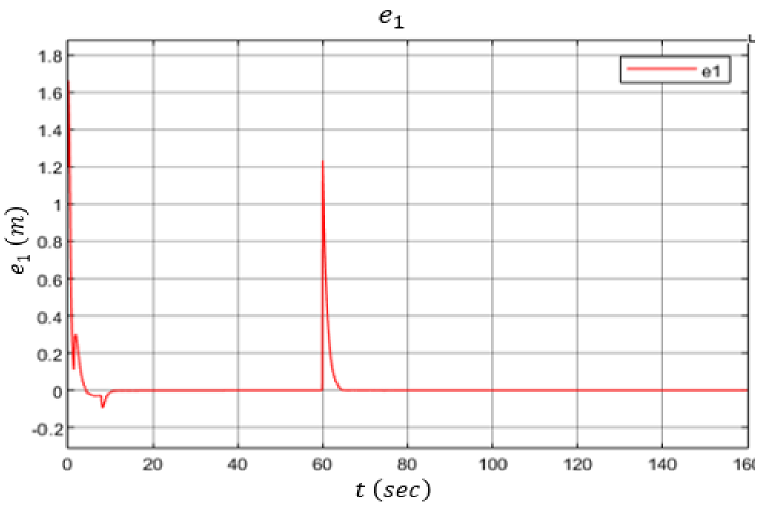


Figure 16b. SSOS follower of formation control state error e_1

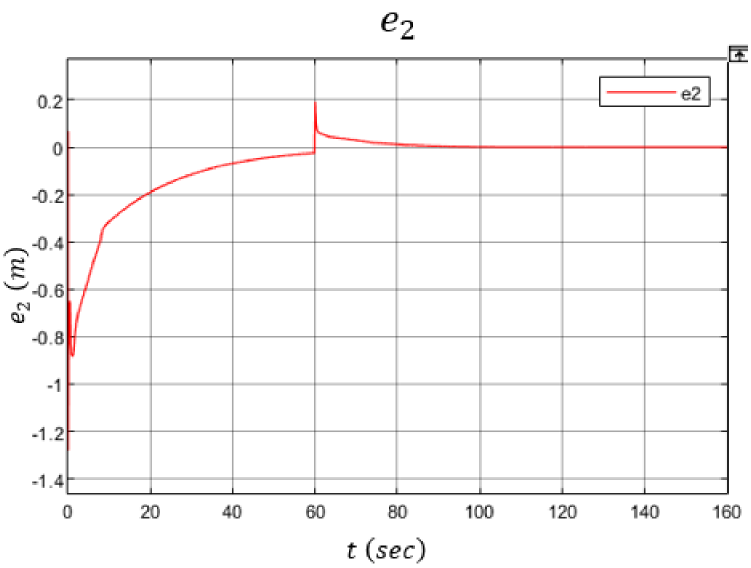


Figure 16c. SSOS follower of formation control state error e_2

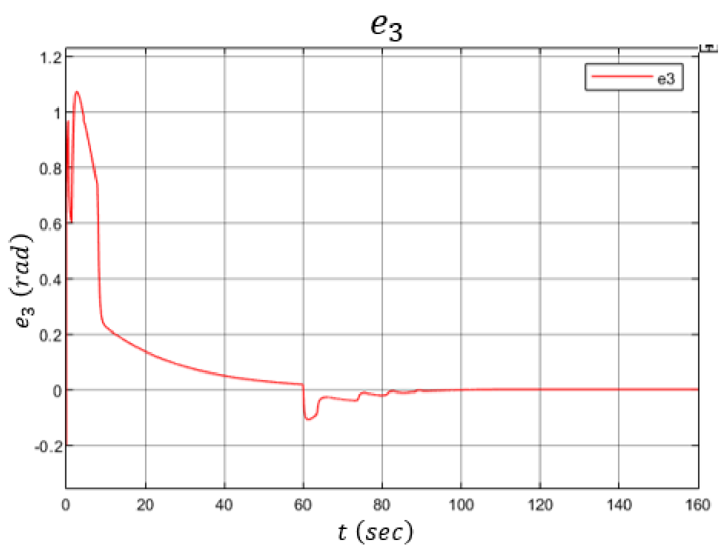


Figure 16d. SSOS follower of formation control state error e_3

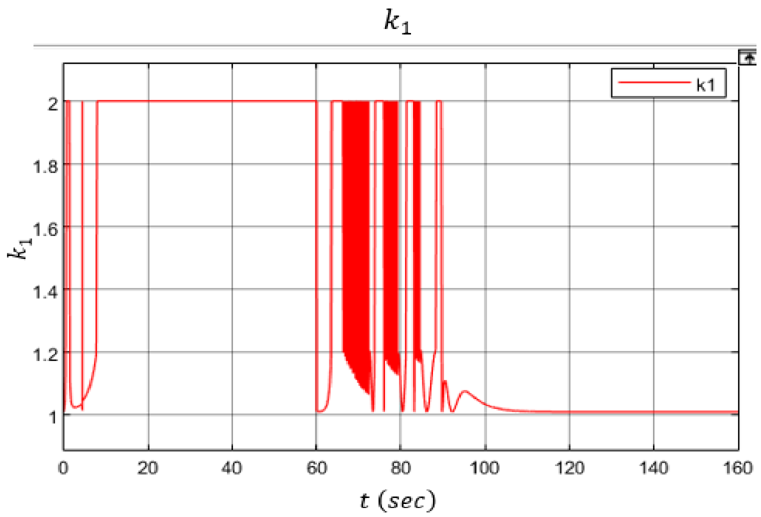


Figure 16e. SSOS follower of formation control gain k_1

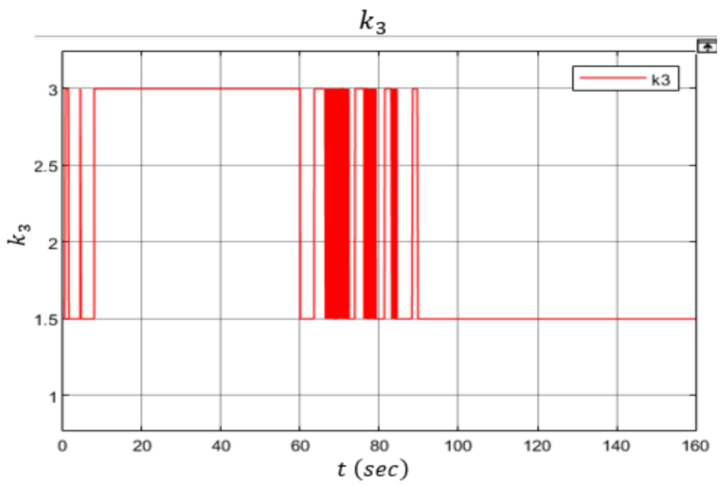


Figure 16f. SSOS follower of formation control gain k_3

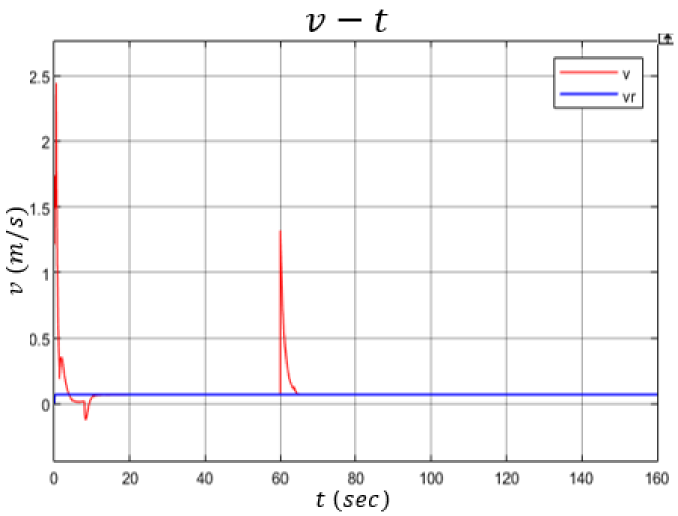


Figure 16g. SSOS follower of formation control velocity

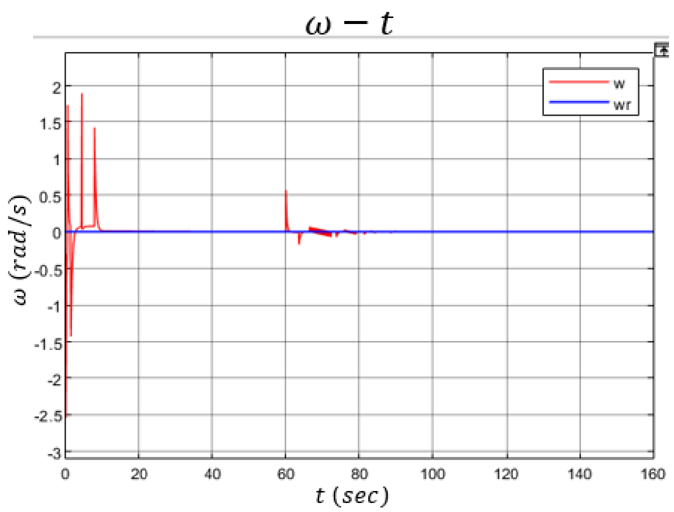


Figure 16h. SSOS follower of formation control angular velocity

In the next section, we will show the convergence of formation control. Note that d is the desire distance and l_{ij} mean the distance between robots at that moment.

