

TRAJECTORY TRACKING CONTROL FOR NONHOLONOMIC ROBOTS
WITH ACTUATION NOISE AND IMPERFECT COMMUNICATION

by

Frank Lawless

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Electrical Engineering

Charlotte

2022

Approved by:

Dr. Dipankar Maity

Dr. Yogendra Kakad

Dr. Artur Wolek

Dr. Jim Conrad

ABSTRACT

FRANK LAWLESS. Trajectory tracking control for nonholonomic robots with actuation noise and imperfect communication. (Under the direction of DR. DIPANKAR MAITY)

When executing safety-critical missions, tracking algorithms must be dependable, accurate, and have quantifiable tracking performance in non-ideal environments (e.g., actuation noise, delayed communication, and noisy measurement). Trajectory tracking has already been widely explored using sophisticated control tools (e.g., feedback linearization, optimal and adaptive control, and sliding-mode control). However, tracking stability analyses are often centered around Lyapunov designs in an ideal environment with full state observation. The proposed state-feedback trajectory tracking control guarantees high precision trajectory tracking for differential drive robots. Operational bounds on output velocities, heading angle error, and tracking error are formulated and explored to guarantee tolerances in non-ideal conditions.

It is common that autonomous systems will operate using communication devices that provide non-instantaneous positional measurements subjected to package loss, delay time, and inaccuracies. The latter part of this thesis will introduce a new control function to the state-feedback controller to be carried out as an intermittent feedback controller. The control function is designed as the optimal control that minimizes the average energy in the system during the time between each triggering event. The event-generator determines each triggering instance that provides sensor information to the plant based on the main objective of guaranteeing tracking stability. The analysis develops the appropriate communication policy (an integral part of the event-generator) for the open-loop critical time instance such that the tracking error asymptotically converges to subsequently remain bounded under a user-defined tolerance. The communication policy provides a direct relationship between the system parameters (e.g., actuation noise, control parameters, and reference velocity) and the

sporadic communication frequency as well as the demanded velocities on the robot. The theoretical, simulation, and hardware experimentation results demonstrate operability and efficiency of the proposed algorithms in non-ideal environments with set physical limitations and constraints.

ACKNOWLEDGEMENTS

The research presented in this thesis was conducted through the Department of Electrical and Computer Engineering at the William States Lee College of Engineering, under the guidance of Dr. Dipankar Maity. The work was funded in parts by the Faculty Research Grant of the University of North Carolina at Charlotte.

I cannot overstate the value of the dedication Dr. Maity has to his students. He has consistently provided me with exceptional assistance throughout my time working under his guidance. I'm deeply grateful; I know that he has assisted me in discovering my passion for research.

I would like to thank my parents for their never ending devotion and belief in me. Thanks to my brother for inspiring me to become who I am today. Finally, thanks to my fiancée for her constant support and encouragement during this time.

TABLE OF CONTENTS

LIST OF FIGURES	viii
LIST OF SYMBOLS	ix
CHAPTER 1: INTRODUCTION	2
1.1. Literature Review	4
1.2. Contributions	6
1.3. Problem Formulation	8
CHAPTER 2: STATE-FEEDBACK CONTROLLER	9
2.1. Tracking Performance	13
2.2. Simulation Results	18
2.2.1. Experiment 1: Controller Performance	18
CHAPTER 3: EFFECTS OF ACTUATION NOISE AND MEASUREMENT DELAY	21
3.1. Bound on the Tracking Error	23
3.2. Simulation Results	27
3.2.1. Experiment 2: The Charlotte Logo	28
3.2.2. Experiment 3: Traffic Circle	28
3.2.3. Experiment 4: Lane Change	29
3.3. Hardware Results	31
3.3.1. The Charlotte Logo	31
3.3.2. Experiment 5: City Street	31

	vii
CHAPTER 4: INTERMITTENT-FEEDBACK CONTROLLER	33
4.1. Cases of Tracking Error	36
4.1.1. Case 1: Converging Tracking Error	36
4.1.2. Case 2: Satisfied Tracking Error	40
4.2. Choice of p_k	41
4.3. Simulation Results	42
4.3.1. Experiment 6: Intermittent Feedback Control	43
4.3.2. Experiment 7: Lane Change with Intermittent Feed- back	44
CHAPTER 5: CONCLUSION	46
5.1. Discussion	48
CHAPTER 6: SOFTWARE AND HARDWARE IMPLEMENTATION DETAILS	50
6.1. Simulations	50
6.2. Hardware Experiments	51
6.2.1. TurtleBot3 Startup	52
6.2.2. Running an Experiment	53
6.2.3. Trajectory Planning Program	53
REFERENCES	55
APPENDIX A: Optimal Steering of Linear Systems	60

LIST OF FIGURES

FIGURE 2.1: Base-point dynamics of a differential drive robot	9
FIGURE 2.2: Continuous state-feedback control system	13
FIGURE 2.3: Control parameter comparison.	19
FIGURE 2.4: Velocity comparison in both cases of Fig. 2.3.	20
FIGURE 3.1: UNC Charlotte logo: simulation and hardware results	28
FIGURE 3.2: UNC Charlotte campus traffic circle: error bounds	29
FIGURE 3.3: Lane change scenario: noise and delay	30
FIGURE 3.4: City street traversal	32
FIGURE 4.1: Intermittent feedback control system	33
FIGURE 4.2: Intermittent feedback performance with a circular reference trajectory	43
FIGURE 4.3: Performance characteristics of (4.14) in Fig. 4.2	44
FIGURE 4.4: Lane change scenario: intermittent feedback	45
FIGURE 6.1: Example of appropriate marker locations	52
FIGURE 6.2: Trajectory planning features	54

LIST OF SYMBOLS

Throughout this thesis, scalars are denoted by lowercase symbols (e.g., x, y, θ), vectors are denoted by bold lowercase symbols (e.g., \mathbf{e}), and matrices by uppercase symbols (e.g., J). The following table lists all of the system, function, control, and physical parameters in the order that they are presented.

Parameter	Description	Type
\mathbf{z}_c	Centroid position	position
\mathbf{z}_b	Base-point position	position
\mathbf{z}_r	Reference position	position
ℓ	Distance between base-point and centroid	distance
v	Linear velocity	velocity
w	Angular velocity	velocity
θ	Heading angle	degree
\mathbf{e}	Tracking error	distance
\mathcal{K}	State-feedback control function	–
k	Proportional gain control	–
m	Saturation control	–
t^*	Control switching point	time
l	Lipschitz constant	–
V	Lyapunov function	–
v_r	Linear reference velocity	velocity
w_r	Angular reference velocity	velocity
v_{max}	Maximum linear velocity	velocity
w_{max}	Maximum angular velocity	velocity
n_v	Linear actuation noise	–
n_w	Angular actuation noise	–

\bar{n}_v	Maximum linear actuation noise	–
\bar{n}_w	Maximum angular actuation noise	–
δ	Communication delay	time
$\hat{\theta}$	Estimated heading angle	degree
θ_e	Heading angle error	degree
ϵ	Tracking error tolerance	distance
t_k	Triggering instance	time
\hat{z}_b	Estimated base-point position	position
\hat{e}	Estimated tracking error	distance
Δ	Base-point estimation error	distance
\mathbf{u}	Intermittent feedback control function	–
t_f	Triggering function	time
p_k	Intermittent communication control	–
\bar{p}	Maximum intermittent communication control	–
N	Number of triggers until desired tolerance	–
e_{upper}	Tracking error upper bound	distance

PREFACE

The objective of this thesis is to provide efficient trajectory tracking algorithms with theoretical guarantees. The developed theory is based on standard control theoretic tools that are easy to interpret and implement for researchers interested in controls and trajectory tracking for nonholonomic robots. It will discuss how to solve realistic trajectory tracking problems by presenting a clear solution with a concrete step-by-step analysis. The basis of this work uses a “state-feedback linearization” control approach which is implemented in the analysis by using the nonlinear dynamics of a nonholonomic differential drive robot.

I have structured the text in a chronological manner. The chapters are intended to build on the previous one to further improve robustness by sequentially adding real-world environmental components. There are three main parts to the analysis: Part 1 includes Chapters 1 to 2 of which discuss the different approaches and basic analysis of the state-feedback controller. Part 2 adds environmental factors (hardware noise and sensor delay) to Part 1 in Chapter 3. Parallel to Part 2, Part 3 introduces a intelligent communication and sensing mechanism in Chapter 4.

In each chapter, I have provided simulations of the proposed controller with various scenarios that corroborates the theoretical analysis and will help the reader internalize the key concepts. I have also fully implemented the controller (which considers real-world aspects such as “actuation noise” and “communication delay”) into hardware to demonstrate realistic scenarios and some appealing applications. In order to provide users an easy way to replicate the experiments demonstrated in this thesis, I have developed an user friendly “trajectory planning” Python program. At the end of the thesis in Chapter 6 there are instructions for the reader to carry out these experiments. Additionally, the code and setup steps to replicate all of the simulations presented in this thesis are provided in the instructions.

CHAPTER 1: INTRODUCTION

Autonomous robotic controls are essential to society with healthcare, military, and safety applications (see Refs. [1–3]). In many cases (e.g., self-driving vehicles, medical assistance robots, and bomb disposal robots), autonomous maneuvers must be completed in a pre-determined time period. Common path planning approaches designed for robotic tracking controls such as Refs. [4,5] are designed without the consideration of where the robot will be with time. A trajectory planning approach considers both the position and time of the robot at each pre-determined instance. The advantages of a velocity based controller centered around a design to guarantee the execution stage of a trajectory planning approach fit directly in these high responsibility based applications.

The authors in Ref. [6] explore safe autonomous control. One major statement they consider is, “...in a more realistic setting, how to deal with the existence of uncertainty remains a problem”. They analyze a probabilistic case study for ‘lane changing’ and ‘intersection’ scenarios by incorporating the finite state machine method in Ref. [7]. The low-level trajectory tracking controller must work at high update frequencies to help provide a fast updating control barrier function. When the control scheme is intruded to disturbance, the barrier functions must provide a quantifiable result. To eliminate the uncertainty, it is greatly useful when the controller provides a relationship between tracking performance and time, such that the barriers are time-variant. This thesis explores the uses of the proposed controller in a similar ‘lane change’ simulation and hardware experiment. It is noteworthy that the tracking error and velocity bounds are shown to be significant to plan and execute safety-critical maneuvers.

In this thesis, a novel state-feedback controller is proposed with continuous and

intermittent feedback to provide trajectory tracking for nonholonomic robots. First, the tracking performance is guaranteed with continuous feedback under disturbance by analyzing the system with noise and measurement delay. Noise is commonly represented as hardware disturbances. In this analysis, noise is implemented in the dynamics as linear and angular velocity actuation noise. Measurement delay is a major factor when communicating with positional estimation devices such as GPS and cameras. Delay is incorporated in the analysis, and it is shown that an estimated heading angle is needed to determine the direction of the robot when only 2D positional coordinates are provided. Then, the track performance is guaranteed with intermittent feedback to consider limited sensor communication to the robot. The analysis for the intermittent feedback portion develops a communication policy dependent on the system parameters such as the actuation noise and the desired performance.

The analysis is completed by implementing the nonholonomic dynamics of a differential drive robot with a linear and angular velocity scheme. The necessary constraints to guarantee stability and tracking performance are developed. Stability is guaranteed when introduced to noisy environments and measurement delay through a Lyapunov approach. In applications where an error tolerance is provided (i.e., safety-critical), the controller must be able to adjust the tracking bounds with disturbances to not surpass the desired performance. The analysis shows that the tracking error bounds are adjustable and time-varying. The linear and angular velocity bounds are provided a relationship with the control parameters to easily adjust the magnitudes to user-desired maximums in the controller. To accomplish this, relationships between the physical constraints and control velocities are formulated with dependence on the reference trajectory and control parameters.

1.1 Literature Review

Trajectory tracking control problems have been approached in numerous ways - commonly adaptive control methods [8–14], sliding mode control [15–25], kinematic approaches [26–29], neural network based learning methods [30–33], and look ahead approaches [34–38]

Ref. [8] uses a Lyapunov approach to derive the tracking performance of their proposed adaptive feedback linearization controller. Their results show the advantages in tracking performance of the adaptive controller versus a non-adaptive controller. However, the cost of linearizing the system dynamics by approximation is demonstrated during more complex trajectories (i.e., sharp turns).

Sliding mode control approaches such as in Ref. [15] are often developed to avoid singularity commonly found in kinematic controls. The tracking performance is bounded through a Lyapunov approach, however, it is achieved without time dependence. When stability is analyzed without a time dependence, the theoretically derived tracking performance is time-invariant resulting in an unknown of how long the tracking performance increases. Furthermore, a key observation is that their second-order subsystem introduces the sliding mode surface which relies on the heading angle error. When the controller is implemented, the heading angle must be observed using tools which provide orientational measurements.

Neural network based controllers have an advantage by providing output feedback control without the knowledge of kinematics. In Ref. [30], the authors propose an observer-based neural network control of a differential drive robot. The control design inherently derives estimations for the angular and linear velocities. The learning controller is based on neural weights. Due to the nature of a weighted-based control, tracking performance will converge to the reference trajectory in an inconsistent manner. When the system is introduced to disturbance, this characteristic will be exaggerated and often tracking performance will become unstable.

A look-ahead control Refs. [34, 35] tracks an offset ‘look-ahead’ point from the robot to follow a provided reference trajectory. The control parameters are non-proportional, which require the eigenvalues of the error dynamics to be computed in order to be properly tuned. The error dynamics are approximated via linearization where its convergences rely on the shape of the reference trajectory. It is not guaranteed that the error will converge to zero or a constant value. Furthermore, the control only considers the angular velocity which eliminates any control over the robot’s linear velocity.

State-feedback control approaches Refs. [39–41] provide time dependent analyses which allow for the tracking performance to be quantified. Ref. [39] derives a continuous vision-based error model to adjust the linear and angular velocities. The control strategy uses a state-feedback control and a model predictive control. The state-feedback controller is introduced to eliminate bounded uncertainties due to the nominal control. A neural network is then used to solve the feedback gain such that the state variables can be constrained. The paper demonstrates the robustness of a state-feedback controller as it is the underlying control.

Event-triggered controls are used to transition from a continuous feedback control scheme to intermittent feedback. The authors in Ref. [42] develop an event-based control scheme for linear systems. They compare the results with the sporadic control versus a non-continuous but periodically active feedback control scheme. Their conclusion is that the event-triggered control provides better performance with a reduction to the control action frequency.

Ref. [43] develops a state observer based intermittent feedback control extended for a state-feedback control with added output noise. The stability analysis shows that the design provides a guaranteed observation error which directly determines the communication frequency. This approach is extended in this thesis as the estimated positional error.

An event-based tracking control of nonholonomic robots without velocity measurements was studied in Ref. [44]. In Ref. [45] an event-triggered control is implemented with constrained disturbances for unicycle dynamics. The authors in Ref. [46] provide an event-triggered control with nonlinear state-feedback for the centroid dynamics of a differential drive robot.

One major objective of this thesis is to provide a functional control design that can be smoothly implemented with new robotic projects. The authors in Ref. [47] have proposed a trajectory tracking and planning human-in-the-loop approach that has served as a large motivation of this thesis. Their state-feedback linearization controller provides a straightforward tuning relationship for the tracking performance and velocities. They mention that the design lacks velocity saturation to consider physical hardware constraints and stability guarantees with input/actuation noise or communication interference that appears in realistic environments. These drawbacks are fully addressed in this thesis, and the results show that the work proposed may be fully implemented with hardware for a similar trajectory tracking human-in-the-loop design.

1.2 Contributions

The contributions of this thesis are as follows. (i) A state-feedback controller is proposed which provides a linear and saturated control response. The saturated control is implemented to remove the controller velocities' dependence on tracking error, and the linear control provides exponential tracking. (ii) The time-varying tracking error is quantified through a Lyapunov approach for both the linear and saturated control cases. The quantified error provides the exact time-varying tracking error of the robot. (iii) The linear and angular output velocities in the controller are upper-bounded based on the reference trajectory, positional error, and controller parameters. (iv) A bounded relationship between the maximum output velocities and adjustable control parameters k, m is provided for tuning. (v) Measurement delay

and noisy dynamics are introduced to the controller. As a result, the heading angle is estimated and bounded with the induced measurement delay. We derive bounds on the noise and delay in order to guarantee bounded tracking error. (vi) An intermittent state-feedback linearization controller is proposed which provides an optimal control function that minimizes the average energy. Furthermore, an efficient communication policy is provided which directly depends on the disturbances and system parameters. (vii) The linear and angular velocities of the robot are bounded by the disturbance and a tunable parameter p_k . We further analyze how the communication intermittency and the robot velocities are depended on this parameter.