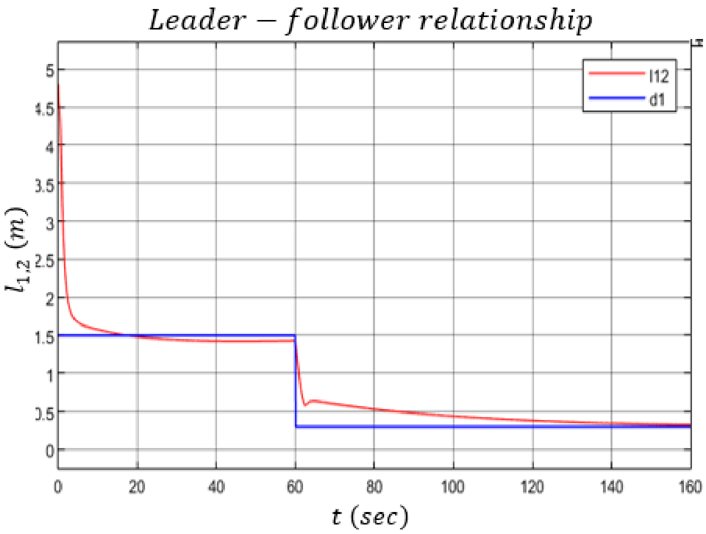


Figure 17a. Convergence of l_{12}

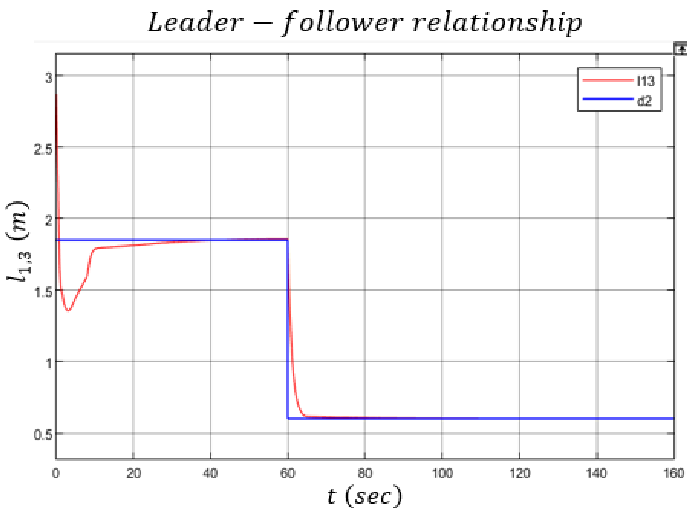


Figure 17b. Convergence of l_{13}

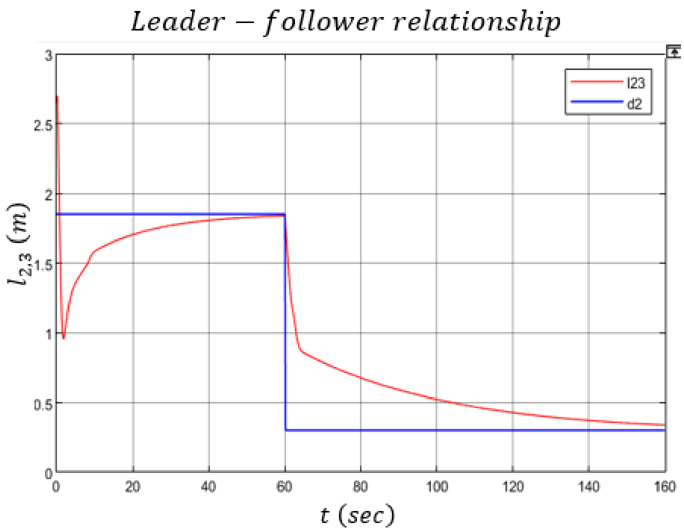


Figure 17c. Convergence of l_{23}

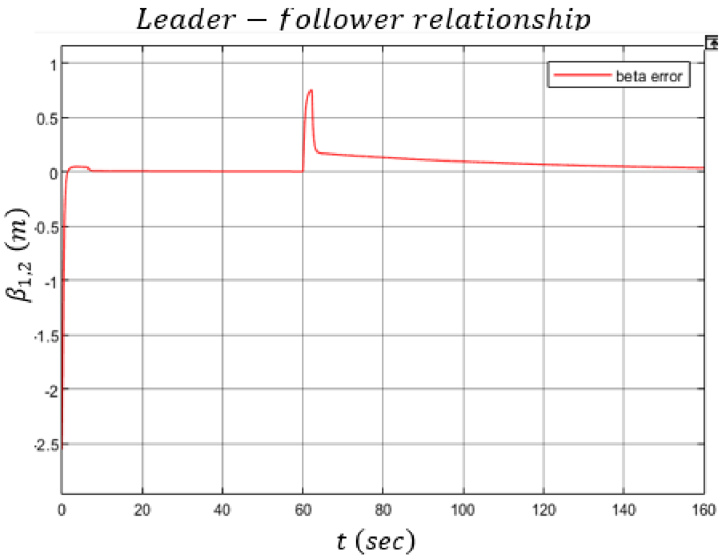


Figure 17d. Convergence of β_{12}

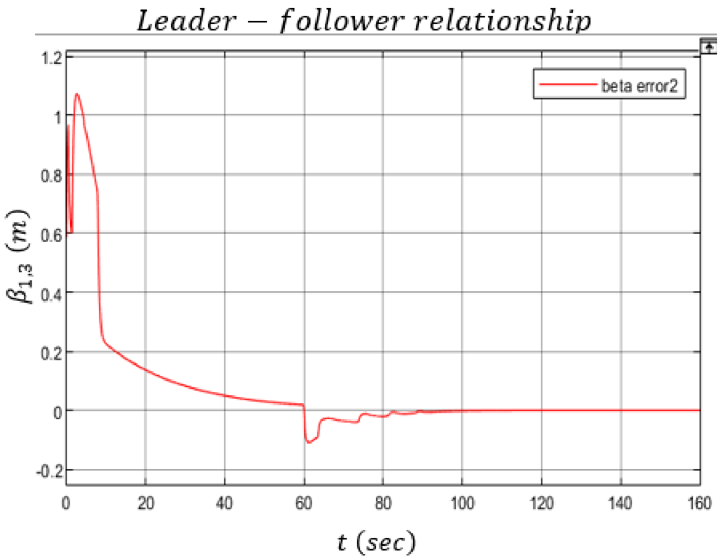


Figure 17e. Convergence of β_{13}

In the above figure, we can see that all of the parameters in our formation control of tracked mobile robots are converge to 0 in finite times.

CHAPTER 4. CONCLUSION

This research is mainly dealing with some problems that might happen in multi-rescue robots which are slip issues and formation control. This study proposed a tracked mobile robot that can track a trajectory with unknown slips. While tracking the trajectory, the tracked mobile robots can also converge the unknown longitude slip with adaptive update law and observe the orientation angle. Second, we can control a team of the robot with specific formation SSOS or SBOS. Receding Horizon control is implemented in the followers, using this control technology, the followers can follow the leader and keep the formation. These two algorithms are successfully implemented together in Simulink. We believe that this algorithm can solve some of the problems in rescue robots,

CHAPTER 5. APPENDIX

[1] SSOS

First, the reason why we do not use the paper's derivation is because we find that it is not correct. We use several numbers to verify the equation, but not of the result show the correct result. And we know that the formation control is to converge the $l_{i,j}^d$, $l_{i,k}^d$, and $\beta_{i,j}^d$. So we know that the idea is same of finding the intersection of two circles.

SSOS is used to find out the reference position of the robot which follow two leaders at the same time. In this section, we will provide our method to find the reference position instead of using the method provided by [8].

The follow should keep the desire distance with two leaders also direct the desired orientation angle. The orientation angle can be obtained by directly compute. θ_i^d is the desired follower orientation angle and θ_j is the orientation angle of the leader. $\beta_{i,j}^d$ is the desired difference between leader and follower angle.

$$\theta_i^d = \theta_j + \beta_{i,j}^d \tag{A.1.1}$$

The desire x position and y position can be obtained by following method. Assume we have two circle A and B. Note that A is the first robot followed by the follower and B is the second one, and C is the follower robot position. The distance between A and C and distance between B and C are denoted as $l_{i,j}^d$ and $l_{i,k}^d$ respectively.

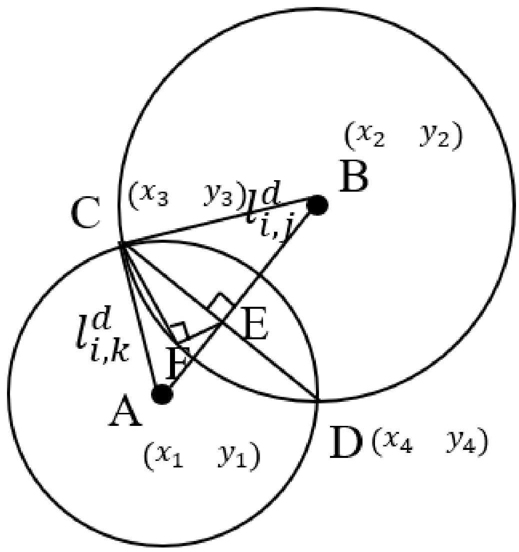


Figure A.1.1 Geometry relationship

We would use this geometry relationship to solve the position $(x_3 \quad y_3)$. First denote the distance between A and B is L, slope of $\overline{AB} = \mathrm{k}_1$ and slope of $\overline{CD} = \frac{1}{k_1}$. We know that the length of $\overline{CE} = \sqrt{{l_{i,j}^{d}}^{2} - \overline{AE}^2} = \sqrt{{l_{i,k}^{d}}^{2} - \overline{BE}^2}, \overline{BE}^2 = (L - \overline{AE})^2$.

$$\sqrt{{l_{i,j}^{d}}^{2} - \overline{AE}^2} = \sqrt{{l_{i,k}^{d}}^{2} - (L - \overline{AE})^2} \tag{A.1.2}$$

$$\overline{AE} = \frac{\left({l_{i,j}^{d}}^{2} - {l_{i,k}^{d}}^{2} + L^2\right)}{2L} \tag{A.1.3}$$

And we can obtain E's position$(x_E \quad y_E) =$
$\left(x_1 + \frac{\overline{AE}}{L}(x_2 - x_1) \quad y_1 + \frac{\overline{AE}}{L}(y_2 - y_1)\right)$. After get the position of E, we will use this information to derive the position of C or D. Note that $\overline{CE} = {l_{i,j}^{d}}^{2} - (x_E - x_1)^2 - (y_E - y_1)^2$ and $\mathrm{k}_2 = \frac{\cos\theta}{\sin\theta}$.

$$\overline{CE}^2 = \overline{EF}^2 + \overline{CF}^2 \tag{A.1.4}$$

$$\overline{EF}\cos\theta = \overline{CF}\sin\theta \tag{A.1.5}$$

$$\overline{CF} = \overline{EF}k_2 \tag{A.1.6}$$

$$\overline{EF} = \frac{\overline{CE}}{\sqrt{1 + k_2^2}} \tag{A.1.7}$$

Therefore, we can obtain the following equations.

$$(x_c \quad y_c) = \left(x_E - \frac{\overline{CE}}{\sqrt{1 + k_2^2}} \quad y_E + k_2(x_c - x_E)\right) \tag{A.1.8}$$

$$(x_d \quad y_d) = \left(x_E + \frac{\overline{CE}}{\sqrt{1 + k_2^2}} \quad y_E + k_2(x_d - x_E)\right) \tag{A.1.9}$$

$(x_c \quad y_c)$ or $(x_d \quad y_d)$ is the desire position of the formation control. Using this method, we can control our follower stay with the desire distance and orientation angle with two leaders.