The overview of this thesis is as follows. In Chapter 2 the base-point dynamics of a differential drive robot and controller are proposed. Later in the introduction to the chapter, the control function is analyzed under the Lipschitz continuous function for each control case. In the next Section 2.1, tracking error is quantified and the velocities are bounded. Furthermore, the velocity bounds are analyzed to provide relationship between the control parameters and maximum velocities. In Chapter 3 actuation noise is introduced to the dynamics and the heading angle is estimated as result of the induced measurement delay. Then, in Section 3.1 the tracking performance is upper-bounded under noisy dynamics and measurement delay. In Chapter 2 and Chapter 3 the controller and control parameters are demonstrated through simulations in Sections 2.2 and 3.2. In Chapter 3 Section 3.3 hardware results are provided to demonstrate the controller with real world disturbances and measurement delay. Chapter 4 introduces the proposed intermittent state-feedback linearization controller. In Section 4.1 the tracking error is bounded by the observed error at each triggering time and estimated error. The system's energy and communication frequency are provided an adjustable relationship in Section 4.2. The intermittent feedback control is demonstrated with simulation results in Section 4.3. In Chapter 5 the thesis is concluded and in Section 5.1 there is a short discussion of current and

alternative works. Finally, the instructions to replicate the simulation and hardware experiments are provided in Chapter 6.

1.3 Problem Formulation

Given a reference trajectory, we desire a time dependent tracking approach for a differential drive robot. The tracking point is located at an offset point from the centroid of the robot by distance ℓ . The tracking error $\mathbf{e}(t)$ must be quantified $\forall t$ and converge to zero as $t \rightarrow \infty$.

The linear and angular velocities outputted by the controller must be bounded to maximum speeds, v_{max} and w_{max} respectively. The user-defined velocity bounds must saturate and have the ability to be tuned by adjusting the control parameters.

The stability analysis must include additive input/actuation noise on the dynamics and measurement delay in the controller. The analysis must provide an upper bound for $\mathbf{e}(t)$ which guarantees the tracking error is bounded under a tolerance $\|\mathbf{e}(t)\| \leq \epsilon, \forall t$.

The continuous feedback time dependent tracking controller must then advance to an intermittent feedback control. The intermittent feedback control must provide limited communication while considering actuation noise. Similar to the continuous feedback controller, the communication policy must guarantee that $\mathbf{e}(t)$ is bounded by a user-desired tolerance $\epsilon, \forall t \geq t_N$.

CHAPTER 2: STATE-FEEDBACK CONTROLLER

We first need to model the dynamics of our robot. We know the configuration transition equation for a differential drive robot from Eq. (13.17) in Ref. [48]. We can derive the linear and angular velocity scheme for the centroid dynamics of a differential drive robot as

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}, \quad (2.1)$$

where $x(t), y(t)$ denote the position of the robot centroid with respect to a static global frame and $\theta(t)$ denotes its heading angle with respect to the x-axis (see Fig. 2.1). The linear and angular velocity inputs at time t are denoted by $v(t)$ and $w(t)$, respectively.

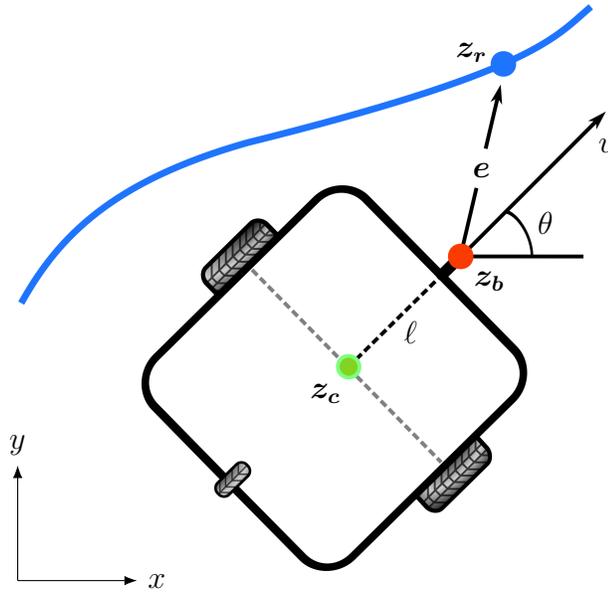


Figure 2.1: Base-point dynamics of a differential drive robot

An offset point \mathbf{z}_b , which is at a distance ℓ from the centroid, is required to track the given reference trajectory. More specifically, we are interested in the point

$$\mathbf{z}_b = \begin{bmatrix} x_b \\ y_b \end{bmatrix} = \begin{bmatrix} x + \ell \cos \theta \\ y + \ell \sin \theta \end{bmatrix}. \quad (2.2)$$

Therefore the base-point dynamics are the following:

$$\begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\ell \sin \theta \\ \sin \theta & \ell \cos \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}. \quad (2.3)$$

The difference between the base-point \mathbf{z}_b and reference position at time t is defined as the tracking error $\mathbf{e}(t)$:

$$\mathbf{e}(t) = \begin{bmatrix} e_x(t) \\ e_y(t) \end{bmatrix} \triangleq \begin{bmatrix} x_r(t) - x_b(t) \\ y_r(t) - y_b(t) \end{bmatrix}, \quad (2.4)$$

where $\mathbf{z}_r(t) = [x_r(t), y_r(t)]^\top$ denotes the coordinates of the reference trajectory at time t . Taking the derivative of (2.4) and then using the dynamics from (2.3), we obtain

$$\dot{\mathbf{e}} = \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \end{bmatrix} - J(\theta) \begin{bmatrix} v \\ w \end{bmatrix}, \quad (2.5)$$

where

$$J(\theta) = \begin{bmatrix} \cos \theta & -\ell \sin \theta \\ \sin \theta & \ell \cos \theta \end{bmatrix}. \quad (2.6)$$

Therefore, our objective is to design a controller to ensure $\mathbf{e}(t)$ goes to zero or remains bounded in presence of actuation noise and measurement delay.

The proposed state-feedback controller in this thesis is as follows

$$\begin{bmatrix} v \\ w \end{bmatrix} = J(\theta)^{-1} \left[\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \end{bmatrix} + \mathcal{K}(\mathbf{e}) \right], \quad (2.7)$$

where the control function $\mathcal{K} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is considered to be the following:

$$\mathcal{K}(\mathbf{e}) = \begin{cases} k\mathbf{e}, & \text{if } \|\mathbf{e}\| \leq m, \\ km \frac{\mathbf{e}}{\|\mathbf{e}\|}, & \text{otherwise,} \end{cases} \quad (2.8)$$

where $k > 0$ and $m \geq 0$ are two user chosen parameters. The effects of k and m on the tracking performance will be discussed in detail, and the choice of these parameters will be discussed as well. When (2.7) is substituted into (2.5), we obtain

$$\dot{\mathbf{e}} = -\mathcal{K}(\mathbf{e}). \quad (2.9)$$

Remark 1 *When $\mathcal{K}(\mathbf{e}) = k\mathbf{e}$, then the error \mathbf{e} converges to zero exponentially fast with rate k . On the other hand, when $\mathcal{K}(\mathbf{e})$ is the saturated function, the error converges to zero linearly with time. Thus, the proposed control law ensures exponential convergence when the error magnitude is less than the user-defined threshold m . Otherwise a linear convergence is guaranteed. Furthermore, the magnitude of $\mathcal{K}(\mathbf{e})$ is upper bounded by km at any time t , where both k and m are chosen by the user. We will show later (Lemmas 3, 4) that the proposed design in (2.8) helps us to bound the magnitude of the linear and angular velocities directly.*

Next, we state an important characteristic of our proposed controller which ensures that a unique solution exists to the differential equation (2.9).

Lemma 1 *$\mathcal{K}(\mathbf{e})$ is a Lipschitz function with the Lipschitz constant $l = 2k$.*

Proof: The proof follows directly from the expression of $\mathcal{K}(\mathbf{e})$ in (2.8) and the definition of Lipschitz functions. To see this, consider the case \mathbf{e}_1 and \mathbf{e}_2 such that $\|\mathbf{e}_1\|, \|\mathbf{e}_2\| \leq m$. Then,

$$\begin{aligned}\|\mathcal{K}(\mathbf{e}_1) - \mathcal{K}(\mathbf{e}_2)\| &= \|k\mathbf{e}_1 - k\mathbf{e}_2\| \\ &= k\|\mathbf{e}_1 - \mathbf{e}_2\| = l\|\mathbf{e}_1 - \mathbf{e}_2\|.\end{aligned}$$

Now, consider the second case that $\|\mathbf{e}_1\|, \|\mathbf{e}_2\| > m$. Then,

$$\begin{aligned}\|\mathcal{K}(\mathbf{e}_1) - \mathcal{K}(\mathbf{e}_2)\| &= \left\| \frac{km}{\|\mathbf{e}_1\|} \mathbf{e}_1 - \frac{km}{\|\mathbf{e}_2\|} \mathbf{e}_2 \right\| \\ &= \frac{km}{\|\mathbf{e}_1\| \|\mathbf{e}_2\|} \left\| \|\mathbf{e}_2\| \mathbf{e}_1 - \|\mathbf{e}_1\| \mathbf{e}_2 \right\| \\ &= \frac{km}{\|\mathbf{e}_1\| \|\mathbf{e}_2\|} \left\| \|\mathbf{e}_2\| (\mathbf{e}_1 - \mathbf{e}_2) + (\|\mathbf{e}_2\| - \|\mathbf{e}_1\|) \mathbf{e}_2 \right\| \\ &\leq \frac{km}{\|\mathbf{e}_1\| \|\mathbf{e}_2\|} \left(\|\mathbf{e}_2\| \|\mathbf{e}_1 - \mathbf{e}_2\| + \|\mathbf{e}_2\| \left| \|\mathbf{e}_1\| - \|\mathbf{e}_2\| \right| \right) \\ &\leq k(\|\mathbf{e}_1 - \mathbf{e}_2\| + \left| \|\mathbf{e}_1\| - \|\mathbf{e}_2\| \right|) \\ &\leq 2k\|\mathbf{e}_1 - \mathbf{e}_2\| = l\|\mathbf{e}_1 - \mathbf{e}_2\|.\end{aligned}$$

Finally, when $\|\mathbf{e}_1\| \leq m$ and $\|\mathbf{e}_2\| > m$,

$$\begin{aligned}\|\mathcal{K}(\mathbf{e}_1) - \mathcal{K}(\mathbf{e}_2)\| &= \left\| k\mathbf{e}_1 - \frac{km}{\|\mathbf{e}_2\|} \mathbf{e}_2 \right\| \\ &\leq k\|\mathbf{e}_1 - \mathbf{e}_2\| + \left| k - \frac{km}{\|\mathbf{e}_2\|} \right| \|\mathbf{e}_2\| \\ &= k\|\mathbf{e}_1 - \mathbf{e}_2\| + k\|\mathbf{e}_2\| - km \\ &\leq k\|\mathbf{e}_1 - \mathbf{e}_2\| + k\|\mathbf{e}_1 - \mathbf{e}_2\| \\ &= 2k\|\mathbf{e}_1 - \mathbf{e}_2\| = l\|\mathbf{e}_1 - \mathbf{e}_2\|.\end{aligned}$$

■

Lipschitz continuity is a desired characteristic in ensuring that a differential equa-

tion has a unique well-defined solution (see Ref. [49]).

Remark 2 *Since $\mathcal{K}(\mathbf{e})$ is a Lipschitz continuous function with the maximum Lipschitz constant of $l = 2k$, there is a unique solution to (2.9) for all time even when the controller (2.7) switches between the linear and nonlinear control modes. This characteristic allows the tracking error stability to be analysed through a Lyapunov approach in the next section.*

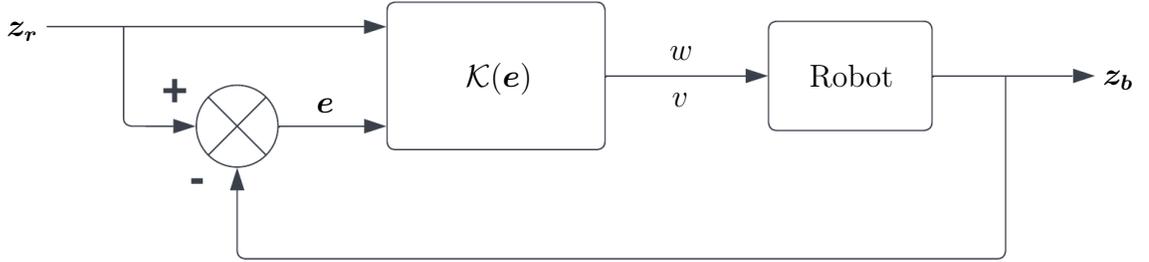


Figure 2.2: Continuous state-feedback control system

2.1 Tracking Performance

In this section, we analyze the error dynamics (2.9) to quantify the tracking performance. For the proposed controller (2.8), we provide the exact error magnitude for all time.

Lemma 2 *Given an initial error $\|\mathbf{e}(0)\| > m$, the tracking error at any time t is given by*

$$\|\mathbf{e}(t)\| = \begin{cases} \|\mathbf{e}(0)\| - kmt, & \text{if } t \leq t^*, \\ me^{-k(t-t^*)}, & \text{otherwise,} \end{cases} \quad (2.10)$$

where $t^* \triangleq \frac{\|\mathbf{e}(0)\| - m}{mk}$ is the time such that $\|\mathbf{e}(t^*)\| = m$. On the other hand, if $\|\mathbf{e}(0)\| \leq m$, then $\|\mathbf{e}(t)\| = \|\mathbf{e}(0)\|e^{-kt}$.

Proof: We consider a Lyapunov based approach to prove this lemma. To that end, we take the Lyapunov function

$$V = \mathbf{e}^\top \mathbf{e}. \quad (2.11)$$

Taking the derivative of V , and using the dynamics (2.9) we obtain

$$\begin{aligned} \dot{V} &= \dot{\mathbf{e}}^\top \mathbf{e} + \mathbf{e}^\top \dot{\mathbf{e}} = -2\mathbf{e}^\top \mathcal{K}(\mathbf{e}) \\ &= \begin{cases} -2mk \frac{\mathbf{e}^\top \mathbf{e}}{\|\mathbf{e}\|}, & \text{if } \|\mathbf{e}\| > m \\ -2k\mathbf{e}^\top \mathbf{e}, & \text{otherwise} \end{cases} \\ &= \begin{cases} -2mk\sqrt{V}, & \text{if } V > m^2 \\ -2kV, & \text{otherwise.} \end{cases} \end{aligned} \quad (2.12)$$

From (2.12) we notice that $V(t)$ is a strictly decreasing function with time.

When $\|\mathbf{e}(0)\| > m$ (or equivalently $V(0) > m^2$), V follows the differential equation $\dot{V} = -2km\sqrt{V}$ according to (2.12). Integrating \dot{V} , we obtain

$$\sqrt{V(t)} = \sqrt{V(0)} - mkt.$$

Therefore, $\sqrt{V(t)}$ (or equivalently $\|\mathbf{e}(t)\|$) decreases linearly with time and at time $t^* = (\|\mathbf{e}(0)\| - m)/mk$, we have $\|\mathbf{e}(t^*)\| = \sqrt{V(t^*)} = m$. At this moment, V will follow the differential equation $\dot{V} = -2kV$ and will have the solution $V(t) = V(t^*)e^{-2k(t-t^*)} = m^2 e^{-2k(t-t^*)}$, which is exponentially decreasing with time. Thus, we may write compactly,

$$\sqrt{V(t)} = \begin{cases} \sqrt{V(0)} - kmt, & \text{if } t \leq t^* \\ me^{-k(t-t^*)}, & \text{otherwise.} \end{cases} \quad (2.13)$$

On the other hand, if $V(0) \leq m^2$ then,

$$V(t) = e^{-2kt}V(0), \quad t \geq 0. \quad (2.14)$$

The lemma is proved by substituting $V = \|\mathbf{e}\|^2$ in equations (2.13) and (2.14) where the final result is (2.10). ■

Corollary 1 *The tracking error $\|\mathbf{e}(t)\|$ converges to zero at a linear rate until switching to an exponential rate after time t^* .*

Proof: The proof follows directly from the expression of $\|\mathbf{e}(t)\|$ in (2.10) where we notice that for an initial error magnitude higher than m , the error converges to zero in a linear manner until t^* and exponentially afterwards. Otherwise when the initial error magnitude is lower than m , the error converges to zero exponentially for all time. ■

Next, we analyze the upper bounds on the linear and angular velocities v and w . The following lemma provides an upper bound for each velocity. These upper bounds can be tuned via the user-defined parameters k, m .