[2] Controllability of a Trajectory

In this section we will derive the equation (8a), (8b) and (8c) and also prove that this system is controllable. We will start from the proof of controllability.

First, we define the variables which are used in the derivation. q_r, q and $\tilde{q}$ are denoted as reference position, current position, and difference of current position and reference position. $\tilde{v}$ and $\widetilde{\omega}$ are denoted as difference of velocity and difference of angular velocity respectively.

$$\tilde{q} = q - q_r \tag{A.2.1}$$

$$\tilde{v} = v - v_r \tag{A.2.2}$$

$$\widetilde{\omega} = \omega - \omega_r \tag{A.2.3}$$

We can write down a state space form of this system as follows equation.

$$\dot{\tilde{q}} = \begin{bmatrix} 0 & 0 & -v_r \sin\theta_r \\ 0 & 0 & v_r \cos\theta_r \\ 0 & 0 & 0 \end{bmatrix} \tilde{q} + \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} \tag{A.2.4}$$

Note that $\begin{bmatrix} 0 & 0 & -v_r \sin\theta_r \\ 0 & 0 & v_r \cos\theta_r \\ 0 & 0 & 0 \end{bmatrix} = A,$ and $\begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} = B$. And then we will rotate this matrix to the trajectory coordinate. The rotation matrix $\mathrm{R} = \begin{bmatrix} \cos\theta_r & \sin\theta_r & 0 \\ -\sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

Then we can get.

$$\tilde{q}_R = \begin{bmatrix} \cos\theta_r & \sin\theta_r & 0 \\ -\sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{q} \tag{A.2.5}$$

Differential equation (A.2.5).

$$\dot{\tilde{q}}_R = \begin{bmatrix} -\omega_r \sin\theta_r & \omega_r \cos\theta_r & 0 \\ -\omega_r \cos\theta_r & -\omega_r \sin\theta_r & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta_r & -\sin\theta_r & 0 \\ \sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{q}_R +$$

$$\begin{bmatrix} \cos\theta_r & -\sin\theta_r & 0 \\ \sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 & 0 & -v_r \sin\theta_r \\ 0 & 0 & v_r \cos\theta_r \\ 0 & 0 & 0 \end{bmatrix} \tilde{q} + \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} \right) \tag{A.2.6}$$

Combining with equation (A.2.5)

$$\dot{\tilde{q}}_R = \begin{bmatrix} -\omega_r \sin\theta_r & \omega_r \cos\theta_r & 0 \\ -\omega_r \cos\theta_r & -\omega_r \sin\theta_r & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta_r & -\sin\theta_r & 0 \\ \sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{q}_R +$$

$$\begin{bmatrix} \cos\theta_r & -\sin\theta_r & 0 \\ \sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} 0 & 0 & -v_r \sin\theta_r \\ 0 & 0 & v_r \cos\theta_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\theta_r & -\sin\theta_r & 0 \\ \sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \tilde{q}_R + \right.$$

$$\left. \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} \right) \tag{A.2.7}$$

After compute above equation we can get equation (A.2.8).

$$\dot{\tilde{q}}_R = \begin{bmatrix} 0 & \omega_d & 0 \\ -\omega_d & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} \tilde{q}_R + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{v} \\ \tilde{\omega} \end{bmatrix} \tag{A.2.8}$$

Derive the controllable matrix $\mathrm{C} = [B \quad AB \quad A^2B]$

$$\mathrm{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & -\omega_r^2 & v_r\omega_r \\ 0 & 0 & -\omega_r & v_r & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.2.9}$$

We can see that the rank(C) is 3 if $(v_r \epsilon R \cap v_r \neq 0) \cup (\omega_r \in R \cap \omega_r \neq 0)$, therefore, it is controllable if $v_r\ or\ \omega_r$ is nonzero.

The derivation of reference angular velocity will be provided as follows.

$$\omega_r = \frac{d}{dt}\theta_r \tag{A.2.10}$$

We know that $\theta_r = \tan\frac{\dot{y}_r}{\dot{x}_r}$, $\frac{d}{dt}\theta_r$ is derive as follows

$$\omega_r = \frac{d}{dt}\left(\frac{\dot{y}_r}{\dot{x}_r}\right)\left(\frac{1}{1+\left(\frac{\dot{y}_r}{\dot{x}_r}\right)^2}\right) \tag{A.2.11}$$

$$\omega_r = \frac{\dot{x}_r\ddot{y}_r-\ddot{x}_r\dot{y}_r}{\dot{x}_r^2}\left(\frac{1}{1+\left(\frac{\dot{y}_r}{\dot{x}_r}\right)^2}\right) \tag{A.2.12}$$

$$\omega_r = \frac{\dot{x}_r\ddot{y}_r-\ddot{x}_r\dot{y}_r}{\dot{x}_r^2+\dot{y}^2} \tag{A.2.13}$$

[3] Time differential of the tracking error

In this part, we will derive the equation (12) which is denotes at (A.3.1) in this section.

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \omega e_2 + v_r \cos e_3 - v \\ -\omega e_1 + v_r \sin e_3 \\ \omega_r - \omega \end{bmatrix} \tag{A.3.1}$$

$$\dot{e} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} e \tag{A.3.2}$$

$$e = [e_1 \quad e_2 \quad e_3]^T \tag{A.3.3}$$

Differential equation (A.3.2), then we can obtain

$$\dot{e}_1 = \frac{\mathrm{d}}{\mathrm{dt}}\left(\cos\theta\,(x_r - x) + \sin\theta\,(y_r - y)\right) \tag{A.3.4}$$

$$\dot{e}_1 = -\omega\sin\theta\,(x_r - x) + \omega\cos\theta\,(y_r - y) + \cos\theta\,(\dot{x}_r - \dot{x}) + \sin\theta\,(\dot{y}_r - \dot{y}) \tag{A.3.5}$$

Because $\dot{x} = v\cos\theta$, $\dot{y} = v\sin\theta$, $\dot{\theta} = \omega$, we can rewrite equation (A.3.5) to (A.3.6) shown as follows. Note that the nonholonomic constraint which means $\dot{x}\sin\theta - \dot{y}\cos\theta = 0$.

$$\dot{e}_1 = \dot{x}_r \cos\theta + \dot{y}_\mathrm{r} \sin\theta - \mathrm{v} + \mathrm{e}_2\omega \quad \text{(A.3.6)}$$
$$\dot{e}_1 = \dot{x}_r \cos(\theta_r - e_3) + \dot{y}_\mathrm{r} \sin(\theta_r - e_3) - \mathrm{v} + \mathrm{e}_2\omega \quad \text{(A.3.7)}$$
$$\dot{e}_1 = \dot{x}_r(\cos\theta_r \cos e_3 + \sin\theta_r \sin e_3) + \dot{y}_\mathrm{r}(\sin\theta_r \cos e_3 - \cos\theta_r \sin e_3) - \mathrm{v} + \mathrm{e}_2\omega \quad \text{(A.3.8)}$$
$$\dot{e}_1 = -\mathrm{v} + \mathrm{e}_2\omega + \cos e_3\,(\dot{x}_r \cos\theta_r + \dot{y}_r \sin\theta_r) + \sin e_3\,(\dot{x}_r \sin\theta_r - \dot{y}_r \cos\theta_r) \quad \text{(A.3.9)}$$
$$\dot{e}_1 = -\mathrm{v} + \mathrm{e}_2\omega + v_r \cos e_3 \quad \text{(A.3.10)}$$

Next, we will derive the $\dot{e}_2$

$$\dot{e}_2 = \omega\cos\theta\,(x_r - x) - \omega\sin\theta\,(y_r - y) + \sin\theta\,(\dot{x}_r - \dot{x}) - \cos\theta\,(\dot{y}_r - \dot{y}) \quad \text{(A.3.11)}$$
$$\dot{e}_2 = -e_1\omega + \dot{x}\sin\theta - \dot{y}\cos\theta - \dot{x}_r\sin\theta + \dot{y}_r\cos\theta \quad \text{(A.3.12)}$$
$$\dot{e}_2 = -e_1\omega - \dot{x}_r\sin(\theta_r - e_3) + \dot{y}_r\cos(\theta_r - e_3) \quad \text{(A.3.13)}$$
$$\dot{e}_2 = -e_1\omega + \dot{y}_r(\cos\theta_r \cos e_3 + \sin\theta_r \sin e_3) + x_\mathrm{r}(\sin\theta_r \cos e_3 - \cos\theta_r \sin e_3) \quad \text{(A.3.14)}$$
$$\dot{e}_2 = -e_1\omega + \sin e_3\,(\dot{x}_r \cos\theta_r + \dot{y}_r \sin\theta_r) + \cos e_3\,(-\dot{x}_r \sin\theta_r + \dot{y}_r \cos\theta_r) \quad \text{(A.3.9)}$$
$$\dot{e}_2 = -e_1\omega + v_r \sin e_3 \quad \text{(A.3.15)}$$