Lemma 3 *For the proposed controller, the linear and angular velocities are upper bounded by the following*

$$|v(t)| \leq v_r(t) + k \min(m, \|\mathbf{e}(t)\|) \quad (2.15a)$$

$$|w(t)| \leq w_r(t) + \frac{k \min(m, \|\mathbf{e}(t)\|)}{\ell}, \quad (2.15b)$$

where $v_r = \|\dot{x}_r, \dot{y}_r\|^\top$ is the linear reference velocity of the given trajectory and $w_r = \frac{v_r}{\ell}$.

Proof: From (2.7), when $\|\mathbf{e}(t)\| \leq m$ we can directly obtain

$$\begin{aligned} v &= \dot{x}_r \cos \theta + \dot{y}_r \sin \theta + k(\cos \theta e_x + \sin \theta e_y), \\ w &= \frac{-\sin \theta \dot{x}_r + \cos \theta \dot{y}_r}{\ell} + \frac{k(-\sin \theta e_x + \cos \theta e_y)}{\ell}. \end{aligned}$$

Using the trigonometric inequality $a \sin \theta + b \cos \theta \leq \sqrt{a^2 + b^2}$, we obtain

$$\begin{aligned} |v| &\leq \|[\dot{x}_r, \dot{y}_r]^\top\| + k\|\mathbf{e}\|, \\ |w| &\leq \frac{\|[\dot{x}_r, \dot{y}_r]^\top\|}{\ell} + \frac{k}{\ell}\|\mathbf{e}\|. \end{aligned} \tag{2.16}$$

Similarly, (2.7) is upper bounded when $\|\mathbf{e}(t)\| > m$.

$$\begin{aligned} |v| &\leq \|[\dot{x}_r, \dot{y}_r]^\top\| + mk, \\ |w| &\leq \frac{\|[\dot{x}_r, \dot{y}_r]^\top\|}{\ell} + \frac{mk}{\ell}. \end{aligned} \tag{2.17}$$

The lemma is proved by combining (2.16) and (2.17) and substituting the expressions v_r and w_r . ■

Remark 3 From (2.15b) we can see that the upper bound of the angular velocity is inversely proportional to the physical distance between the centroid and base-point distance ℓ . This relationship is important to note for certain applications (e.g., when tight turns are required with constrained velocities). Robots which have higher ℓ distances, demand lower velocities in relation to the turn's sharpness. Also, we can see that as $\max\|\mathbf{e}\|$ increases, the upper bound increases until $\|\mathbf{e}\| = m$. When taking in consideration physical constraints, it may be desired to decrease the maximum velocities at high positional error which is accomplished by introducing the saturated control. This controller decreases the linear and angular velocities by normalizing the tracking error \mathbf{e} with the introduction of $\frac{1}{\|\mathbf{e}\|}$.

The next lemma analyzes how the control parameters m and k can be tuned to

guarantee maximum linear and angular velocities.

Lemma 4 (2.15) *are tuned for user-desired velocity bounds by using the control parameters k, m in the following relationship:*

$$mk \leq \min(v_{\max}, \ell w_{\max}) - v_r. \quad (2.18)$$

Proof: Consider v_{\max} and w_{\max} the maximum velocity of v and w respectively. When these constraints are substituted into (2.15) during the time $\|e(t)\| > m$ we obtain,

$$\begin{aligned} v_{\max} &\geq v_r + mk, \\ w_{\max} &\geq w_r + \frac{mk}{\ell}. \end{aligned}$$

The lemma is proved after solving for mk which provides two inequalities that are compacted into one relationship between the minimum values of v_{\max} and ℓw_{\max} in (2.18). ■

Remark 4 *The linear and angular velocities are physically limited in robotic systems. Using the relationship in (2.18), v_{\max} and w_{\max} can be directly tuned by m, k to saturate the linear and angular velocities in the controller. Notice that the bounds are only dependent on the case $\|e(t)\| > m$ where the dependence on the tracking error is normalized.*

In this chapter we proposed and analyzed the state-feedback controller in an ideal environment. The controller provided exponential stability without the cost of spiked velocities as a result of the switching control function. In (2.18) we are provided a direct relationship to the control parameters and reference trajectory to tune the

saturated velocities. In the next chapter, we discuss the controller in a non-ideal noisy environment with the goal to observe similar results.

2.2 Simulation Results

The following simulations are completed using MATLAB to demonstrate the controller in an ideal environment with zero communication delay and noiseless actuation. The directions to replicate these and all other simulations in this thesis can be found in Chapter 6.

2.2.1 Experiment 1: Controller Performance

Fig. 2.3 demonstrates the tracking characteristics and performance when using the controller (2.7). The robot is initially placed with a large positional error and an initial heading angle of $\theta(0) = \frac{\pi}{4}$ [rad]. In this experiment we study the effects of the design parameter m . To that end, we pick two m values (1 and 0.3, respectively) such that the $m > \|\mathbf{e}(0)\|$ for one case and $m < \|\mathbf{e}(0)\|$ for the other. Case 1 ensures only the linear control is used for the entire duration, whereas Case 2 demonstrates the use of both control modes as defined in (2.8). The reference trajectory is a circle with constant linear and angular velocities, v_r and ω_r , respectively. The robot's trajectory for Case 1 (i.e., $m = 1$) is shown in green and for Case 2 (i.e., $m = 0.3$) in red. The $-k\mathbf{e}$ control ensures an exponential convergence, whereas the $-k\mathbf{e}/\|\mathbf{e}\|$ control ensures a slower linear convergence, which are reflected in the green and red trajectories, respectively. In the figure, the robot is shown at four separate time instances for Case 2. Their respective time instances for the reference trajectory are shown as blue dots.

In Fig. 2.4 we plot the linear and angular velocities along with their corresponding upper-bounds derived from (2.15). The control parameters were chosen for Case 2 from the relationship provided in (2.18) given v_{\max} and w_{\max} . The system parameters of the experiment are as follows: $v_{\max} = 1.5$ [m/s], $w_{\max} = 5$ [rad/s], $v_r = 1$ [m/s], $\ell =$

0.3, for Case 1: $k = 2$, $m = 1$ and for Case 2: $k = 1.3$, $m = 0.3$.

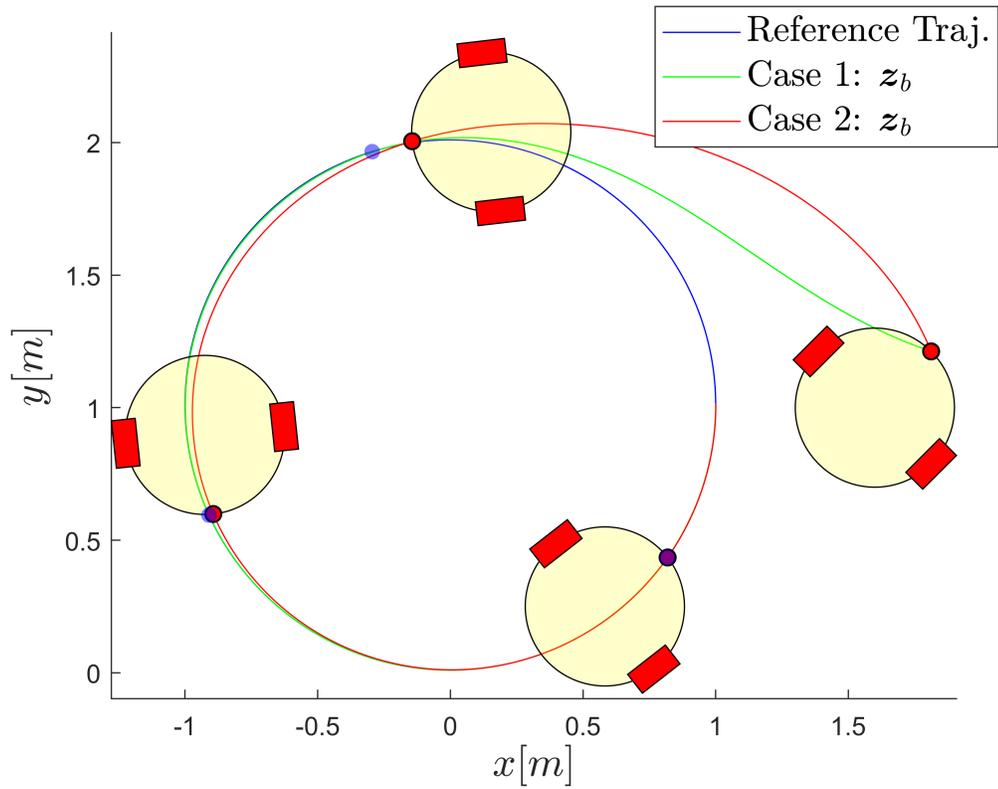


Figure 2.3: Control parameter comparison.

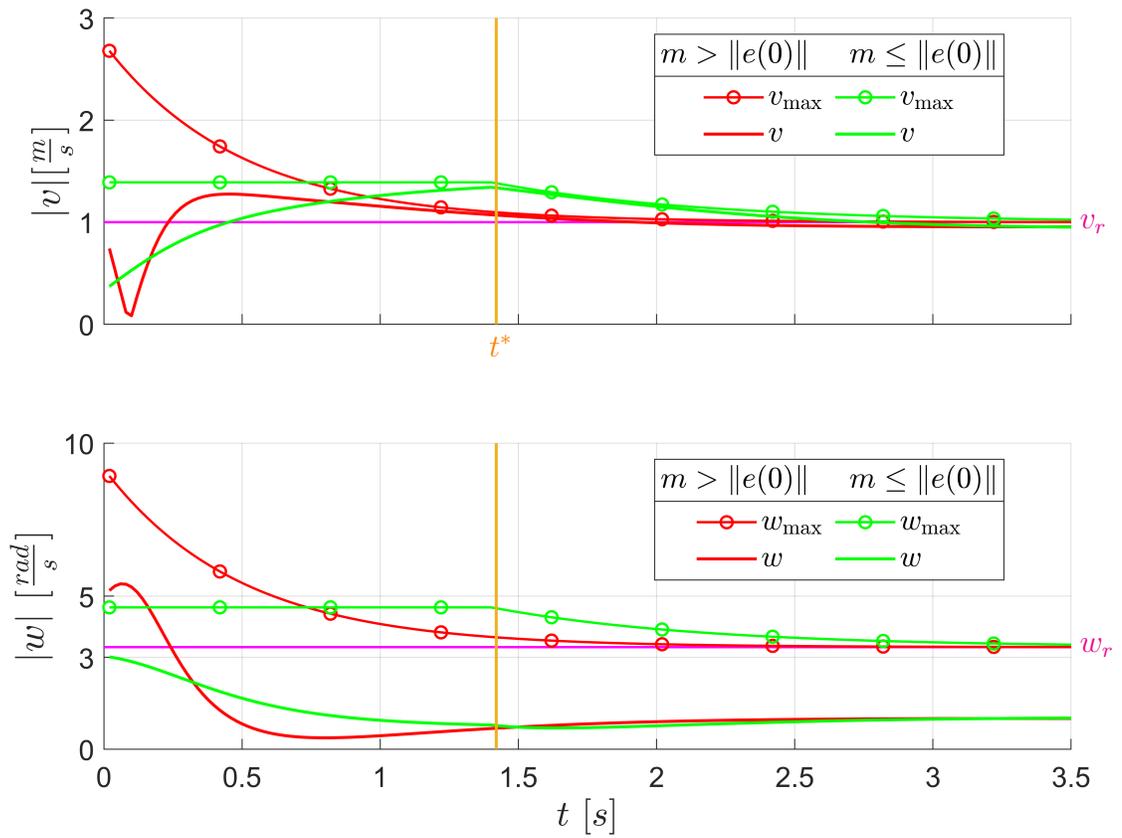


Figure 2.4: Velocity comparison in both cases of Fig. 2.3.

CHAPTER 3: EFFECTS OF ACTUATION NOISE AND MEASUREMENT
DELAY

In the previous chapter we analyzed the error dynamics (2.9) without any disturbance or communication delay. In this chapter we will consider the following extensions: (1) We introduce actuation noise to the linear and angular velocities v and w , (2) The controller only receives information about the position (x_b, y_b) and not the orientation θ , and (3) A nonzero measurement delay δ is introduced to the controller. That is, the controller at time t receives the position information for time $t - \delta$.

Let n_v and n_w denote the unknown but smooth and time-varying input noise on the linear and the angular velocities respectively. Therefore, under noisy actuation, the dynamics (2.3) become

$$\begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -\ell \sin \theta \\ \sin \theta & \ell \cos \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v + n_v \\ w + n_w \end{bmatrix}, \quad (3.1)$$

where we assume that actuation noise is bounded (i.e., $|n_v(t)| \leq \bar{n}_v$ and $|n_w(t)| \leq \bar{n}_w$ for all t).

Recall that our proposed controller in (2.7) requires the perfect knowledge of $\theta(t)$. Since the measurement information does not contain θ , we construct an estimate of $\hat{\theta}$ and use the following updated controller:

$$\begin{bmatrix} v \\ w \end{bmatrix} = J(\hat{\theta})^{-1} \left[\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \end{bmatrix} + \mathcal{K}(e) \right]. \quad (3.2)$$

In the following lemma, the heading angle θ is estimated as $\hat{\theta}$ and the heading angle error θ_e is upper-bounded. We assume that we have knowledge of the control inputs $v(t)$ and $w(t)$ during the time $t \in [t - \delta, t]$. The result shows that the heading angle error is dependent on the angular noise and delay time.

Lemma 5 *The heading angle error $\theta_e(t)$ is bounded by the delay time and maximum angular noise in each measurement delay period by the following inequality:*

$$|\theta_e(t)| \leq \delta \max |n_w(s)|, \quad s \in [t - \delta, t]. \quad (3.3)$$

Proof: $\theta_e(t)$ is defined as

$$\theta_e(t) \triangleq \theta(t) - \hat{\theta}(t), \quad (3.4)$$

where $\theta(t)$ is the actual heading angle and $\hat{\theta}(t)$ is the estimated heading angle. $\dot{\theta}$ is derived from (3.1)

$$\dot{\theta} = w + n_w,$$

where

$$\theta(t) = \theta(t_1) + \int_{t_1}^t \dot{\theta}(s) ds,$$

for $t \geq t_1$. Therefore, with $t_1 = t - \delta$, we have

$$\theta(t) = \theta(t - \delta) + \int_{t-\delta}^t w(s) ds + \int_{t-\delta}^t n_w(s) ds. \quad (3.5)$$

We can measure \dot{x}, \dot{y} by the change in position while using an initial guess of θ at $t = 0$. The heading angle estimate is to be

$$\hat{\theta}(t) = \tan^{-1} \left(\frac{\dot{y}_b(t - \delta)}{\dot{x}_b(t - \delta)} \right) + \int_{t-\delta}^t w(s) ds. \quad (3.6)$$

The difference between the actual heading angle (3.5) and the estimated (3.6) is

$$\theta_e(t) = \theta(t) - \hat{\theta}(t) = \int_{t-\delta}^t n_w(s) ds. \quad (3.7)$$

We prove the lemma by upper bounding (3.7) with the delay time and maximum angular noise using maximum L^∞ norm. \blacksquare

In the next section, the heading angle error is constrained in order to upper bound the tracking error with noise and measurement delay.

3.1 Bound on the Tracking Error

This section analyzes the error dynamics for the control (3.2) that includes noise and measurement delay. Through the Lyapunov function proposed previously (2.11), the next lemma proves that under a heading angle constraint the tracking error is upper bounded and guaranteed to asymptotically converge to a constant value. Furthermore, the switching time t^* required for the control (2.7) to switch from the saturated to the linear response is upper bounded.

Lemma 6 $\|\mathbf{e}(t)\|$ is bounded $\forall t$ when $|\theta_e(t)| \leq \frac{\pi}{2}$ by the following:

$$\|\mathbf{e}(t)\| \leq \begin{cases} \sqrt{\left(\|\mathbf{e}(0)\|^2 - \frac{c_3}{\gamma}\right)e^{\gamma t} + \frac{c_3}{\gamma}}, & \text{if } t \leq t^* \\ \sqrt{\left(m^2 - \frac{c_2}{c_4}\right)e^{-c_4(t-t^*)} + \frac{c_2}{c_4}}, & \text{otherwise,} \end{cases} \quad (3.8)$$

where

$$t^* \leq \frac{1}{\gamma} \ln \left(\frac{m^2 - c_3/\gamma}{\|\mathbf{e}(0)\|^2 - c_3/\gamma} \right)$$

is the bounded time before switching from the saturated to linear control.

Proof: By substituting the noisy dynamics (3.1) and control (3.2) into the track-

ing error dynamics (2.5) we obtain,

$$\begin{aligned}
\dot{\mathbf{e}} &= \dot{\mathbf{z}}_r - J(\theta) \left[J(\hat{\theta})^{-1} [\dot{\mathbf{z}}_r + \mathcal{K}(\mathbf{e})] + [n_v \ n_w]^\top \right] \\
&= [I - M(\theta - \hat{\theta})] \dot{\mathbf{z}}_r - M(\theta - \hat{\theta}) \mathcal{K}(\mathbf{e}) - J(\theta) [n_v \ n_w]^\top \\
&= A \dot{\mathbf{z}}_r - B \mathcal{K}(\mathbf{e}) - \mathbf{d},
\end{aligned} \tag{3.9}$$

where

$$\begin{aligned}
M(\theta - \hat{\theta}) &= J(\theta) J(\hat{\theta}), \quad A = I - M(\theta - \hat{\theta}), \\
B &= M(\theta - \hat{\theta}), \quad \mathbf{d} = J(\theta) [n_v \ n_w]^\top.
\end{aligned} \tag{3.10}$$

The error dynamics (3.9) now include the non-linearizable matrices A , B . \mathbf{d} is the added disturbance to the equality, while A and B consider the actual and estimated dynamics $J(\theta)J(\hat{\theta})^{-1}$ which are equivalent to the rotation matrix $M(\theta - \hat{\theta})$. Notice, when there is no disturbance or delay (i.e., $\mathbf{d} = 0$ and $\theta = \hat{\theta}$), $M(\theta - \hat{\theta}) = I$ and (3.9) = (2.9), however, as a result of the added disturbance, $\theta \neq \hat{\theta}$ and (3.9) \neq (2.9). Next, we apply the Lyapunov function (2.11) to (3.9) and obtain

$$\begin{aligned}
\dot{V} &= (A \dot{\mathbf{z}}_r - B \mathcal{K}(\mathbf{e}) - \mathbf{d})^\top \mathbf{e} + \mathbf{e}^\top (A \dot{\mathbf{z}}_r - B \mathcal{K}(\mathbf{e}) - \mathbf{d}) \\
&= -(\mathbf{e}^\top B \mathcal{K}(\mathbf{e}) + \mathcal{K}(\mathbf{e})^\top B^\top \mathbf{e}) \\
&\quad + \dot{\mathbf{z}}_r^\top A^\top \mathbf{e} + \mathbf{e}^\top A \dot{\mathbf{z}}_r - (\mathbf{e}^\top \mathbf{d} + \mathbf{d}^\top \mathbf{e}).
\end{aligned}$$

Through Young's inequality, \dot{V} is upper bounded:

$$\dot{V} \leq -q + \gamma \mathbf{e}^\top \mathbf{e} + \frac{1}{\alpha^2} \dot{\mathbf{z}}_r^\top A^\top A \dot{\mathbf{z}}_r + \frac{1}{\beta^2} \mathbf{d}^\top \mathbf{d}, \tag{3.11}$$

where $q = \mathbf{e}^\top B \mathcal{K}(\mathbf{e}) + \mathcal{K}(\mathbf{e})^\top B^\top \mathbf{e}$ and $\gamma = \alpha^2 + \beta^2$. Further bounding V by its maximum quantities (denoted as \max) and maximum eigenvalues (denoted as λ_{\max}),

we obtain the maximum relationship of the inequality

$$\dot{V} \leq -q + \gamma \|\mathbf{e}\|^2 + \frac{1}{\alpha^2} \lambda_{\max}\{A^\top A\} \|\dot{\mathbf{z}}_r\|^2 + \frac{1}{\beta^2} \max\{\mathbf{d}^\top \mathbf{d}\}. \quad (3.12)$$

We will now analyze the conditional manner of \dot{V} in order to guarantee stability. First, let us look into each component of (3.12).

$$q = \mathbf{e}^\top B \mathcal{K}(\mathbf{e}) + \mathcal{K}(\mathbf{e})^\top B^\top \mathbf{e} = \begin{cases} k \mathbf{e}^\top (B + B^\top) \mathbf{e}, & \|\mathbf{e}\| \leq m \\ \frac{km}{\|\mathbf{e}\|} \mathbf{e}^\top (B + B^\top) \mathbf{e}, & \|\mathbf{e}\| > m \end{cases} \leq 2 \cos(\theta_e) \mathbf{e}^\top \mathcal{K}(\mathbf{e}),$$

where we used (3.10) to obtain

$$B^\top + B = 2 \cos(\theta_e) I. \quad (3.13)$$

We can also see that

$$\lambda_{\max}\{A^\top A\} = 2 \sin^2\left(\frac{\theta_e}{2}\right), \quad \max\{\mathbf{d}^\top \mathbf{d}\} = \bar{n}_v^2 + \bar{n}_w^2 \ell^2.$$

We need $q > 0$ for the Lyapunov analysis and thus we want to ensure $2 \cos(\theta_e) > 0$. Therefore, (3.12) is guaranteed to be stable under the constraint that the heading angle error is bounded by $\theta_e \in [-\pi/2, \pi/2]$. We can now look into the Lyapunov function with both cases of $\mathcal{K}(\mathbf{e})$ as

$$\dot{V} \leq \begin{cases} -(c_1 - \gamma) \|\mathbf{e}\|^2 + c_2, & \text{if } V \leq m^2, \\ \gamma \|\mathbf{e}\|^2 - mc_1 \|\mathbf{e}\| + c_2, & \text{otherwise,} \end{cases} \quad (3.14)$$

where

$$c_1 = 2k \cos(\max \theta_e), \quad c_2 = \frac{2v_r^2}{\alpha^2} \max\{\sin^2(\frac{\theta_e}{2})\} + \frac{\eta}{\beta^2},$$

$$\eta = \bar{n}_v^2 + \bar{n}_w^2 \ell^2.$$

Alternatively, (3.14) can be simplified in terms of V

$$\dot{V} \leq -c_1 \min\{m\sqrt{V}, V\} + \gamma V + c_2, \quad (3.15)$$

In the case $\|\mathbf{e}\| > m$, (3.15) uses the saturated control and \dot{V} becomes

$$\dot{V} \leq -mc_1\sqrt{V} + \gamma V + c_2. \quad (3.16)$$

We can further simplify (3.16) to a linear differential equation by using the inequality $\sqrt{V} > m$ for the first term. This way, for the saturated control case, we obtain the upper bound

$$\|V(t)\| \leq \left(\|V(0)\| - \frac{c_3}{\gamma} \right) e^{\gamma t} + \frac{c_3}{\gamma}, \quad (3.17)$$

where

$$c_3 = m^2 c_1 - c_2.$$

In the other case $\|\mathbf{e}\| \leq m$, (3.15) uses the linear control since $V \leq m\sqrt{V}$. \dot{V} is now

$$\dot{V} \leq -c_4 V + c_2, \quad (3.18)$$

where $c_4 = c_1 - \gamma$. We can now easily solve for $\|V(t)\|$ since (3.18) is the form of a linear differential equation,

$$\|V(t)\| \leq \left(\|V(t^*)\| - \frac{c_2}{c_4} \right) e^{-c_4(t-t^*)} + \frac{c_2}{c_4}. \quad (3.19)$$

The lemma is proved after substituting \mathbf{e} into (3.17) and (3.19) by $V = \|\mathbf{e}\|^2$ which results in the compact equation (3.8). ■

Remark 5 *Notice from (3.8) that when $t \rightarrow \infty$, the linear control is used where $\|\mathbf{e}\|$ converges to the constant*

$$\lim_{t \rightarrow \infty} \|\mathbf{e}(t)\| \leq \sqrt{\frac{c_2}{c_4}}. \quad (3.20)$$

We now know that even with actuation noise and communication, the tracking error will converge in a linear manner until time t^ and exponentially after to a constant value dependent on the measurement delay and noise amount. As expected, the tracking error bounds show that the tracking accuracy will increase with lower amounts of measurement delay and actuation noise. The bounds can be further decreased by adjusting the control parameters. By increasing k , c_1 will increase, and in both cases of (3.8) the bounds will decrease faster with time. Furthermore, (3.20) will decrease with increasing k .*

This concludes the continuous feedback trajectory tracking controller analysis. We have observed that the tracking error is bounded even with actuation noise and communication noise. In the next chapter, the continuous feedback controller will be developed into an intermittent feedback controller. Similar to the previous analysis, the goal of the intermittent analysis is to demonstrate stable and bounded tracking error with relation to the system parameters.

3.2 Simulation Results

The following simulations are developed to visualize and verify the controller performance with actuation noise and communication delay. In these simulations the parameters are chosen by the derived relationships in the previous two chapters, such as the velocity saturation bounds (2.18) and tracking error bounds (3.8).

3.2.1 Experiment 2: The Charlotte Logo

Fig. 3.1 showcases the controller with a reference trajectory drawing UNC Charlotte's logo. This experiment demonstrates the tracking performance provided a complicated trajectory and sharp corners. The hardware results of this reference trajectory are also provided in this figure and are further discussed in the next section. The parameters of the scenario are as follows: $v_{\max} = 0.3$ [m/s], $v_r = 1$ [m/s], $\ell = 0.3$, for Case 1: $k = 1$, $m = 1$, and for Case 2: $k = 1$, $m = 0.3$.

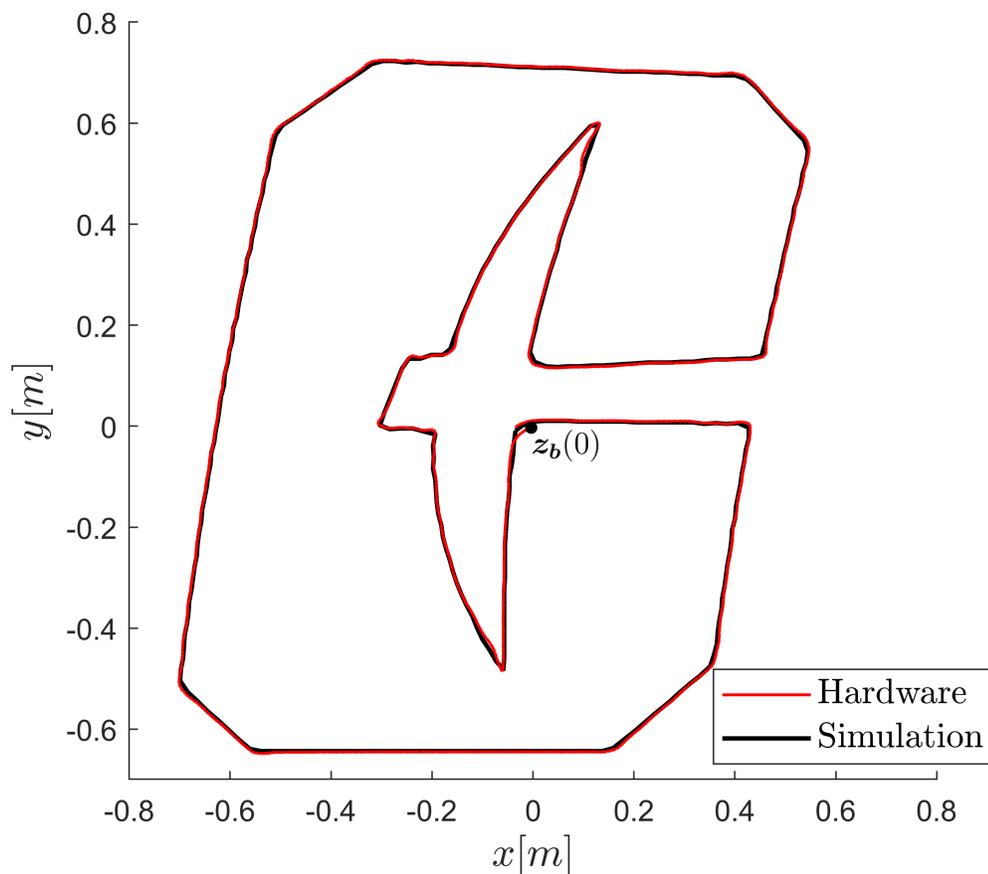


Figure 3.1: UNC Charlotte logo: simulation and hardware results

3.2.2 Experiment 3: Traffic Circle

Fig. 3.2 demonstrates the controller (3.2) with noise and communication delay. The trajectory is a realistic outdoor scenario where the robot is required to traverse a traffic

circle, and the tracking bounds (3.8) are required to guarantee that the robot will remain in the appropriate traffic lane. The grey circles are the time-varying tracking error upper-bounds. The robot starts on the adjacent sidewalk from the reference trajectory to simulate a large initial error. The parameters of the simulations are as follows: $n_v = 0.5v_r$, $n_w = 0.5w_r$, $\ell = 1.5$, $k = 1$, $m = 3$.

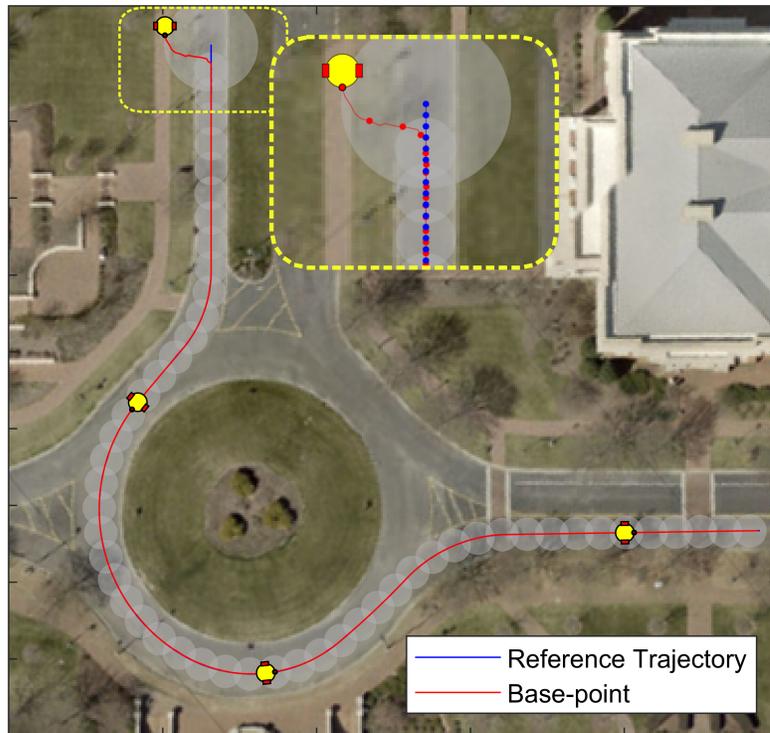


Figure 3.2: UNC Charlotte campus traffic circle: error bounds

3.2.3 Experiment 4: Lane Change

Fig. 3.3 is a simulation of the previously discussed lane change scenario. The reference trajectory provides the appropriate maneuver for a vehicle to merge from the left to right lane. The simulation is set up to have four different levels of measurement delay and noise with twenty tests being completed for each level. The tracking bounds are generated for the worst case possible for each level, and multiple tests are completed to demonstrate the tracking performance with random noise levels and set delay.