Derivation of $\dot{e}_3$ is pretty intuitive. We differential the e_3 and obtain the following equation.

$$\dot{e}_3 = \dot{\theta}_r - \dot{\theta} = \omega_r - \omega \quad \text{(A.3.16)}$$

[4] Introduction of Lyapunov function

First, we will discuss about the Lyapunov theorem. Lyapunov theorem means that the differential equation $\dot{x} = Ax, A \in C^{n\times n}$, is asymptotically stable if for any Q $\in \mathrm{C}^{n\times n}, Q > 0$, there exist P $\in \mathrm{C}^{\mathrm{n}\times\mathrm{n}}, \mathrm{P} > 0$ and it satisfy the following equation

$$\mathrm{A}^*P + PA = -Q \quad \text{(A.4.1)}$$

Note that P is unique. Briefly proof is provided as follows.
Assume that P > 0 and Q > 0and also satisfy equation (A.4.1). And assume that λ is the eigenvalue of A matrix and v is the corresponding eigenvector. v $\epsilon\ \mathrm{C}^\mathrm{n}$. Then we can obtain.

$$\mathrm{v}^*(A^*P + PA)v = -v^*Qv \quad \text{(A.4.2)}$$
$$\lambda^*\mathrm{v}^*Pv + \lambda\mathrm{v}^*Pv = -v^*Qv \quad \text{(A.4.3)}$$
$$(\lambda^* + \lambda)\mathrm{v}^*Pv = -v^*Qv \quad \text{(A.4.4)}$$

Note that P > 0 and Q > 0, so $\mathrm{v}^*Pv > 0$ and $v^*Qv > 0$ which means the real part of $(\lambda^* + \lambda) < 0$. Therefore, we can obtain that the system is asymptotically stable.

[5] Derivation of equation (22)

In this section, we will derive the equation (22) which denotes as (A.5.1) in this section from equation (21) which denotes as (A.5.2).

$$\dot{V}(\mathrm{t}) = -k_{1_i}k_{2_i}e_{1_i}^2 - k_{3_i}e_{3_i}^2 \leq 0 \tag{A.5.1}$$

$$\dot{V}(\mathrm{t}) = k_{2_i}e_{1_i}\dot{e}_{1_i} + k_{2_i}e_{2_i}\dot{e}_{2_i} + \dot{e}_{3_i}e_{3_i} \tag{A.5.5}$$

Import the equation (A.5.3), (A.5.4) and (A.5.5) into equation (A.5.2).

$$\dot{e_i} = \begin{bmatrix} 0 & \omega_{r_i} & 0 \\ -\omega_{r_i} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} e_i + \begin{bmatrix} 0 \\ \sin e_{3_i} \\ 0 \end{bmatrix} v_{r_i} + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{1_i} \\ u_{2_i} \end{bmatrix} \tag{A.5.3}$$

$$u_{1_i} = -k_{1_i}e_{1_i} \tag{A.5.4}$$

$$u_{2_i} = -\frac{k_{2_i}v_{r_i}(t)\sin e_{3_i}}{e_{3_i}} e_{2_i} - k_{3_i}e_{3_i} \tag{A.5.5}$$

Then we can obtain

$$k_{2_i}e_{1_i}\left(\omega_{r_i}e_{2_i} - k_{1_i}e_{1_i}\right) + k_{2_i}e_{2_i}\left(-\omega_{r_i}e_{1_i} + \mathrm{v}_{\mathrm{r_i}}\sin e_{3_i}\right) + e_{3_i}\left(-\frac{k_{2_i}v_{r_i}(t)\sin e_{3_i}}{e_{3_i}} e_{2_i} - k_{3_i}e_{3_i}\right) \tag{A.5.6}$$

After calculate (A.5.6) we can get $\dot{V}(\mathrm{t}) = -k_{1_i}k_{2_i}e_{1_i}^2 - k_{3_i}e_{3_i}^2$, k_{1_i}, k_{2_i} and k_{3_i} are all positive, therefore, $\dot{V}(\mathrm{t}) \leq 0$.

[6] Introduction of Barbalat Lemma

In this section, we will briefly introduce the Barbalat Lemma which is used in analysis of our control system.

Assume a function denote as $\mathrm{f(t)}$. if this function $\mathrm{f(t)}$ satisfy two condition, then when time approach infinite, $\dot{\mathrm{f}}(\mathrm{t}) \to 0$. The conditions are shown as follow.

(A.6.1) when $\mathrm{t} \to \infty, \mathrm{f(t)} has\ finite\ limits$. This can also be written as $\lim_{t\to\infty} f(t) = C$. C is a limited constant number.

(A.6.2) $\dot{f}(t)$ consistently continuous which means $\ddot{f}(t)$ is bounded, then if t approach infinite $\dot{f}(t)$ will converge to 0.

For condition (A.6.1), $\mathrm{t} \to \infty$ is not specified whether it is positive or negative.

For condition (A.6.2) the meaning of consistently continuous is that every point in the defined domain is continuous.

[7] Derivation of equation (32)

First, Note that some part of the derivation of the paper is not correct. We are very sure it is not correct because it Lyapunov function is not guaranteed that it will always smaller or equal to 0. And using the equation to do simulation is also not stable. Based on these observation we are very sure that it is not correct. Furthermore, we find out that the derivation of $\dot{e}_1$ is not correct. We provide the correct derivation in this part. Also, Note that we choose a different Lyapunov function.

Equations (30a), (30b), and (30c) are denoted as (A.7.1), (A.7.2), and (A.7.3).

$$\dot{e}_1 = \frac{a_L+\tilde{a}_L}{a_L}\left[\left(\frac{e_2 v}{b}+\frac{v}{2}\right)-\left(\frac{e_2\omega}{2}+\frac{b\omega}{4}\right)\right]+\frac{a_R+\tilde{a}_R}{a_R}\left[\left(\frac{e_2 v}{b}-\frac{v}{2}\right)+\left(\frac{e_2\omega}{2}-\frac{b\omega}{4}\right)\right]+\mathrm{v_r}\cos e_3 \tag{A.7.1}$$

$$\dot{e}_2 = \frac{a_L+\tilde{a}_L}{a_L}\left(\frac{e_1 v}{b}-\frac{e_1\omega}{2}\right)-\frac{a_R+\tilde{a}_R}{a_R}\left(\frac{e_1 v}{b}-\frac{e_1\omega}{2}\right)+\mathrm{v_r}\sin e_3 \tag{A.7.2}$$

$$\dot{e}_3 = \frac{a_L+\tilde{a}_L}{a_L}\left(\frac{v}{b}-\frac{\omega}{2}\right)-\frac{a_R+\tilde{a}_R}{a_R}\left(\frac{v}{b}+\frac{\omega}{2}\right)+\omega_r \tag{A.7.3}$$

The above equations are obtained by a transformation matrix T which transform original velocity and angular velocity to the velocity and angular velocity which are affected by slippage. The matrix are shown as follows. Note that v_o and ω_o are original velocity and original angular velocity.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = T\begin{bmatrix} v_o \\ \omega_o \end{bmatrix} \tag{A.7.4}$$

$$T = \begin{bmatrix} \frac{a_L+\tilde{a}_L}{2a_L}+\frac{a_R+\tilde{a}_R}{2a_R} & \frac{a_L+\tilde{a}_L}{a_L}\frac{b\omega}{4}+\frac{a_R+\tilde{a}_R}{a_R}\frac{b\omega}{4} \\ \frac{a_L+\tilde{a}_L}{a_L}\frac{v}{b}-\frac{a_R+\tilde{a}_R}{a_R}\frac{v}{b} & \frac{a_L+\tilde{a}_L}{a_L}\frac{\omega}{2}-\frac{a_R+\tilde{a}_R}{a_R}\frac{\omega}{2} \end{bmatrix}\begin{bmatrix} v_o \\ \omega_o \end{bmatrix} \tag{A.7.5}$$

And we have the Lyapunov candidate (A.7.6).

$$\mathrm{V(t)} = \frac{1}{2}e_1^2+\frac{1}{2}(e_2+k_3 e_3)^2+\frac{(1-\cos e_3)}{k_2}+\frac{\tilde{a}_L^2}{2\rho_1 a_L}+\frac{\tilde{a}_R^2}{2\rho_2 a_R} \tag{A.7.6}$$

Differential equation (A.7.6) and Combine equations (24), (30a), (30b) and (30c), then we obtain

$$\dot{V}(\mathrm{t}) = e_1\dot{e}_1+e_2\dot{e}_2+\frac{\dot{e}_3\sin e_3}{k_2}+\frac{\dot{\hat{a}}_L}{\rho_1}\frac{\tilde{a}_L}{a_L}+\frac{\dot{\hat{a}}_R}{\rho_2}\frac{\tilde{a}_R}{a_R} \tag{A.7.7}$$

$$\dot{V}(\mathrm{t}) = e_1\left\{\frac{a_L+\tilde{a}_L}{a_L}\left[\left(\frac{e_2 v}{b}+\frac{v}{2}\right)-\left(\frac{e_2\omega}{2}+\frac{b\omega}{4}\right)\right]+\frac{a_R+\tilde{a}_R}{a_R}\left[\left(\frac{e_2 v}{b}-\frac{v}{2}\right)+\left(\frac{e_2\omega}{2}-\frac{b\omega}{4}\right)\right]+\right.$$

$$\left.\mathrm{v_r}\cos e_3\right\}+e_2\left[\frac{a_L+\tilde{a}_L}{a_L}\left(\frac{e_1 v}{b}-\frac{e_1\omega}{2}\right)-\frac{a_R+\tilde{a}_R}{a_R}\left(\frac{e_1 v}{b}-\frac{e_1\omega}{2}\right)+\mathrm{v_r}\sin e_3\right]+$$

$$\frac{\left[\frac{a_L+\tilde{a}_L}{a_L}\left(\frac{v}{b}-\frac{\omega}{2}\right)-\frac{a_R+\tilde{a}_R}{a_R}\left(\frac{v}{b}+\frac{\omega}{2}\right)+\omega_r\right]\sin e_3}{k_2}+\frac{\dot{\hat{a}}_L}{\rho_1}\frac{\tilde{a}_L}{a_L}+\frac{\dot{\hat{a}}_R}{\rho_2}\frac{\tilde{a}_R}{a_R} \tag{A.7.8}$$

Using the control law equation (19) which is derived in previous section

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r\cos e_3 + k_1 e_1 \\ \omega_r + k_2 v_r e_2 + k_3\sin e_3 \end{bmatrix} \tag{A.7.9}$$

The derivative of Lyapunov function will become

$$\dot{V}(\mathrm{t}) = e_1\left\{\frac{a_L+\tilde{a}_L}{a_L}\left[\left(\frac{e_2}{b}+\frac{1}{2}\right)(v_r\cos e_3+k_1e_1)-\left(\frac{e_2}{2}+\frac{b}{4}\right)(\omega_r+k_2v_re_2+k_3\sin e_3)\right]+\right.$$

$$\left.\frac{a_R+\tilde{a}_R}{a_R}\left[\left(\frac{e_2}{b}-\frac{1}{2}\right)(v_r\cos e_3+k_1e_1)+\left(\frac{e_2}{2}-\frac{b}{4}\right)(\omega_r+k_2v_re_2+k_3\sin e_3)\right]+\mathrm{v_r}\cos e_3\right\}+$$

$$e_2\left[\frac{a_L+\tilde{a}_L}{a_L}\left(\frac{e_1(v_r\cos e_3+k_1e_1)}{b}-\frac{e_1(\omega_r+k_2v_re_2+k_3\sin e_3)}{2}\right)-\frac{a_R+\tilde{a}_R}{a_R}\left(\frac{e_1(v_r\cos e_3+k_1e_1)}{b}-\right.\right.$$

$$\left.\left.\frac{e_1(\omega_r+k_2v_re_2+k_3\sin e_3)}{2}\right)+\mathrm{v_r}\sin e_3\right]+\frac{\left[\frac{a_L+\tilde{a}_L}{a_L}\left(\frac{v}{b}-\frac{\omega}{2}\right)-\frac{a_R+\tilde{a}_R}{a_R}\left(\frac{v}{b}+\frac{\omega}{2}\right)+\omega_r\right]\sin e_3}{k_2}+\frac{\dot{\hat{a}}_L}{\rho_1}\frac{\tilde{a}_L}{a_L}+\frac{\dot{\hat{a}}_R}{\rho_2}\frac{\tilde{a}_R}{a_R} \tag{A.7.10}$$

$$\dot{V}(\mathrm{t}) = e_1\left\{\frac{a_L+\tilde{a}_L}{a_L}\left[\left(\frac{e_2}{b}+\frac{1}{2}\right)(v_r\cos e_3+k_1e_1)-\left(\frac{e_2}{2}+\frac{b}{4}\right)(\omega_r+k_2v_re_2+k_3\sin e_3)\right]+\right.$$

$$\left.\frac{a_R+\tilde{a}_R}{a_R}\left[\left(-\frac{1}{2}\right)(v_r\cos e_3+k_1e_1)+\left(-\frac{b}{4}\right)(\omega_r+k_2v_re_2+k_3\sin e_3)\right]+\mathrm{v_r}\cos e_3\right\}+$$

$$e_2\left[\frac{a_L+\tilde{a}_L}{a_L}\left(\frac{e_1(v_r\cos e_3+k_1e_1)}{b}-\frac{e_1(\omega_r+k_2v_re_2+k_3\sin e_3)}{2}\right)+\mathrm{v_r}\sin e_3\right]+$$

$$\frac{\left[\frac{a_L+\tilde{a}_L}{a_L}\left(\frac{v}{b}-\frac{\omega}{2}\right)-\frac{a_R+\tilde{a}_R}{a_R}\left(\frac{v}{b}+\frac{\omega}{2}\right)+\omega_r\right]\sin e_3}{k_2}+\frac{\dot{\hat{a}}_L}{\rho_1}\frac{\tilde{a}_L}{a_L}+\frac{\dot{\hat{a}}_R}{\rho_2}\frac{\tilde{a}_R}{a_R} \tag{A.7.11}$$

Combining all part which including $\tilde{a}_L$ and do the same thing at the parts including $\tilde{a}_R$ the equation will become

$$\dot{V}(\mathrm{t}) = \frac{\tilde{a}_L}{a_L}\left\{\frac{\dot{\hat{a}}_L}{\rho_1}-(v-b\omega)\left[\left(\frac{e_2}{2b}+\frac{1}{2}\right)e_1-\frac{(e_2+k_3e_3)e_1}{2b}-\frac{k_3(e_2+k_3e_3)}{2b}-\frac{\sin e_3}{2bk_2}\right]\right\}+\frac{\tilde{a}_R}{a_R}\left\{\frac{\dot{\hat{a}}_R}{\rho_2}-\right.$$

$$\left.(v+b\omega)\left[-\left(\frac{e_2}{2b}-\frac{1}{2}\right)e_1+\frac{(e_2+k_3e_3)e_1}{2b}+\frac{k_3(e_2+k_3e_3)}{2b}-\frac{\sin e_3}{2bk_2}\right]\right\}+e_1\left\{\left(\frac{e_2}{b}+\frac{1}{2}\right)(v_r\cos e_3+\right.$$

$k_1e_1) - \left(\frac{e_2}{2} + \frac{b}{4}\right)(\omega_r + k_2v_re_2 + k_3 \sin e_3) + \left(-\frac{1}{2}\right)(v_r \cos e_3 + k_1e_1) + \left(-\frac{b}{4}\right)(\omega_r +$

$k_2v_re_2 + k_3 \sin e_3)\Big\} + e_2\left[\left(\frac{e_1(v_r \cos e_3 + k_1e_1)}{b} - \frac{e_1(\omega_r + k_2v_re_2 + k_3 \sin e_3)}{2}\right) + \mathrm{v_r} \sin e_3\right]$

(A.7.12)

After calculate the equation becomes

$$\dot{V}(\mathrm{t}) = \frac{\tilde{a}_L}{a_L}\left\{\frac{\dot{\hat{a}}_L}{\rho_1} - (v - b\omega)\left[\left(\frac{e_2}{2b} + \frac{1}{2}\right)e_1 - \frac{(e_2+k_3e_3)e_1}{2b} - \frac{k_3(e_2+k_3e_3)}{2b} - \frac{\sin e_3}{2bk_2}\right]\right\} + \frac{\tilde{a}_R}{a_R}\left\{\frac{\dot{\hat{a}}_R}{\rho_2} - (v + b\omega)\left[-\left(\frac{e_2}{2b} - \frac{1}{2}\right)e_1 + \frac{(e_2+k_3e_3)e_1}{2b} + \frac{k_3(e_2+k_3e_3)}{2b} - \frac{\sin e_3}{2bk_2}\right]\right\} - k_1e_1^2 - \frac{k_2k_3v_r}{2}(e_2^2 + k_3^2e_3^2 + 2k_3e_2e_3) - \frac{v_r}{2k_2k_3}\sin^2 e_3 \quad \text{(A.7.13)}$$

$$\dot{V}(\mathrm{t}) = -k_1e_1^2 - \frac{k_2k_3v_r}{2}(e_2 + k_3e_3)^2 - \frac{v_r}{2k_2k_3}\sin^2 e_3 + \frac{\tilde{a}_L}{a_L}\left\{\frac{\dot{\hat{a}}_L}{\rho_1} - (v - b\omega)\left[\left(\frac{e_2}{2b} + \frac{1}{2}\right)e_1 - \frac{(e_2+k_3e_3)e_1}{2b} - \frac{k_3(e_2+k_3e_3)}{2b} - \frac{\sin e_3}{2bk_2}\right]\right\} + \frac{\tilde{a}_R}{a_R}\left\{\frac{\dot{\hat{a}}_R}{\rho_2} - (v + b\omega)\left[-\left(\frac{e_2}{2b} - \frac{1}{2}\right)e_1 + \frac{(e_2+k_3e_3)e_1}{2b} + \frac{k_3(e_2+k_3e_3)}{2b} - \frac{\sin e_3}{2bk_2}\right]\right\} \quad \text{(A.7.14)}$$

We can design the adaptive update law from (32), the variables we can design are $\dot{\hat{a}}_L$ and $\dot{\hat{a}}_R$. We observe that the first three part is negative, and it does not have any possibility to higher than 0. Therefore, we only have to delete the part which have possibility that higher than 0.

$$\frac{\dot{\hat{a}}_L}{\rho_1} - (v - b\omega)\left[\left(\frac{e_2}{2b} + \frac{1}{2}\right)e_1 - \frac{(e_2+k_3e_3)e_1}{2b} - \frac{k_3(e_2+k_3e_3)}{2b} - \frac{\sin e_3}{2bk_2}\right] = 0 \quad \text{(A.7.15)}$$

$$\frac{\dot{\hat{a}}_L}{\rho_1} = (v - b\omega)\left[\left(\frac{e_2}{2b} + \frac{1}{2}\right)e_1 - \frac{(e_2+k_3e_3)e_1}{2b} - \frac{k_3(e_2+k_3e_3)}{2b} - \frac{\sin e_3}{2bk_2}\right] \quad \text{(A.7.16)}$$

$$\dot{\hat{a}}_L = \rho_1(v - b\omega)\left[\left(\frac{e_2}{2b} + \frac{1}{2}\right)e_1 - \frac{(e_2+k_3e_3)e_1}{2b} - \frac{k_3(e_2+k_3e_3)}{2b} - \frac{\sin e_3}{2bk_2}\right] \quad \text{(A.7.17)}$$

$$\frac{\dot{\hat{a}}_R}{\rho_2} - (v + b\omega)\left[-\left(\frac{e_2}{2b} - \frac{1}{2}\right)e_1 + \frac{(e_2+k_3e_3)e_1}{2b} + \frac{k_3(e_2+k_3e_3)}{2b} - \frac{\sin e_3}{2bk_2}\right] = 0 \quad \text{(A.7.18)}$$

$$\frac{\dot{\hat{a}}_R}{\rho_2} = (v + b\omega)\left[-\left(\frac{e_2}{2b} - \frac{1}{2}\right)e_1 + \frac{(e_2+k_3e_3)e_1}{2b} + \frac{k_3(e_2+k_3e_3)}{2b} - \frac{\sin e_3}{2bk_2}\right] \quad \text{(A.7.19)}$$

$$\dot{\hat{a}}_R = \rho_2(v + b\omega)\left[-\left(\frac{e_2}{2b} - \frac{1}{2}\right)e_1 + \frac{(e_2+k_3e_3)e_1}{2b} + \frac{k_3(e_2+k_3e_3)}{2b} - \frac{\sin e_3}{2bk_2}\right] \quad \text{(A.7.20)}$$