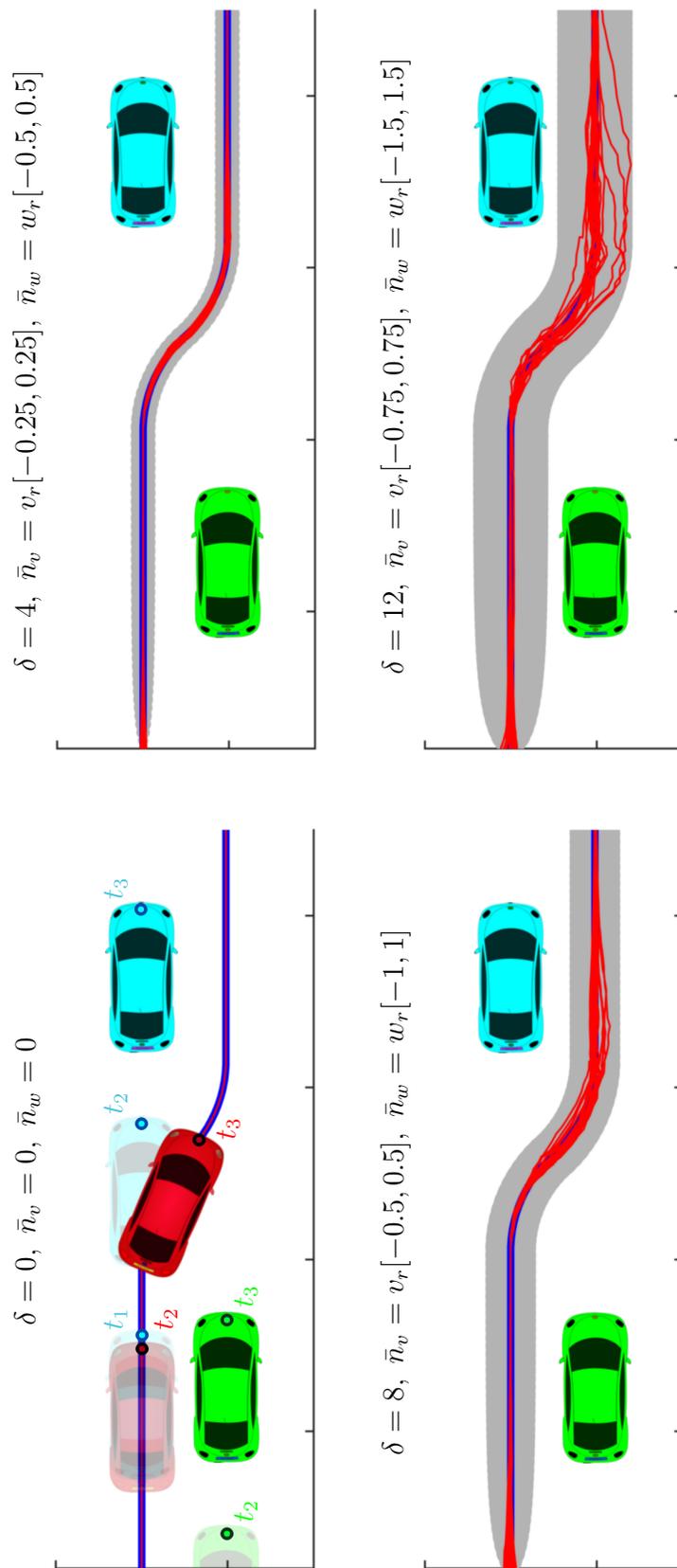


Figure 3.3: Lane change scenario: noise and delay

3.3 Hardware Results

This section presents the proposed state-feedback linearization controller implemented with a TurtleBot3 through the ROS framework. The TurtleBot3 uses an onboard ROS Noetic image installed on a Raspberry Pi to allow communication to a Host computer which uses a SSH Protocol. In the following experiments, the linear and angular velocities are published to the robot through the `cmd_vel` topic at 120 Hz. Eight OptiTrack cameras are used to provide positional tracking of the robot's rigid body pivot point at a frequency of 10 Hz. The pivot point is set to the robot's base-point at distance $\ell = .0476$ [m] from the centroid of the robot.

Host PC specifications: Ubuntu 20.04 OS, Intel i7-10700 CPU @ 2.90GHzx16, AMD Radeon rx 640 GPU, 16 GB DDR4 RAM.

TurtleBot3 specifications: $v_{\max} = 0.22$ [m/s], $w_{\max} = 2.84$ [rad/s]

3.3.1 The Charlotte Logo

The first hardware experiment uses the UNC Charlotte logo reference trajectory provided in the simulation results from Experiment 2. The robot's base-point trajectory is plotted in Fig. 3.1 with the simulated base-point trajectory to compare the results. The base-point position was initially placed where shown in the figure as $\mathbf{z}_b(0)$ with an initial heading angle of roughly $\theta(0) = \frac{\pi}{4}$ [rad].

3.3.2 Experiment 5: City Street

This experiment is shown in Fig. 3.4. The environment represents a city street layout with three 90° turns, one traffic circle, and one stop half way through the trajectory. The street layout is physically traversed by the Turtlebot3, as well as virtually in a trajectory planning program created to tune the bounded velocities in the controller and observe the real-time base-point trajectory. The robot traverses the first half of the trajectory with a upper bounded linear velocity of $v_{\max} = 0.20$ [m/s] until the stop at $[1.5, -0.6]$ where the linear velocity is increased to $v_{\max} = 0.22$ [m/s] for the

remainder of the trajectory. The control parameters are as follows: $v_r = 0.07$ [m/s], $k = 1$, $m = 1$.

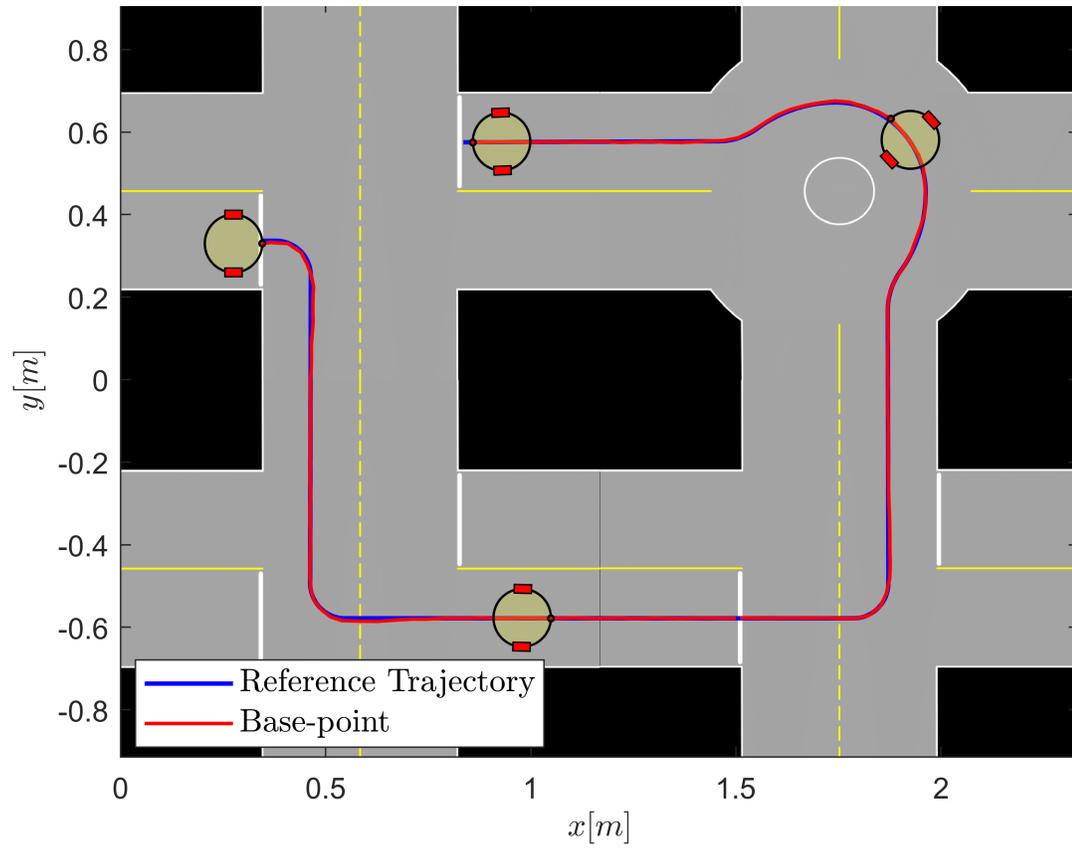


Figure 3.4: City street traversal

CHAPTER 4: INTERMITTENT-FEEDBACK CONTROLLER

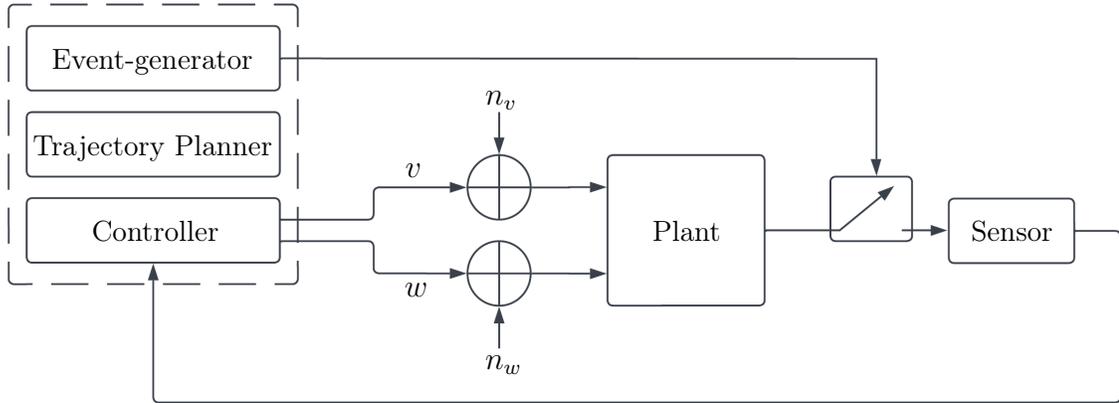


Figure 4.1: Intermittent feedback control system

We will now consider intermittent feedback and, as a result, design an appropriate communication policy. The communication policy must guarantee that the tracking error is eventually bounded by any desired upper bound ϵ . The time t_k denotes the k -th triggering time when the controller received the position $\mathbf{z}_b(t_k)$ and pose $\theta(t_k)$.

1

During the interval $[t_k, t_{k+1})$, the controller is open-loop and must use the estimated dynamics of our robot. Let the controller's estimate of \mathbf{z}_b at time t be $\hat{\mathbf{z}}_b(t) \triangleq \begin{bmatrix} \hat{x}_b(t) \\ \hat{y}_b(t) \end{bmatrix}$ and its estimation of the heading angle be $\hat{\theta}(t)$. The controller uses the estimated

¹In cases where $\theta(t_k)$ cannot be measured, the robot can compute this quantity by measuring both $\mathbf{z}_b(t_k)$ and $\mathbf{z}_c(t_k)$ and using (2.2) to obtain

$$\theta(t_k) = \tan^{-1} \left(\frac{y_b(t_k) - y_c(t_k)}{x_b(t_k) - x_c(t_k)} \right).$$

dynamics for all $t \in [t_k, t_{k+1})$,

$$\begin{bmatrix} \dot{\hat{x}}_b \\ \dot{\hat{y}}_b \\ \dot{\hat{\theta}} \end{bmatrix} = \begin{bmatrix} \cos \hat{\theta} & -l \sin \hat{\theta} \\ \sin \hat{\theta} & l \cos \hat{\theta} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}, \quad \begin{bmatrix} \hat{x}_b(t_k) \\ \hat{y}_b(t_k) \\ \hat{\theta}(t_k) \end{bmatrix} = \begin{bmatrix} x_b(t_k) \\ y_b(t_k) \\ \theta(t_k) \end{bmatrix}. \quad (4.1)$$

Recall from Chapter 3 that, under continuous state observation and noisy actuation, the dynamics of the robot were

$$\begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta & -l \sin \theta \\ \sin \theta & l \cos \theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v + n_v \\ w + n_w \end{bmatrix}. \quad (4.2)$$

We must now consider an estimated tracking error,

$$\hat{\mathbf{e}}(t) \triangleq \mathbf{z}_r(t) - \hat{\mathbf{z}}_b(t), \quad (4.3)$$

and the error between our estimated and actual base position,

$$\Delta(t) \triangleq \mathbf{z}_b(t) - \hat{\mathbf{z}}_b(t). \quad (4.4)$$

The actual tracking error at time t is defined to be

$$\mathbf{e}(t) \triangleq \mathbf{z}_r(t) - \mathbf{z}_b(t).$$

Therefore, due to (4.3) and (4.4) we obtain

$$\mathbf{e}(t) = \hat{\mathbf{e}}(t) - \Delta(t). \quad (4.5)$$

Remark 6 *From the above decomposition of $\mathbf{e}(t)$ in (4.5), one may notice that the*

part $\hat{\mathbf{e}}$ does not depend on the actuation noise and the part Δ does. As we will show soon, the dynamics of $\hat{\mathbf{e}}$ are controllable and thus the value of $\hat{\mathbf{e}}(t)$ can be controlled at time t by an appropriately chosen controller. We also notice that $\Delta(t_k) = 0$ at any triggering instance and thus its growth can be controlled by an appropriate choice of a triggering function. In the subsequent analysis, our objective is to minimize $\hat{\mathbf{e}}$ and make it 0 by designing an intelligent controller while also ensuring that (the upper bound) Δ does not grow large by designing an intelligent triggering function.

The proposed intermittent feedback controller is as follows:

$$\begin{bmatrix} v \\ w \end{bmatrix} = J(\hat{\theta})^{-1} \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \end{bmatrix} + \mathbf{u}, \quad (4.6)$$

with the control function to be defined.

Remark 7 *The proposed controller in (4.6) is similar in structure to the controllers (2.7) and (3.2) in Chapters 2 and 3, respectively. The major difference is, due to the lack of continuous communication, the intermittent feedback controller does not have access to the error $\mathbf{e}(t)$ for all time and consequently the $\mathcal{K}(\mathbf{e})$ part of (2.7) and (3.2) cannot be implemented. However, the controller can compute $\hat{\mathbf{e}}(t)$ and therefore, we use this information to design an intelligent controller \mathbf{u} as described in Lemma 7.*

Lemma 7 *For any $t_f > t_k$, the optimal control \mathbf{u} that ensures $\hat{\mathbf{e}}(t_f) = 0$ while minimizing the quadratic control cost $\int_{t_k}^{t_f} \|\mathbf{u}(s)\|^2 ds$ is*

$$\mathbf{u}(t) = \begin{cases} \frac{\hat{\mathbf{e}}(t_k)}{t_f - t_k}, & \text{if } t \leq t_f \\ 0, & \text{otherwise.} \end{cases} \quad (4.7)$$

Proof: Let us compute $\dot{\hat{e}}$ and use (4.1) and (4.6) to obtain

$$\dot{\hat{e}} = -\mathbf{u}. \quad (4.8)$$

Given the linear dynamics of (4.8) and the quadratic cost function $\int_{t_k}^{t_f} \|\mathbf{u}(s)\|^2 ds$, we may invoke the standard results from linear systems (c.f. Lemma 12 in Appendix A for details) to show that (4.7) is the optimal control. ■

The time t_f in (4.7) is a design parameter and its choice will be discussed later in this chapter (see Lemma 8).

4.1 Cases of Tracking Error

It will be shown in this section that the relationship of $\|e(t_k)\|$ with ϵ changes the choice of t_{k+1} . Case 1 represents when $\|e(t_k)\| > \epsilon$, where error will decrease for all $t \in [t_k, t_{k+1})$, while Case 2 is when $\|e(t_k)\| \leq \epsilon$, where error is bounded by ϵ .

4.1.1 Case 1: Converging Tracking Error

By substituting (4.7) in (4.8) and integrating, we obtain for all $t \leq t_f$ that

$$\hat{e}(t) = \frac{t_f - t}{t_f - t_k} \hat{e}(t_k) = \frac{t_f - t}{t_f - t_k} \mathbf{e}(t_k), \quad (4.9)$$

where the last equality is due to the fact that

$$\hat{e}(t_k) = \mathbf{z}_r(t_k) - \hat{\mathbf{z}}_b(t_k) = \mathbf{z}_r(t_k) - \mathbf{z}_b(t_k) = \mathbf{e}(t_k).$$

Given (4.9), we need to solve for $\Delta(t)$ to compute $\mathbf{e}(t)$. To that end, let us take the derivative of (4.4) and use (4.6) to obtain

$$\dot{\Delta}(t) = \dot{\mathbf{z}}_b(t) - \dot{\mathbf{z}}_r(t) + \mathbf{u}. \quad (4.10)$$

Substituting the noisy dynamics (4.2) along with the chosen controller (4.6) into

(4.10) and integrating over $[t_k, t]$, we obtain

$$\Delta(t) = \int_{t_k}^t \mathbf{d}(s)ds - \int_{t_k}^t A(\dot{\mathbf{z}}_r(s) - \mathbf{u})ds, \quad (4.11)$$

where A and \mathbf{d} are defined in (3.10) and we have used the fact that $\Delta(t_k) = 0$.