Combine equation (A.7.5), (A.7.8), and (A.7.11), the Lyapunov equation will become

$$\dot{V}(\mathrm{t}) = -k_1 e_1^2 - \frac{v_r}{2} k_2 k_3 (e_2 + k_3 e_3)^2 - \frac{v_r}{2k_2 k_3} \sin^2 e_3 \leq 0 \tag{A.7.21}$$

[8] Derivation of equation (41) and equation (42)

$$\dot{z}_1 = z_3 v \tag{A.8.1}$$

$$\dot{z}_2 = z_4 v \tag{A.8.2}$$

$$\dot{z}_3 = -\omega \sin\theta = -\omega z_4 \tag{A.8.3}$$

$$\dot{z}_4 = \omega \cos\theta = \omega z_3 \tag{A.8.4}$$

Defining observer error parameters $\tilde{z}_k$ which can be written as $\tilde{z}_k = z_k - \hat{z}_k$, k = 3 or 4. Introduce these new parameters to the differential equation we can get new dynamic equation as follow

$$\begin{cases} \dot{\tilde{z}}_3 = \dot{z}_3 - \dot{\hat{z}}_3 = -\tilde{z}_4 \omega - L\tilde{z}_3 v \\ \dot{\tilde{z}}_4 = \dot{z}_4 - \dot{\hat{z}}_4 = \tilde{z}_3 \omega - L\tilde{z}_4 v \end{cases} \tag{A.8.5}$$

In order to analysis the observer we choose a Lyapunov function which is shown as follow

$$\mathrm{V_O} = \frac{1}{2}\tilde{z}_3^2 + \frac{1}{2}\tilde{z}_4^2 \tag{A.8.6}$$

$$\dot{V}_O = \dot{\tilde{z}}_3 \tilde{z}_3 + \dot{\tilde{z}}_4 \tilde{z}_4 = (-\tilde{z}_4 \omega - L\tilde{z}_3 v)\tilde{z}_3 + \tilde{z}_4(\tilde{z}_3 \omega - L\tilde{z}_4 v) \tag{A.8.7}$$

After defining our Lyapunov function we take the differential of equation (A.8.7) and the result is shown as follow

$$\dot{V}_O = -\mathrm{L}(\tilde{z}_3^2 + \tilde{z}_4^2)v \leq -2v_{min} L V_O \tag{A.8.8}$$

$$V_O \leq V_O(0) e^{-2v_{min}Lt} \tag{A.8.9}$$

We can solve equation (A.8.8) by using integral factor. After compute the calculation we can obtain equation (A.8.9). Using equation (A.8.7) and (A.8.9) we can obtain the following equation. Note that the will be provided in

$$\left\|(\tilde{z}_3, \check{z}_4)\right\|_2 \leq \left\|\left(\tilde{z}_3(0), \check{z}_4(0)\right)\right\|_2 e^{-2v_{min}Lt} \tag{A.8.10}$$

[9] Separation Principle

In this section, we will show the separation principle which means that your controller and observer can be designed separately. First, we define a system as follows.

$$\dot{x} = Ax + Bu \tag{A.9.1}$$

$$\dot{\hat{x}} = A\hat{x} + L(Cx - C\hat{x}) + Bu \tag{A.9.2}$$

$$\mathrm{u} = \mathrm{r} - \mathrm{K}\hat{x} \tag{A.9.3}$$

Then, we can get the following equation (A.9.4), by combining above equations.

$$\begin{bmatrix}\dot{x}\\ \dot{\hat{x}}\end{bmatrix} = \begin{bmatrix}A & -BK\\ LC & A-LC-BK\end{bmatrix}\begin{bmatrix}x\\ \hat{x}\end{bmatrix} + \begin{bmatrix}B\\ B\end{bmatrix}r \tag{A.9.4}$$

Let $\begin{bmatrix}A & -BK\\ LC & A-LC-BK\end{bmatrix} = A_{clp}, \begin{bmatrix}B\\ B\end{bmatrix} = B_{clp}$and define a new state vector $\begin{bmatrix}x\\ e\end{bmatrix}$. e is $x - \hat{x}$. Then, we can find a transformation matrix T to transform the old state to the new state.

$$\mathrm{T} = \begin{bmatrix}I & 0\\ I & -I\end{bmatrix} \tag{A.9.5}$$

And we can rewrite the state equation in new coordinate.

$$\begin{bmatrix}\dot{x}\\ \dot{e}\end{bmatrix} = T^{-1}A_{clp}T\begin{bmatrix}x\\ e\end{bmatrix} + T^{-1}B_{clp}r = \begin{bmatrix}A-BK & BK\\ 0 & A-LC\end{bmatrix}\begin{bmatrix}x\\ e\end{bmatrix} + \begin{bmatrix}B\\ 0\end{bmatrix}r \tag{A.9.6}$$

From equation (A.9.6) we know that the eigenvalue of the state feedback and state observer can be designed separately.

[10] RH formulation for regulation

Assume a dynamic system as follow

$$\mathrm{x}(k+1) = Ax(k) + Bu(k) \tag{A.10.1}$$

$$\mathrm{y(k)} = Cx(k) \tag{A.10.2}$$

Also define a cost function as follows

$$\mathrm{J} = \sum_{j=0}^{N}\{||y(k+j|k)|| + ||u(k+j|k)||\} + F\big(x(k+N|k)\big) \tag{A.10.3}$$

Note that $F\big(x(k+N|k)\big)$ is the terminal cost function. If N approach infinite, there would be no constraint on control input u and observe output y. and it will become a LQR problem. For a finite N, and assume F = 0.

$$\min_{u} J = \sum_{j=0}^{N}\{||y(k+j|k)|| + ||u(k+j|k)||\} + 0 \tag{A.10.4}$$

$$\text{s.t.}\quad \mathrm{x}(k+j+1|k) = Ax(k+j|k) + Bu(k+j|k) \tag{A.10.5}$$

$$\mathrm{x}(k|k) \equiv x(k) \tag{A.10.6}$$

$$\mathrm{y}(k+j|k) = Cx(k+j|k) \tag{A.10.7}$$

Assume that $|\mathrm{u}(k+j|k)| \le u_m$. To solve this problem, we have to convert above equations to a standard optimization form.

$$\mathrm{y}(k|k) = Cx(k|k) \tag{A.10.8}$$

$$\mathrm{y}(k+1|k) = Cx(k+1|k) \tag{A.10.9}$$

Using equation (A.11.5) we can obtain

$$\mathrm{y}(k+1|k) = C\big(Ax(k|k) + Bu(k|k)\big) = \mathrm{CAx}(k|k) + CBu(k|k) \tag{A.10.10}$$

And then we can derive to

$$\mathrm{y}(k+2|k) = Cx(k+2|k) \tag{A.10.11}$$

$$\mathrm{y}(k+2|k) = C\big(Ax(k+1|k) + Bu(k+1|k)\big) \tag{A.10.12}$$

$$\mathrm{y}(k+2|k) = CA\big(Ax(k|k) + Bu(k|k)\big) + CBu(k+1|k) \tag{A.10.13}$$

$$\mathrm{y}(k+2|k) = CA^2x(k|k) + CABu(k|k) + CBu(k+1|k) \tag{A.10.14}$$

From above equations we can derive

$$\mathrm{y}(k+N|k) = CA^N x(k|k) + CA^{N-1}Bu(k|k) + \cdots + CBu(k+1|k) \tag{A.10.15}$$

Rewrite above equations to matrix form.

$$\begin{bmatrix} \mathrm{y}(k|k) \\ \mathrm{y}(k+1|k) \\ \mathrm{y}(k+2|k) \\ \vdots \\ \mathrm{y}(k+N|k) \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix} x(k|k) +$$

$$\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \cdots & CB \end{bmatrix} \begin{bmatrix} u(k|k) \\ u(k+1|k) \\ u(k+2|k) \\ \vdots \\ u(k+N-1|k) \end{bmatrix} \tag{A.10.16}$$

Define four variables $\mathrm{G} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix}$, $\mathrm{H} =$

$\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & CA^{N-3}B & \cdots & CB \end{bmatrix}$, $\mathrm{Y}(k) \equiv \begin{bmatrix} \mathrm{y}(k|k) \\ \mathrm{y}(k+1|k) \\ \mathrm{y}(k+2|k) \\ \vdots \\ \mathrm{y}(k+N|k) \end{bmatrix}$, $\mathrm{U(k)} \equiv$

$\begin{bmatrix} u(k|k) \\ u(k+1|k) \\ u(k+2|k) \\ \vdots \\ u(k+N-1|k) \end{bmatrix}$ combine with $\mathrm{x}(k|k) = x(k)$, we can obtain

$$\mathrm{Y}(k) = Gx(k) + HU(k) \tag{A.10.17}$$

Thus, the optimal problem can be derived as follows

$$\mathrm{Y}^{\mathrm{T}}(k)\mathrm{Y}(k) + U^T(k)U(k) = \big(Gx(k) + HU(k)\big)^T\big(Gx(k) + HU(k)\big) + U^T(k)U(k) =$$

$$x(k)^T G^T G x(k) + \tfrac{1}{2}U(k)^T 2(H^T H + I)U(k) + 2(x(k)^T G^T H)U(k) \tag{A.10.18}$$

Assume $\mathrm{Q} = 2(H^T H + I), \mathrm{q}^{\mathrm{T}} = 2(x(k)^T G^T H)$, the RH control can be written as

$$\min_{U(k)} \tilde{J} = \mathrm{q}^{\mathrm{T}} U(k) + \frac{1}{2} U(k)^T Q U(k) \tag{A.10.19}$$

From above equation the RH control problem fall into the standard formulation of quadratic programming problem.