We can now obtain the tracking performance by substituting (4.9) and (4.11) into (4.5),

$$\mathbf{e}(t) = \frac{t_f - t}{t_f - t_k} \mathbf{e}(t_k) - \int_{t_k}^t \mathbf{d}(s)ds + \int_{t_k}^t A(\dot{\mathbf{z}}_r(s) - \mathbf{u})\mathbf{d}(s). \quad (4.12)$$

Since t_f is a design parameter, we will choose t_f big enough such that $t_{k+1} \leq t_f$ whenever $\|\mathbf{e}(t_k)\| > \epsilon$. We will discuss soon how to conduct the design such that $t_{k+1} \leq t_f$ is ensured. Therefore, for any time $t \in [t_k, t_{k+1})$, the tracking error norm is upper bounded by

$$\begin{aligned} \|\mathbf{e}(t)\| &\leq \frac{t_f - t}{t_f - t_k} \|\mathbf{e}(t_k)\| + \int_{t_k}^t \|\mathbf{d}(s)\|ds + \int_{t_k}^t \|A(\dot{\mathbf{z}}_r(s) - \mathbf{u})\|ds, \\ &\stackrel{\dagger_1}{\leq} \frac{t_f - t}{t_f - t_k} \|\mathbf{e}(t_k)\| + \eta_1 \int_{t_k}^t ds + \int_{t_k}^t 2 \left| \sin\left(\frac{\theta_e(s)}{2}\right) \right| \sqrt{(\dot{\mathbf{z}}_r - \mathbf{u})^\top (\dot{\mathbf{z}}_r - \mathbf{u})} ds \\ &\stackrel{\dagger_2}{\leq} \frac{t_f - t}{t_f - t_k} \|\mathbf{e}(t_k)\| + \eta_1 \int_{t_k}^t ds + \left(v_r + \frac{\|\mathbf{e}(t_k)\|}{t_f - t_k} \right) \int_{t_k}^t |\theta_e(s)| ds \\ &\leq \frac{t_f - t}{t_f - t_k} \|\mathbf{e}(t_k)\| + \eta_1 \int_{t_k}^t ds + \left(v_r + \frac{\|\mathbf{e}(t_k)\|}{t_f - t_k} \right) \bar{n}_w \int_{t_k}^t (s - t_k) ds \\ &= \frac{t_f - t}{t_f - t_k} \|\mathbf{e}(t_k)\| + (t - t_k)\eta_1 + \left(v_r + \frac{\|\mathbf{e}(t_k)\|}{t_f - t_k} \right) \frac{\bar{n}_w}{2} (t - t_k)^2, \end{aligned} \quad (4.13)$$

where

$$\eta_1 = \sqrt{\bar{n}_v^2 + \ell^2 \bar{n}_w^2}.$$

The inequality \dagger_1 follows directly from the triangle inequality $\|x + y\| \leq \|x\| + \|y\|$ and \dagger_2 by Lagrange's theorem $\sin(x) \leq |x|$. We can rearrange (4.13) in the form of a

polynomial with respect to $(t - t_k)$ as

$$\|\mathbf{e}(t)\| \leq \|\mathbf{e}(t_k)\| - b_1(t - t_k) + b_2(t - t_k)^2 \triangleq e_{\text{upper}}(t), \quad (4.14)$$

where

$$b_1 = \left(\frac{\|\mathbf{e}(t_k)\|}{t_f - t_k} - \eta_1 \right), \quad b_2 = \left(v_r + \frac{\|\mathbf{e}(t_k)\|}{t_f - t_k} \right) \frac{\bar{n}_w}{2}. \quad (4.15)$$

To reduce the tracking error, it is sufficient to reduce the upper bound $e_{\text{upper}}(t)$.

Remark 8 Given $e_{\text{upper}}(t)$ depends on t as a quadratic polynomial, it is efficient to find conditions to determine whether $e_{\text{upper}}(t)$ decreases with time. The following lemma prescribes such a condition on t_f .

Lemma 8 Pick any $p_k \in [0, 1)$ such that

$$t_f \triangleq t_k + p_k \frac{\|\mathbf{e}(t_k)\|}{\eta_1} \geq t_k + \frac{b_1}{2b_2}. \quad (4.16)$$

Then, $e_{\text{upper}}(t)$ decreases in the interval $[t_k, t_k + \frac{b_1}{2b_2}]$.

Proof: A necessary condition to ensure $e_{\text{upper}}(t)$ is decreasing for $t > t_k$ is $b_1 > 0$.

Thus, from the expression of b_1 in (4.15), we obtain the condition

$$t_f < t_k + \frac{\|\mathbf{e}(t_k)\|}{\eta_1}. \quad (4.17)$$

We can observe that any t_f satisfying (4.17) can be expressed as

$$t_f = t_k + p_k \frac{\|\mathbf{e}(t_k)\|}{\eta_1},$$

for some $p_k \in [0, 1)$.

Since $t_f > t_k + \frac{b_1}{2b_2}$ as per the lemma condition, (4.13) holds true for all $t \in [t_k, t_k + \frac{b_1}{2b_2}]$. By taking the derivative of $e_{\text{upper}}(t)$ we obtain that $\dot{e}_{\text{upper}}(t) \leq 0$ only for $t \in [t_k, t_k + \frac{b_1}{2b_2}]$. ■

Corollary 2 *Let t_f be chosen according to Lemma 8 for some $p_k \in [0, 1)$, then the optimal trigger time $t_{k+1} > t_k$ that minimizes $e_{\text{upper}}(t)$ is*

$$t_{k+1} = t_k + \frac{b_1}{2b_2}. \quad (4.18)$$

Proof: From the quadratic nature of $e_{\text{upper}}(t)$, we obtain that $e_{\text{upper}}(t)$ is minimum at $t_k + \frac{b_1}{2b_2}$ and thus, the triggering must be done at that time. ■

In this work we use (4.18) to generate the triggering instances. As shown in the above analysis, this choice ensures that the upper bound on the error is minimized. Therefore, at the next trigger time t_{k+1} , we have

$$\|\mathbf{e}(t_{k+1})\| \leq e_{\text{upper}}(t_{k+1}) = \|\mathbf{e}(t_k)\| - \frac{b_1^2}{4b_2}. \quad (4.19)$$

Clearly the error norm decreases with each triggering and there exists an integer N such that $\|\mathbf{e}(t_N)\| \leq \epsilon$ for a user defined tolerance parameter ϵ . The decrease in actual error norm $\|\mathbf{e}(t)\|$ from time t_k to t_{k+1} is at least

$$\frac{b_1^2}{4b_2} = \frac{\eta_1^2(1-p_k)^2}{2\bar{n}_w(v_r p_k^2 + \eta_1 p_k)}. \quad (4.20)$$

This quantity depends on the choice of p_k and it increases as p_k decreases. While p_k can be chosen as desired by the user, for the subsequent analysis we will assume that $p_k \leq \bar{p}$ for all k . The triggering instances depend on p_k . In fact,

$$t_{k+1} - t_k = \frac{b_1}{2b_2} = \frac{\frac{\|\mathbf{e}(t_k)\|}{t_f - t_k} - \eta_1}{(v_r + \frac{\|\mathbf{e}(t_k)\|}{t_f - t_k})\bar{n}_w} = \frac{\eta_1(1-p_k)}{(v_r p_k + \eta_1)\bar{n}_w}.$$

Lemma 9 Let $N = \left\lceil \frac{2\bar{n}_w(v_r\bar{p}^2 + \eta_1\bar{p})(\|e(t_0)\| - \epsilon)}{\eta_1^2(1 - \bar{p})^2} \right\rceil$, then

$$\|e(t)\| \leq \epsilon,$$

for all $t \geq t_N$, where t_k 's are found from (4.18) for all $k = 0, \dots, N - 1$ with $t_0 = 0$.

Proof: From (4.19) we notice that $\|e(t_k)\|$ decreases with k . Furthermore, (4.19) along with (4.20) provide us with

$$\|e(t_k)\| \leq \|e(t_0)\| - \sum_{i=0}^{k-1} \frac{\eta_1^2(1 - p_i)^2}{2\bar{n}_w(v_r p_i^2 + \eta_1 p_i)}. \quad (4.21)$$

Given that $p_k \leq \bar{p}$ for all k , we can observe from 4.21 that

$$\|e(t_k)\| \leq \|e(t_0)\| - k \frac{\eta_1^2(1 - \bar{p})^2}{2\bar{n}_w(v_r \bar{p}^2 + \eta_1 \bar{p})}.$$

Therefore, for $k = N$ where N is given in the lemma statement, we obtain

$$\|e(t_N)\| \leq \|e(t_0)\| - N \frac{\eta_1^2(1 - \bar{p})^2}{2\bar{n}_w(v_r \bar{p}^2 + \eta_1 \bar{p})} \leq \epsilon.$$

■

4.1.2 Case 2: Satisfied Tracking Error

When $\|e(t_k)\| \leq \epsilon$ we must consider $t_{k+1} > t_f$. In Case 1, we designed t_f such that $t \leq t_f \forall t \in [t_k, t_{k+1})$. For this case we may still choose t_f from (4.16) but must also consider that $t_{k+1} > t_f$ since t_f can be very close to t_k .

Lemma 10 When $\|\mathbf{e}(t_k)\| \leq \epsilon$, t_{k+1} is maximized by the following:

$$t_{k+1} = \begin{cases} t_k + \frac{b_1 + \sqrt{b_1^2 - 4b_2(\|\mathbf{e}(t_k)\| - \epsilon)}}{2b_2}, & \text{if } t_{k+1} \leq t_f \\ t_k + \max\left(\frac{-\eta_1 + \sqrt{\eta_1^2 + 4b_2\epsilon}}{2b_2}, p_k \frac{\|\mathbf{e}(t_k)\|}{\eta_1}\right) & \text{otherwise.} \end{cases} \quad (4.22)$$

Proof: We must consider $t_{k+1} > t_f$, where t_f is chosen from (4.16). When $t \leq t_f$, we have found t_{k+1} from (4.18). When $t > t_f$ the control from (4.7) is $\mathbf{u}(t) = 0$, resulting in the following new bound of $\|\mathbf{e}(t)\|$ in replace of (4.13),

$$\|\mathbf{e}(t)\| \leq (t - t_k)\eta_1 + \left(v_r + \frac{\eta_1}{p_k}\right) \frac{\bar{n}_w}{2} (t - t_k)^2. \quad (4.23)$$

Substituting $t = t_{k+1}$ into (4.23) when the r.h.s. of (4.23) is ϵ , we can solve for t_{k+1} :

$$t_{k+1} = t_k + \frac{-\eta_1 + \sqrt{\eta_1^2 + 4b_2\epsilon}}{2b_2}. \quad (4.24)$$

■

4.2 Choice of p_k

In this section we discuss the choice of p_k and the trade-offs associated with that choice. Notice that the choice of p_k directly affects the triggering instances; hence, the total number of triggers before the error becomes small enough (i.e., less than ϵ). It also affects the magnitude of the control input \mathbf{u} .

Lemma 11 The linear and angular velocity are dependent on the choice of p_k and are bounded by the following,

$$|v(t)| \leq v_r + \|\mathbf{u}(t)\| = v_r + \frac{\eta_1}{\bar{p}}, \quad (4.25a)$$

$$|w(t)| \leq w_r + \frac{\|\mathbf{u}(t)\|}{\ell} = w_r + \frac{\eta_1}{\ell \bar{p}}. \quad (4.25b)$$

Proof: Recall that when $t \leq t_f$,

$$\mathbf{u}(t) = \frac{\mathbf{e}(t_k)}{t_f - t_k} = \frac{\eta_1}{p_k} \frac{\mathbf{e}(t_k)}{\|\mathbf{e}(t_k)\|}$$

for all $t \in [t_k, t_{k+1})$. Therefore, $\|\mathbf{u}(t)\| = \eta_1/p_k$.

From (4.6), we obtain

$$\begin{aligned} |v(t)| &\leq v_r + \|\mathbf{u}(t)\| = v_r + \frac{\eta_1}{p_k}, \\ |w(t)| &\leq w_r + \frac{\|\mathbf{u}(t)\|}{\ell} = w_r + \frac{\eta_1}{\ell p_k}. \end{aligned}$$

Similarly, when $t > t_f$, $\mathbf{u}(t) = 0$ and we obtain,

$$\begin{aligned} |v(t)| &\leq v_r, \\ |w(t)| &\leq w_r. \end{aligned}$$

Therefore, the actuated velocity when $t \leq t_f$ depends on p_k . ■

By this point, the intermittent feedback controller has a communication policy from (4.18) when $\|\mathbf{e}(t_k)\| > \epsilon$ and (4.22) when $\|\mathbf{e}(t_k)\| \leq \epsilon$. The policy guarantees that the tracking performance converges to the desired tolerance ϵ . The simulation results for this chapter demonstrate the decrease in required communications after the time t_N when the tracking error becomes $\mathbf{e} \leq \epsilon$. This is a direct result of the policy allowing the tracking bound (4.14) to increase to ϵ after time t_N .

4.3 Simulation Results

The following simulations implement the proposed intermittent feedback controller (4.6). The figures show the locations of where each triggering instance t_k occurs by magenta colored dots. The actuation noise is generated based on a percentage of the reference linear and angular velocity which will be mentioned for each figure.

4.3.1 Experiment 6: Intermittent Feedback Control

Fig. 4.2 demonstrates the intermittent feedback controller with the circular reference trajectory in Fig. 2.3. The blue and magenta colored dots visualize the corresponding reference trajectory and base-point positions, respectively, for each trigger. The grey boundary shows the user-desired tracking error upper-bound, $\epsilon = 0.2$ [m]. The actuation noises are bounded: $\bar{n}_v = 0.1v_r$, $\bar{n}_w = 0.1w_r$, where $v_r = 1$ [m/s], $w_r = 3.33$ [rad/s]. It is shown that $N = 9$ and after the trigger time t_9 the tracking error remains bounded by ϵ . The error performance of this simulation is provided in Fig. 4.3. The figure demonstrates time-varying triggering instances which are providing a guarantee that the tracking error will decrease. After t_N , the time between each trigger is increased since $\|e(t)\| \leq \epsilon$, $\forall t \geq t_N$. It can be observed that because the communication policy is determined for the worst case possible, the tracking performance remains much less than ϵ . The next lane change scenario demonstrates that the worst case is possible, however, the tracking error remains bounded.