CHAPTER 6. CODE IN SIMULINK

We will show the code and the Simulink file in this part. However, because we use Simulink to do this project, we can only provide the function we write in Simulink. If you only run the function, it will not able to run the simulation.

If you want to run the simulation, we provide the Simulink file in following links.

https://drive.google.com/file/d/1WfesN4BqGXoPAz2Pm_WCoES8ncHySbSg/view?usp=sharing
https://drive.google.com/file/d/17_2AKZuJMZsDZg4H2pGBcI977WFXtl3z/view?usp=sharing
https://drive.google.com/file/d/1Onct5L6e8cgr0o6mRQaBoNtqyjkdkRa2/view?usp=sharing

The first Simulink file contain the RH-formation control and observer-based adaptive control. This simulation only tracks a simple trajectory. The second file mainly implement the adaptive trajectory tracking the circular trajectory and a more complex linear trajectory. The code will be provided as follow. Note that the repeated code will not be provided twice. The third is similar with the first file, but it transform from triangle formation to line formation.

```
function [td,vd,wd] = cmd_generation(xd_d,xd_dd,yd_dd,yd_d)
%%% This function is used to compute desire velocity and
desire angular velocity
%%%

%    if atan2(yd_d,xd_d)<0
%        td=atan2(yd_d,xd_d)+2*pi;
%    else
    td=atan2(yd_d,xd_d);
%    end

    vd = sqrt(xd_d^2+yd_d^2);
    if(xd_d^2+yd_d^2==0)
        wd=0;
    else
        wd=(yd_dd*xd_d-xd_dd*yd_d)/(xd_d^2+yd_d^2);
```

```
    end
end


function [k1,k2,k3] = K(vd,wd)
%%% This function is used to compute the feedback gain
which is used in the
%%% leader of the formation control
    epsilon=1;
    kb=14;

    k1=2*epsilon*sqrt(wd^2+kb*vd^2);
    k3=k1;
    k2=kb*abs(vd);

end


function [e1,e2,e3] = error(x_err,y_err,theta_error,theta)
%%% This function is used to compute the state error.
    e=[cos(theta) sin(theta) 0;-sin(theta) cos(theta) 0;0 0
1]*[x_err;y_err;theta_error];
    e1=e(1);
    e2=e(2);
    e3=e(3);
end


function [v,w] = fcn(wd,vd,k1,k2,k3,e1,e2,e3)
%%% This function is used to compute the velocity command
and angular
%%% velocity command
    v=zeros(1);
    w=zeros(1);
    vmax=0.15;
    wmax=0.45;
```

```
%    vc=vd*cos(e3)+k1*e1;
%    wc=wd+k2*sign(vd)*e2+k3*e3;

    %% gain

    k11 = 2;
    k22 = 10;
    k33 = 2;

    %%
    wc = wd + vd / 2 * (k22 * (e2 + k33 * e3) + sin(e3) /
k33);
    vc = vd * cos(e3) + k33 * e3 * wc + k11 * e1;
%     wc=wd+k2*(vd)*e2+k3*sin(e3);

    sigma=max([abs(vc/vmax),abs(wc/wmax),1]);
    if(sigma==1)
        v=vc;
        w=wc;
    elseif(sigma==vc/vmax)
        v=sign(vc)*vmax;
        w=wc/sigma;
    else
        v=vc/sigma;
        w=sign(wc)*wmax;

    end
end

function [wl,wr] = IK(v,w,Pl_hat,Pr_hat)
%%% This function is used to compute the inverse kinematic
with estimate slip,
    %parameters
    r=0.075;
    b=0.3;
    %inverse kinematics
```

```
%     wl = v/(r*il) - w*b/(r*il);
%     wr = v/(r*ir) + w*b/(r*ir);
    wl = v*Pl_hat/(r) - w*b*Pl_hat/(2*r);
    wr = v*Pr_hat/(r) + w*b*Pr_hat/(2*r);
end

function [z3_hat,z4_hat,L,theta] = K(z1,z2,alpha,beta)
%%% Observer is designed in this function
    L = 100;
    z3_hat = alpha + L * z1;
    z4_hat = beta + L * z2;
    theta = atan2(z4_hat,z3_hat);
end

function [Pl_dot,Pr_dot]  = observer(e1,e2,e3,v,w,k2,k3,
t,pl, pr)
%%% Adaptive update law
    %% Paramaters
    gamma1 = 0.05;
    gamma2 = 0.05;
    r=0.075;
    b=0.3;
    k1 = 2;
    k2 = 10;
    k3 = 2;

       Pl_dot = gamma1 * (v - b*w/2) * (e1 * (e2/b + 1/2) -
e1 * (e2 + k3 * e3)/b - k3 * (e2 + k3 * e3) / b -
sin(e3)/(b*k2));
       Pr_dot = gamma2 * (v + b*w/2) * (-e1 * (e2/b - 1/2)
+ e1 * (e2 + k3 * e3)/b + k3 * (e2 + k3 * e3) / b +
sin(e3)/(b*k2));




end
```

```
function [fx1,fy1, ft1, fx2, fy2, fz2] =
fcn(Lx,Ly,Lt,time)
    % Triangle Formation
     if(time < 60)

        d = 1.5;
        theta1 = 3.14159/1.5;
        theta2 = 3.14159*3/2;
%          theta3 = ;
        %% First Robot SBOS
        fx1 = Lx + d * cos(Lt + theta1);
        fy1 = Ly + d * sin(Lt + theta1);
        ft1 = Lt;
        %% Second Robot SSOS
        d1 = 1.5;
        d2 = 1.5;
        dx = Lx - fx1;
        dy = Ly - fy1;
        M = (1/2) * (Lx * Lx - fx1 * fx1 + Ly * Ly -
fy1 * fy1 - d1*d1 + d2*d2);
        a = 2 * (dx * dx + dy * dy);
        b = 2 * (M * dx - dy* (dx * Ly - dy * Lx));
        c = (M - dy * Ly) * (M - dy * Ly) - (dy * Lx) *
(dy * Lx) - (d1 * dy) * (d1 * dy);
        fx2 = (b + (b * b - 4 * a * c)) / a;
%          fx2 = (b - (b * b - 4 * a * c)) / a;
        fy2 = (M - dx * fx2) / dy;
        fz2 = Lt;


%          fx2 = Lx + d * cos(Lt + theta2);
%          fy2 = Ly + d * sin(Lt + theta2);
%          fz2 = Lt;
%          %% Third Robot
%          fx3 = Lx + d * cos(Lt + theta2);
%          fy3 = Ly + d * sin(Lt + theta2);
%          fz3 = Lt;
```

```
    %% My method of tracking two robot
        L = sqrt((Lx - fx1) * (Lx - fx1) + (Ly - fy1) *
(Ly - fy1))
        k1 = (Ly - fy1) / (Lx - fx1);
        k2 = -1/k1;
        AE = (d1*d1 - d2*d2 + L*L)/(2*L);
        CE = d1 * d1 - AE * AE;
        x0 = Lx + AE / L * (fx1 - Lx);
        y0 = Ly + AE / L * (fy1 - Ly);
        fx2 = x0 - CE/sqrt(1+k2*k2);
        fy2 = y0 + k2 * (fx2 - x0);
        fz2 = Lt;
    %Linear Formation
     else
%
         d = 0.3;
         theta = 3.14159;

         fx1 = Lx + d * cos(Lt + theta);
         fy1 = Ly + d * sin(Lt + theta);
         ft1 = Lt;
         fx2 = Lx + d * 2 * cos(Lt + theta);
         fy2 = Ly + d * 2 * sin(Lt + theta);
         fz2 = ft1;


     end


end

function [k1,k2,k3] = fcn(ex,ey,et,
vd,wd,time,x,y,theta)
%%% RH control of SBOS follower

    ex_total = 0;
```

```
ey_total = 0;
et_total = 0;
%% Select weiting matrix Q and matrix R

q1 = 0.02;
q2 = 0.02;
q3 = 0.015;
R1 = 0;
R2 = 0;
R3 = 0;

%% Formation control paramters
d = 1.5;
theta1 = 3.14159/1.5;
theta2 = 3.14159*3/2;

%% Receding Horizon control
Predict_horizon = 0.1;
Sampling_time = 0.01;

k3 = 0.35 + q3;
k2 = 0.5 + vd * sin(et) / et;
k1 = 0.5 + q1 + q2 * ey * ey / (ex * ex);

for t = 0:Predict_horizon:Sampling_time

    % Reference trajectory
    if (time < 60)
        fx1 = 0.05*(t+time) + d * cos(theta1);
        fy1 = 0.05*(t+time) + d * sin(theta1);
        fz1 = 3.14159/4;
    else % times > 20 (sec)
        d = 0.3
        fx1 = 0.05*(t+time) + d * cos(3.14159*1.25);
        fy1 = 0.05*(t+time) + d * sin(3.14159*1.25);
        fz1 = 3.14159/4;
    end
```

```
        % Optimal strategy LQR control

        k3 = 0.35 + q3;
        if(et == 0)
            k2 = 0.5;
        else
            k2 = 0.5 + vd * sin(et) / et;
        end
        k1 = 0.5 + q1 + q2 * ey * ey / (ex * ex);

        % caculate control input

        if(et == 0)
            v = vd * cos(et) + k1 * ex;
            w = wd + k3 * et + k2 * vd * ey;
        else
            v = vd * cos(et) + k1 * ex;
            w = wd + k3 * et +  k2 * vd * ey *
sin(et)/et;
        end

        % Forward kinematic

        x = x + v * cos(theta) * Sampling_time;
        y = y + v * sin(theta) * Sampling_time;
        theta = theta + w * Sampling_time;

        % caculate next error
        ex_total = ex_total + ex * Sampling_time;
        ey_total = ey_total + ey * Sampling_time;
        et_total = et_total + et * Sampling_time;


        e=[cos(theta) sin(theta) 0;-sin(theta)
cos(theta) 0;0 0 1]*[x-fx1;y-fy1;theta-fz1];
        ex=e(1);
        ey=e(2);
```

```
        et=e(3);
    end

    %% compute control gain
    k3 = 1 + q3;
    if(et_total == 0)
        k2 = 15;
    else
        k2 = 15 + 2 * vd * sin(et_total) / et_total;
    end



%     k1 = max(k1,5);
%     k2 = max(k2,100);
%     k3 = max(k3,5);

%     k1 = min(max(k1,5),10);
    k2 = min(max(k2,2),40);
    k3 = min(max(k3,1),2);


    if(ey_total * ey_total / (ex_total * ex_total) <=
30)
        k1 = 1 + q1 + q2 * ey_total * ey_total /
(ex_total * ex_total);
    else
        k1 = 2;
        k2 = 10;
        k3 = 3;
    end



end
```

```
function [k1,k2,k3] = fcn(ex,ey,et,
vd,wd,time,x,y,theta)
% SSOS
    ex_total = 0;
    ey_total = 0;
    et_total = 0;
    %% Select weiting matrix Q and matrix R

    q1 = 0.01;
    q2 = 0.1;
    q3 = 0.5;
    R1 = 0;
    R2 = 0;
    R3 = 0;
    %% Formation control paramters
    d = 1.5;
    theta1 = 3.14159/1.5;
    theta2 = 3.14159*3/2;
    theta3 = 3.141593*1.25;

    %% Receding Horizon control
    Predict_horizon = 0.1;
    Sampling_time = 0.01;

    k3 = 0.35 + q3;
    k2 = 0.5 + vd * sin(et) / et;
    k1 = 0.5 + q1 + q2 * ey * ey / (ex * ex);

    for t = 0:Predict_horizon:Sampling_time
        if(time < 60)
            % Reference trajectory
%            % First Robot SBOS
%            fx1 = Lx + d * cos(Lt + theta1);
%            fy1 = Ly + d * sin(Lt + theta1);
%            ft1 = Lt;
            fx1 = 0.05*(t+time) + d * cos(theta1);
            fy1 = 0.05*(t+time) + d * sin(theta1);
```

```
            fz1 = 3.14159/4;
            % Second Robot SSOS
            d1 = 1.5;
            d2 = 1.5;
            dx = 0.05*(t+time) - fx1;
            dy = 0.05*(t+time) - fy1;
            M = (1/2) * (0.05*(t+time) * 0.05*(t+time) -
fx1 * fx1 + 0.05*(t+time) * 0.05*(t+time) - fy1 * fy1
- d1*d1 + d2*d2);
            a = 2 * (dx * dx + dy * dy);
            b = 2 * (M * dx - dy* (dx * 0.05*(t+time) -
dy * 0.05*(t+time)));
            c = (M - dy * 0.05*(t+time)) * (M - dy *
0.05*(t+time)) - (dy * 0.05*(t+time)) * (dy *
0.05*(t+time)) - (d1 * dy) * (d1 * dy);
            fx2 = (b + (b * b - 4 * a * c)) / a;
    %         fx2 = (b - (b * b - 4 * a * c)) / a;
            fy2 = (M - dx * fx2) / dy;
            fz2 = 3.14159/4;
    %         fx2 = 0.05*(t+time) + d * cos(theta2);
    %         fy2 = 0.05*(t+time) + d * sin(theta2);
    %         fz2 = 3.14159/4;
    %         fx1 = Lx + d * cos(Lt + theta1);
    %         fy1 = Ly + d * sin(Lt + theta1);
    %         ft1 = Lt;
    %
    %         dx = Lx - fx1;
    %         dy = Ly - fy1;
    %         M = (1/2) * ((Lx * Lx - fx1 * fx1) + (Ly *
Ly - fy1 * fy1) - d*d + d*d);
    %         a = 2 * (dx * dx + dy * dy);
    %         b = 2 * (M * dx - dy* (dx * Ly - dy *
Lx));
    %         c = (M - dy * Ly) * (M - dy * Ly) - (dy *
Lx) * (dy * Lx) - (d * dy) * (d * dy);
    %         fx2 = (b + sqrt((b * b - 4 * a * c))) / a;
    % %         fx2 = (b - (b * b - 4 * a * c)) / a;
```

```
    %         fy2 = (M - dx * fx2) / dy;
    %         fz2 = Lt;
        % My method of tracking two robot
            L = sqrt((0.05*(t+time) - fx1) *
(0.05*(t+time) - fx1) + (0.05*(t+time) - fy1) *
(0.05*(t+time) - fy1))
            k1 = (0.05*(t+time) - fy1) / (0.05*(t+time)
- fx1);
            k2 = -1/k1;
            AE = (d1*d1 - d2*d2 + L*L)/(2*L);
            CE = d1 * d1 - AE * AE;
            x0 = 0.05*(t+time) + AE / L * (fx1 -
0.05*(t+time));
            y0 = 0.05*(t+time) + AE / L * (fy1 -
0.05*(t+time));
            fx2 = x0 - CE/sqrt(1+k2*k2);
            fy2 = y0 + k2 * (fx2 - x0);
            fz2 = 3.14159/4;

        else % times > 20 (sec)
            d = 0.3
            fx2 = 0.05*(t+time) + 2 * d *
cos(3.14159*1.25);
            fy2 = 0.05*(t+time) + 2 * d *
sin(3.14159*1.25);
            fz2 = 3.14159/4;
        end
            % Optimal strategy LQR control

        k3 = 0.35 + q3;
        if(et == 0)
            k2 = 0.5;
        else
            k2 = 0.5 + vd * sin(et) / et;
        end
        k1 = 0.5 + q1 + q2 * ey * ey / (ex * ex);
```

```
            % cauculate control input

        if(et == 0)
            v = vd * cos(et) + k1 * ex;
            w = wd + k3 * et + k2 * vd * ey;
        else
            v = vd * cos(et) + k1 * ex;
            w = wd + k3 * et +  k2 * vd * ey *
sin(et)/et;
        end

        % Forward kinematic

        x = x + v * cos(theta) * Sampling_time;
        y = y + v * sin(theta) * Sampling_time;
        theta = theta + w * Sampling_time;

        % calculate next error
        e=[cos(theta) sin(theta) 0;-sin(theta)
cos(theta) 0;0 0 1]*[x-fx2;y-fy2;theta-fz2];
        ex=e(1);
        ey=e(2);
        et=e(3);

        ex_total = ex_total + ex * Sampling_time;
        ey_total = ey_total + ey * Sampling_time;
        et_total = et_total + et * Sampling_time;
    end

    %% compute control gain

    k3 = 1 + q3;
    if(et_total == 0)
        k2 = 15;
    else
        k2 = 15 + 2 * vd * sin(et_total) / et_total;
    end
```

```
%     if(ey_total * ey_total / (ex_total * ex_total) <=
0.0000001)
%         k1 = 1 + q1 + q2 * ey_total * ey_total /
(ex_total * ex_total);
%     else
%         k1 = 2;
%     end

%     k1 = max(k1,5);
%     k2 = max(k2,100);
%     k3 = max(k3,5);

%     k1 = min(max(k1,5),10);
    k2 = min(max(k2,40),60);
    k3 = min(max(k3,1),2);

    if(ey_total * ey_total / (ex_total * ex_total) <=
2)
        k1 = 1 + q1 + q2 * ey_total * ey_total /
(ex_total * ex_total);
    else
        k1 = 2;
        k2 = 30;
        k3 = 3;
    end




end
```