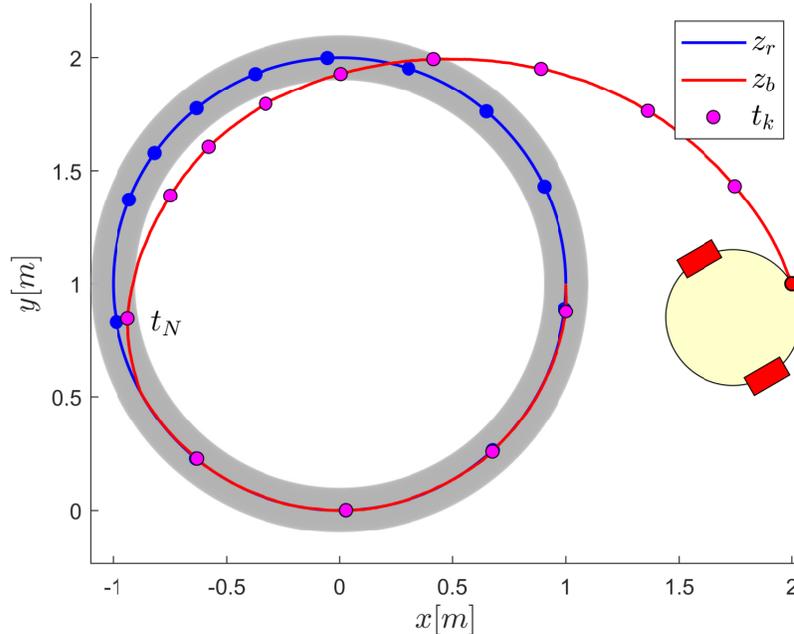


Figure 4.2: Intermittent feedback performance with a circular reference trajectory

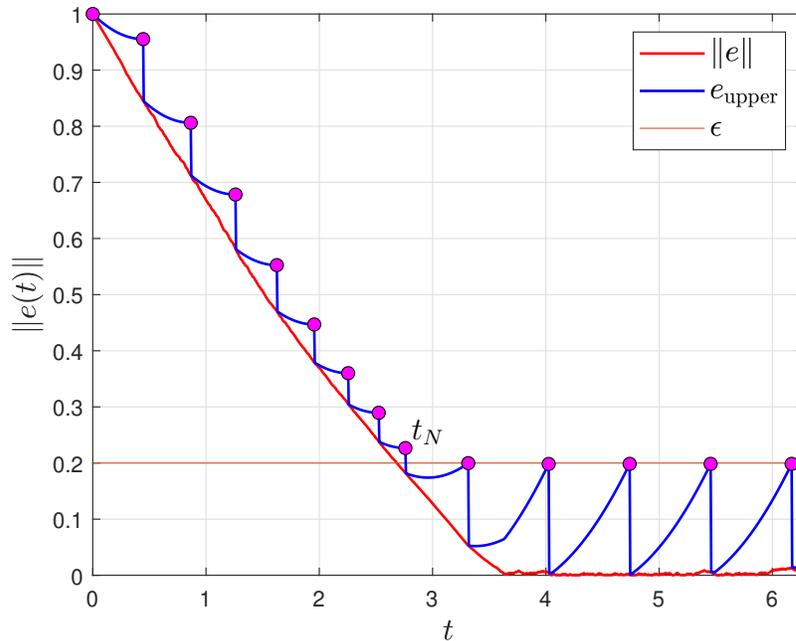


Figure 4.3: Performance characteristics of (4.14) in Fig. 4.2

4.3.2 Experiment 7: Lane Change with Intermittent Feedback

Fig. 4.4 showcases the intermittent control in the lane change scenario. There are four levels of actuation noise with twenty trials in each. Both actuation noise and ϵ are increased for each level. The first level demonstrates that with low amounts of noise fewer communications are required. The even-triggered communication policy provides the requirement of more communications in each increasing level as a direct result of the added noise. In the highest noise level it can be seen that the base-point trajectory will often diverge close to ϵ during the time $t \in [t_k, t_{k+1})$ but right after the time $t = t_{k+1}$ the trajectory will converge.

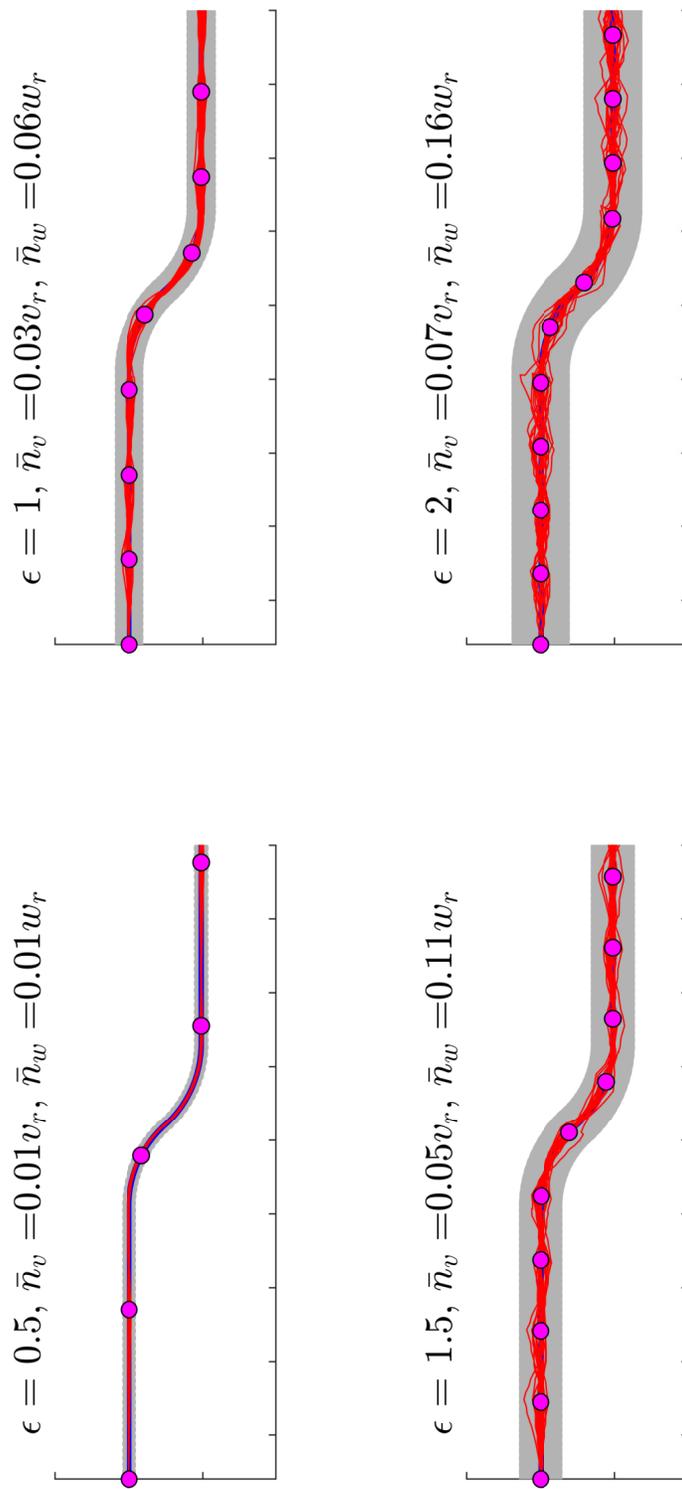


Figure 4.4: Lane change scenario: intermittent feedback

CHAPTER 5: CONCLUSION

The proposed state-feedback controller (2.7) provides exponentially accurate trajectory tracking when $\|\mathbf{e}\| \leq m$ without the expense of high velocities by incorporating the saturated control when $\|\mathbf{e}\| > m$. One can see in the simulation results from Fig. 2.3 that the controller performs accurately even when the reference trajectory demands higher angular velocities. The velocities were upper bounded by (2.15) and can be tuned with (2.18) by adjusting the control parameter k and m . Fig. 2.4 provides a visualization to the reader for the guarantee that the velocities will not exceed a set user-defined threshold.

The dynamics were introduced to linear and angular actuation noise while the heading angle estimation incorporates delayed measurements. The estimated heading angle was then calculated without the knowledge of the noise in the system with a defined delay period. The analysis shows that the estimated heading angle error $|\theta_e(t)|$ is upper bounded by the delay period and maximum angular noise. Through a Lyapunov approach the tracking error $\|\mathbf{e}\|$ was shown to be upper bounded $\forall t$ under the constraint $|\theta_e(t)| \leq \pi/2$. The bounds guarantee that the tracking error is constrained by (2.10).

First in the simulation results, the controller (2.7) is tested with a circular reference trajectory as shown in Figs. 2.3, 2.4. These figures show that the controller provides exponential tracking accuracy with the linear control without the cost of unconstrained velocities by incorporating the saturated control. Furthermore, the results demonstrate the ease of tuning the controller with the derived relationship (2.18). Next, the controller with actuation noise and communication delay (3.2) was demonstrated with a complicated reference trajectory (the UNC Charlotte logo) in

Fig. 3.1, a realistic traffic circle scenario in Fig. 3.2, and a lane change scenario in Fig. 3.3. The UNC Charlotte logo scenario demonstrates the effectiveness with a trajectory that has sharp corners and high angular velocities. In the traffic circle scenario, the tracking bounds (3.8) are demonstrated to provide safety-critical trajectory tracking. The final lane change scenario demonstrates the reliability of the controller with multiple tests at varying actuation noise and measurement delay. As observed, none of the trails crossed the guaranteed tracking bound. Videos for both the simulation and hardware results can be found in Ref. [50].

In the hardware experimentation results, the controller (2.7) is implemented into hardware through ROS with the TurtleBot3 Burger. The tracking performance was shown to be excellent even with a complicated reference trajectory. The results are compared directly to the noiseless simulation expectation in Fig. 3.1. The city street layout in Fig. 3.4 demonstrated the true tracking performance and shows that the vehicle is able to remain in its appropriate lane.

The trajectory tracking intermittent state-feedback linearization controller provides the optimal control function to minimize energy. The tracking performance is guaranteed to be bounded by any desired ϵ after time t_N . The control policy was shown to provide a direct relationship to the actuation noise and control parameters. The triggering time is designed to be tuned by p_k which directly determines the magnitude of the linear and angular velocities as shown in (4.25) versus the communication requirement in (4.18), (4.22). The velocities are shown to be saturated and bounded by the disturbance, reference velocity, and control parameter p_k . The intermittent feedback control is simulated in Figs. 4.2 and 4.3 with a circular reference trajectory. The figure shows the time-varying event-based triggering and shows that the true performance will often be below the desired tolerance. The controller was also implemented with the lane change scenario in Fig. 4.4. One can observe that the performance of the intermittent feedback controller can be comparable to the continuous

feedback controller but with limited communication.

The advantages and trade-offs have been made clear for when deciding between implementation of the proposed continuous and intermittent feedback controller. When communication costs are not an issue, the continuous state-feedback controller could be chosen to provide exponential stability. The tracking performance was guaranteed with direct relationship to communication delay and actuation noise. However, when limited communication is desired, the intermittent feedback control has a major advantage by only requiring sensor information at specific event generated by the desired performance, actuation noises, and system parameters.

5.1 Discussion

This thesis provides a robust trajectory tracking approach that can be extended with numerous autonomous nonholonomic robots. Time constrained maneuvers are necessary for applications such as multi-robot surveillance and mapping, similar to the underwater vehicle control proposed in Ref. [51]. Game theory for robotic systems require time dependent and optimal maneuvers with as limited information as possible. These requirements are directly answered by the proposed optimal intermittent feedback control. Military applications, such as guided missiles, need a reliable control to guarantee accuracy. The proposed controller could be expanded into this military application with its highly tunable time-varying tracking guarantees, similar to what has been done in Ref. [52]. These are just a few examples among the countless other works that can incorporate this control synthesis.

Many real-life factors and constraints are considered in this thesis, but it could be further enhanced. The intermittent feedback control analysis considers actuation noise but can also be further expanded by considering communication delay. The continuous feedback analysis reveals that, due to the communication delay, the heading angle must be estimated. It can be assumed that communication delay would show a similar effect on the intermittent feedback control but, because the exact base-point

position is not provided until each trigger, the analysis must then consider more factors. Furthermore, one could also expand the analysis by incorporating observation noise and analyze the differences of the tracking bounds with observation versus actuation noise. Kalman filtering is a sophisticated and widely used state estimation method that works hand-in-hand with a state-feedback controller. The effects of actuation noise on the tracking bounds could be reduced by incorporating a Kalman filter which will further decrease the tracking tolerance. While the bounds and the analyses in the thesis consider the worst case scenarios, one could possibly consider a probabilistic scenario where probabilistic guarantees on the bounds can be provided. Such bounds are generally less conservative than the ones derived in this thesis under worst case scenario. These probabilistic bounds are of particular interest for several less safety-critical scenarios (e.g., surveillance of an empty land). Under these scenarios, it would be possible to account for unbounded noise and time-varying random communication delay.

This thesis was developed to provide a strong and conclusive analysis with the intention to motivate and influence future extensions and implementations. Extensions of this thesis along the lines of above-mentioned items will further contribute to the controls and robotics community and push the frontiers of trajectory tracking research.

CHAPTER 6: SOFTWARE AND HARDWARE IMPLEMENTATION DETAILS

This chapter provides the necessary instruction for readers to replicate the simulation and hardware experiments performed in this thesis. The required code for the MATLAB simulations and ROS hardware experiments can be found in Ref. [50] or the following repository:

<https://github.com/frankllawless/Trajectory-Tracking-Control.git>.

Software

- MATLAB R2022b
- ROS Noetic
- Ubuntu 20.04
- Python 3.7.3

Python Libraries

- Numpy
- Pygame.camera
- Matplotlib.pyplot
- Pygame
- Rospy
- Math

6.1 Simulations

Experiment 1 (see Figs. 2.3,2.4) provides the state-feedback controller (2.7) without any actuation noise or communication noise. The reference trajectory for each experiment are provided in their respective experiment folder. Experiment 2 provides a comparison between the simulation results and the hardware results saved in `hardware_results.csv`.

Experiments 2-4 (see Figs. 3.1,3.2,3.3) provide the controller (3.2) with an estimated heading angle and noisy actuation. Experiment 4 provides the lane change scenario where the user may adjust the control parameters, number of trails, actuation noise, and communication delay for each level to observe their direct effect to the tracking error upper-bounds. Experiments 6-7 (see Figs. 4.2,4.4) provide the intermittent feedback controller (4.6) with the estimated dynamics and heading angle.

In these experiments the user may adjust the desired tracking performance ϵ and the control parameter p to adjust the convergence rate, communication frequency, and velocities. The actuation noises can be adjusted by changing the parameters *angular_noise*, *linear_noise* $\in [0.1, 1]$. The user may also adjust the initial state conditions, such as $\mathbf{z}_c(0)$ and $\theta(0)$ for all of the provided experiments.¹

6.2 Hardware Experiments

The following section provides the appropriate steps to perform the hardware experiments with the developed trajectory planning Pygame program. The provided code is formatted to be created as ROS catkin packages. The executable files provided in the `trajectory_tracking` folder are completed open-loop (i.e., they do not use any positional feedback and the dynamics are approximated). The files provided in `OptiTrack_trajectory_tracking` are specifically designed to use OptiTrack motion capture cameras as base-point and centroid positional feedback. To use OptiTrack cameras with the provided program, the user must place enough physical tracking markers on the robot (see Fig. 6.1) in-order to create a rigid body with pivot points placed specifically at the centroid and desired base-point position on the robot (multiple rigid bodies may be created with the same unique markers). For further information please refer to the following OptiTrack quick start guide: <https://docs.optitrack.com/quick-start-guides>

¹Note that Eurlers-method is used for all simulation experiments to approximate the dynamics with a sample time of h which should be set to a small value for higher accuracy i.e., $h = 0.02$.

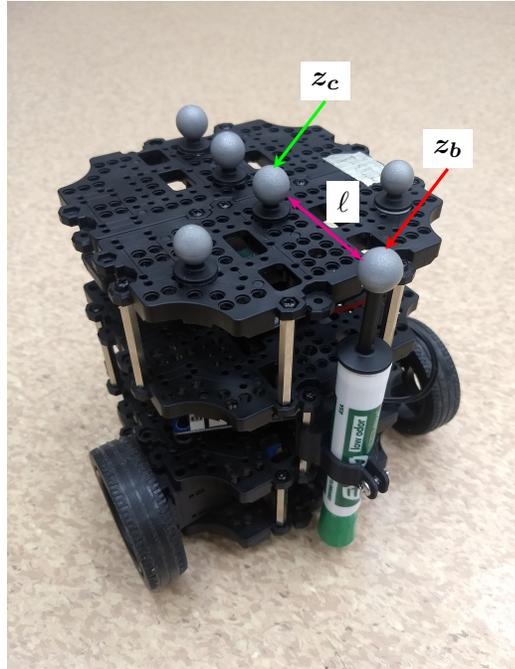


Figure 6.1: Example of appropriate marker locations

6.2.1 TurtleBot3 Startup

ROBOTIS provides the TurtleBot3 with an e-Manual that contains all of the necessary steps to install a ROS Noetic image into the on-board Raspberry Pi:

<https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/>

Make sure to connect the TurtleBot3 to the same WiFi network as the Host PC by changing the access point in `50-cloud-init.yaml` located in the `netplan` directory. The IP of both the Host machine and Raspberry Pi should be exported in `.bashrc` on the Raspberry Pi.