Chapter 7. BIBLIOGRAPHY

[1] M. Asif, M. J. Khan, M. Safwan, and M. Rehan, “Feedforward and feedback kinematics controller for wheeled mobile robot trajectory tracking,” J. Autom. Control, vol. 3, no. 3, pp. 178–182, 2015. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[2] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, “A stable tracking control method for a nonholonomic mobile robot,” in Proc. IEEE/RSJ Int. Workshop Intelligent Robots and Systems, 1991, pp. 1236–124.

[3] A. De Luca, G. Oriolo, and M. Vendittelli, “Control of wheeled mobile robots: An experimental overview,” in RAMSETE—Articulated and Mobile Robotics for Services and Technologies (Lecture Notes in Control and Information Sciences), vol. 270. Berlin, Germany: Springer-Verlag, 2001, pp. 181–226.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892.

[4] Song, Z.; Hutangkabodee, S.; Zweiri, Y.H.; Seneviratne, L.D.; Althoefer, K., “Identification of soil parameters for unmanned ground vehicles track-terrain interaction dynamics”, SICE 2004 Annual Conference, Volume 3, 4-6 Aug. 2004 Page(s):2255 – 2260.

[5] A. Le, D. Rye, and H. Durrant-Whyte, “Estimation of track-soil interactions for autonomous tracked vehicles,” in Proc. IEEE Int. Conf. Robotics and Automation, 1997, pp. 1388–1393.

[6] IOSSAQUI, Juliano G.; CAMINO, Juan F.; ZAMPIERI, Douglas E. A nonlinear control design for tracked robots with longitudinal slip. *IFAC Proceedings Volumes*, 2011, 44.1: 5932-5937.

[7] CUI, Mingyue. Observer-based adaptive tracking control of wheeled mobile robots with unknown slipping parameters. *IEEE Access*, 2019, 7: 169646-169655.

[8] CHEN, Jian, et al. Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme. *The International Journal of Robotics Research*, 2010, 29.6: 727-747.

[9] JIAN, C. H. E. N.; DONG, S. U. N.; JIE, Y. A. N. G. A receding-horizon formation tracking controller with leader-follower strategies. *IFAC Proceedings Volumes*, 2008, 41.2: 4400-4405.