Enter the following into the Host terminal to check connection to Raspberry PI:

```
$ ping {IP_ADDRESS_OF_RASPBERRY_PI}
```

If the connection is successful proceed with the following commands to bring up the TurtleBot3:

```
$ roscore
```

```
$ ssh ubuntu@{IP_ADDRESS_OF_RASPBERRY_PI}
```

By default – Username: ubuntu, Password: turtlebot.

```
$ export TURTLEBOT3_MODEL=burger
$ roslaunch turtlebot3_bringup turtlebot3_robot.launch
```

If bring up is successful, the topic `cmd_vel` is shown after running the following command:

```
$ rostopic list
```

6.2.2 Running an Experiment

For each experiment the velocity publisher is initialized as a twist class to the parameter `pub`. By default this publishes to the `cmd_vel` topic. Double check that the publisher matches the TurtleBot3's topic. To run the open-loop experiments, execute the appropriate `.py` file in the terminal with `roslaunch`. When executed, the reference trajectory will display and the experiment will begin after the user exits out of the figure. After exiting, the published linear and angular velocities will be visible in the terminal, and the TurtleBot3 should now be performing the experiment. Once the experiment is completed, the ideal simulation environment is displayed with the linear and angular velocities. The user may use the displayed simulation to compare the ideal environment versus the actual hardware trajectory.

In all experiments that use the OptiTrack cameras, the parameter `pos` is defined as the currently subscribed pivot point position. The pose subscriber should be set to the rigid body base-point location. The parameter `l` should be set to the physical distance between the base-point and centroid position on the robot.

6.2.3 Trajectory Planning Program

The user may provide hand-drawn reference trajectories to the TurtleBot3 by executing the `hand-drawn.py` file in both the open-loop and closed-loop experiments. The user may then draw out any trajectory by holding left-click in the Pygame window. After the trajectory is drawn, press the space key to then adjust the maximum

velocities allowed for the experiments by using the up and down arrows keys. Once the user presses the space key one more time, the experiment will begin and the trajectory planning Pygame program will display the live position and orientation of the robot in the closed-loop experiments or the simulated position and orientation in the open-loop experiments. See Fig. 6.2 for the expected visualization.

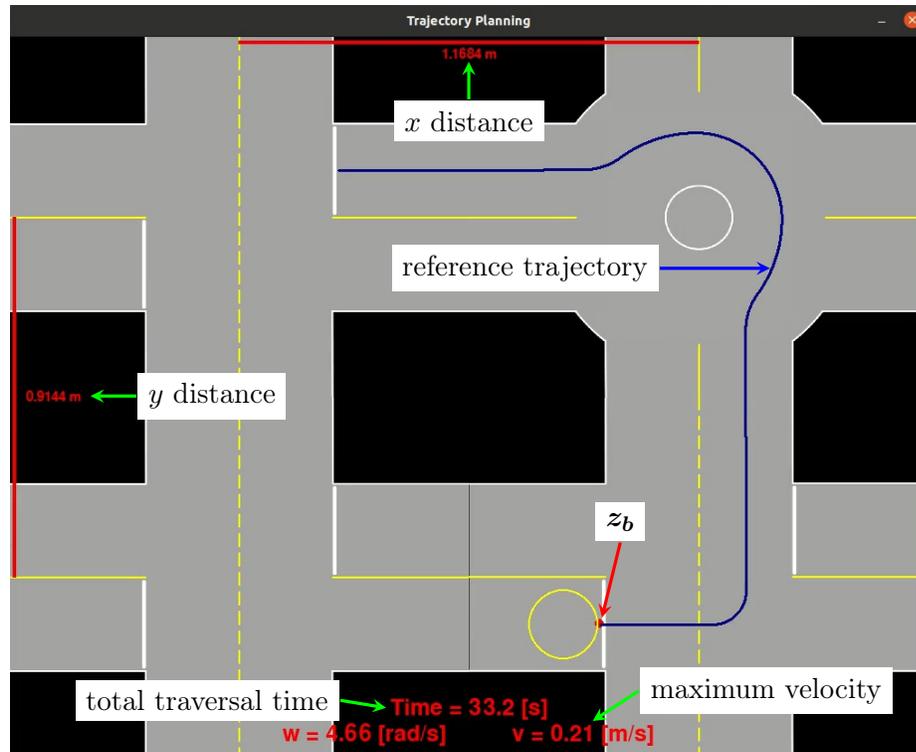


Figure 6.2: Trajectory planning features

REFERENCES

- [1] A. G. Ozkil, Z. Fan, S. Dawids, H. Aanes, J. K. Kristensen, and K. H. Christensen, "Service robots for hospitals: A case study of transportation tasks in a hospital," in *International Conference on Automation and Logistics*, pp. 289–294, 2009.
- [2] L. G. Weiss, "Autonomous robots in the fog of war," *IEEE Spectrum*, vol. 48, no. 8, pp. 30–57, 2011.
- [3] R. R. Murphy, V. B. M. Gandudi, and J. Adams, "Applications of robots for COVID-19 response," *CoRR*, 2020.
- [4] E. Masehian and N. Mohamadnejad, "Path planning of nonholonomic flying robots using a new virtual obstacle method," in *3rd RSI international conference on robotics and mechatronics*, pp. 612–617, IEEE, 2015.
- [5] G. Indiveri, A. Nuchter, and K. Lingemann, "High speed differential drive mobile robot path following control with bounded wheel speed commands," in *International Conference on Robotics and Automation*, pp. 2202–2207, IEEE, 2007.
- [6] S. Van Koevering, Y. Lyu, W. Luo, and J. Dolan, "Provable probabilistic safety and feasibility-assured control for autonomous vehicles using exponential control barrier functions," *arXiv preprint*, 2022.
- [7] S. He, J. Zeng, B. Zhang, and K. Sreenath, "Rule-based safety-critical control design using control barrier functions with application to autonomous lane change," in *American Control Conference*, pp. 178–185, IEEE, 2021.
- [8] K. Shojaei, A. M. Shahri, A. Tarakameh, and B. Tabibian, "Adaptive trajectory tracking control of a differential drive wheeled mobile robot," *Robotica*, vol. 29, no. 3, pp. 391–402, 2011.
- [9] J. Wu, G. Xu, and Z. Yin, "Robust adaptive control for a nonholonomic mobile robot with unknown parameters," *Journal of Control Theory and Applications*, vol. 7, no. 2, pp. 212–218, 2009.
- [10] E. Canigur and M. Ozkan, "Model reference adaptive control of a nonholonomic wheeled mobile robot for trajectory tracking," in *International Symposium on Innovations in Intelligent Systems and Applications*, pp. 1–5, IEEE, 2012.
- [11] P. Guo, Z. Liang, X. Wang, and M. Zheng, "Adaptive trajectory tracking of wheeled mobile robot based on fixed-time convergence with uncalibrated camera parameters," *ISA transactions*, vol. 99, pp. 1–8, 2020.
- [12] S. Xiao and S. Wuxi, "Adaptive trajectory tracking control of wheeled mobile robot," in *Chinese Control And Decision Conference*, pp. 5161–5165, IEEE, 2019.

- [13] D. Huang, J. Zhai, W. Ai, and S. Fei, “Disturbance observer-based robust control for trajectory tracking of wheeled mobile robots,” *Neurocomputing*, vol. 198, pp. 74–79, 2016.
- [14] B. Moudoud, H. Aissaoui, and M. Diany, “Finite-time adaptive trajectory tracking control based on sliding mode for wheeled mobile robot,” in *18th International Multi-Conference on Systems, Signals & Devices*, pp. 1148–1153, IEEE, 2021.
- [15] J.-y. Zhai and Z.-b. Song, “Adaptive sliding mode trajectory tracking control for wheeled mobile robots,” *International Journal of Control*, vol. 92, no. 10, pp. 2255–2262, 2019.
- [16] N. Wang, H. R. Karimi, H. Li, and S.-F. Su, “Accurate trajectory tracking of disturbed surface vehicles: A finite-time control approach,” *IEEE Transactions on Mechatronics*, vol. 24, no. 3, pp. 1064–1074, 2019.
- [17] N. K. Goswami and P. K. Padhy, “Sliding mode controller design for trajectory tracking of a non-holonomic mobile robot with disturbance,” *Computers & Electrical Engineering*, vol. 72, pp. 307–323, 2018.
- [18] R. Solea, A. Filipescu, and U. Nunes, “Sliding-mode control for trajectory-tracking of a wheeled mobile robot in presence of uncertainties,” in *7th Asian Control Conference*, pp. 1701–1706, IEEE, 2009.
- [19] L. Kunpeng, W. Xuewen, Y. Mingxin, L. Xiaohu, and W. Sunan, “Adaptive sliding mode trajectory tracking control of mobile robot with parameter uncertainties,” in *International Symposium on Computational Intelligence in Robotics and Automation*, pp. 148–152, IEEE, 2009.
- [20] V. Alakshendra and S. S. Chiddarwar, “Adaptive robust control of mecanum-wheeled mobile robot with uncertainties,” *Nonlinear Dynamics*, vol. 87, no. 4, pp. 2147–2169, 2017.
- [21] B. Moudoud, H. Aissaoui, and M. Diany, “Robust trajectory tracking control based on sliding mode of differential driving four-wheeled mobile robot,” in *6th International Conference on Optimization and Applications*, pp. 1–5, IEEE, 2020.
- [22] H. Yu, N. Sheng, and Z. Ai, “Sliding mode control for trajectory tracking of mobile robots,” in *40th Chinese Control Conference*, pp. 13–17, IEEE, 2021.
- [23] M. Galicki and M. Banaszkiwicz, “Optimal trajectory tracking control of omnidirectional mobile robots,” in *12th International Workshop on Robot Motion and Control*, pp. 137–142, IEEE, 2019.
- [24] J. Zhang, F. Wang, and G. Wen, “Sliding mode observer-based control of teleoperation system with uncertain dynamics and kinematics,” 2021.

- [25] Y. Yang, X. Yan, K. Sirlantzis, and G. Howells, "Application of sliding mode trajectory tracking control design for two-wheeled mobile robots," in *Conference on Adaptive Hardware and Systems*, pp. 109–114, IEEE, 2019.
- [26] S. Blažič, "A novel trajectory-tracking control law for wheeled mobile robots," *Robotics and Autonomous Systems*, vol. 59, no. 11, pp. 1001–1007, 2011.
- [27] M. Vazquez, M. Ardito-Proulx, and S. Wadoo, "Nonlinear kinematic control of qbot2-trajectory tracking," in *MIT Undergraduate Research Technology Conference*, pp. 1–4, IEEE, 2019.
- [28] G. Díaz-García, L. F. Giraldo, and S. Jimenez-Leudo, "Dynamics of a differential wheeled robot: Control and trajectory error bound," in *5th Colombian Conference on Automatic Control*, pp. 25–30, IEEE, 2021.
- [29] M. Vazquez, M. Ardito-Proulx, and S. Wadoo, "Lyapunov based trajectory tracking dynamic control for a qbot-2," in *Integrated STEM Education Conference*, pp. 1–6, IEEE, 2020.
- [30] W. Zeng, Q. Wang, F. Liu, and Y. Wang, "Learning from adaptive neural network output feedback control of a unicycle-type mobile robot," *ISA transactions*, vol. 61, pp. 337–347, 2016.
- [31] Z. Chen, Y. Liu, W. He, H. Qiao, and H. Ji, "Adaptive-neural-network-based trajectory tracking control for a nonholonomic wheeled mobile robot with velocity constraints," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 6, pp. 5057–5067, 2020.
- [32] M. Asai, G. Chen, and I. Takami, "Neural network trajectory tracking of tracked mobile robot," in *16th International Multi-Conference on Systems, Signals & Devices*, pp. 225–230, IEEE, 2019.
- [33] N. Hassan and A. Saleem, "Neural network-based adaptive controller for trajectory tracking of wheeled mobile robots," *IEEE Access*, vol. 10, pp. 13582–13597, 2022.
- [34] P. Petrov and I. Kralov, "A look-ahead approach to mobile robot path tracking based on distance-only measurements," in *AIP Conference Proceedings*, vol. 2172, p. 110005, AIP Publishing LLC, 2019.
- [35] A. Bayuwindra, E. Lefeber, J. Ploeg, and H. Nijmeijer, "Extended look-ahead tracking controller with orientation-error observer for vehicle platooning," *Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4808–4821, 2020.
- [36] Z. Zheng and M. Feroskhan, "Path following of a surface vessel with prescribed performance in the presence of input saturation and external disturbances," *IEEE Transactions on Mechatronics*, vol. 22, no. 6, pp. 2564–2575, 2017.

- [37] A. Bayuwindra, E. Lefeber, J. Ploeg, and H. Nijmeijer, “Extended look-ahead tracking controller with orientation-error observer for vehicle platooning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4808–4821, 2019.
- [38] N. Wang, Z. Sun, J. Yin, Z. Zou, and S.-F. Su, “Fuzzy unknown observer-based robust adaptive path following control of underactuated surface vehicles subject to multiple unknowns,” *Ocean Engineering*, vol. 176, pp. 57–64, 2019.
- [39] F. Ke, Z. Li, and C. Yang, “Robust tube-based predictive control for visual servoing of constrained differential-drive mobile robots,” *Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3437–3446, 2017.
- [40] Z. Liu and Y. Wang, “Trajectory tracking of mobile service robot,” in *Chinese Control And Decision Conference*, pp. 1516–1520, IEEE, 2019.
- [41] S. Miah, W. Gueaieb, P. A. Farkas, S. Al-Sharhan, and D. Spinello, “Linear time-invariant feedback operator for mobile robot trajectory tracking,” in *International Instrumentation and Measurement Technology Conference Proceedings*, pp. 751–756, IEEE, 2015.
- [42] T. Henningsson, E. Johannesson, and A. Cervin, “Sporadic event-based control of first-order linear stochastic systems,” *Automatica*, vol. 44, no. 11, pp. 2890–2895, 2008.
- [43] D. Lehmann and J. Lunze, “Event-based output-feedback control,” in *19th Mediterranean Conference on Control & Automation*, pp. 982–987, 2011.
- [44] J. Yang, F. Xiao, and T. Chen, “Event-triggered formation tracking control of nonholonomic mobile robots without velocity measurements,” *Automatica*, vol. 112, p. 108671, 2020.
- [45] A. Eqtami, S. Heshmati-Alamdari, D. V. Dimarogonas, and K. J. Kyriakopoulos, “Self-triggered model predictive control for nonholonomic systems,” in *2013 European Control Conference*, pp. 638–643, IEEE, 2013.
- [46] C. Xie, Y. Fan, and J. Qiu, “Event-based tracking control for nonholonomic mobile robots,” *Nonlinear Analysis: Hybrid Systems*, vol. 38, p. 100945, 2020.
- [47] F. Lawless, H. Suthar, T. Chambers, S. Butts, D. Eure, N. Bousaba, and D. Maity, “Trajectory tracking for mobile robots with human-in-the-loop,” in *SoutheastCon*, pp. 385–390, IEEE, 2022.
- [48] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.
- [49] H. K. Khalil, *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002.

- [50] F. Lawless, “Trajectory tracking simulation and hardware code,” 2022. available at: <https://drive.google.com/drive/folders/13WPrQLB3peyvEUG3f659AAQZ6YZBawin?usp=sharing>.
- [51] O. Egeland, M. Dalsmo, and O. J. Soerdalen, “Feedback control of a nonholonomic underwater vehicle with a constant desired configuration,” *The International journal of robotics research*, vol. 15, no. 1, pp. 24–35, 1996.
- [52] M. Yaghi and M. Ånder Efe, “ H_2/H_∞ -neural-based fopid controller applied for radar-guided missile,” *Transactions on Industrial Electronics*, vol. 67, no. 6, pp. 4806–4814, 2020.

APPENDIX A: Optimal Steering of Linear Systems

Let us consider a linear time invariant system

$$\dot{X} = AX + BU, \quad X(t_0) = X_{\text{initial}}$$

where the objective is to construct a control input U^* that steers the initial state X_{initial} at time t_0 to a final state X_{final} at time t_f .

Lemma 12 *The optimal controller U^* to steer initial state X_{initial} at time t_0 to a final state X_{final} at time t_f while minimizing $\int_{t_0}^{t_f} \|U^*(t)\|^2 dt$ is*

$$U^*(t) = -B^\top e^{A^\top(t_f-t)} W(t_f, t_0)^{-1} \left(e^{A(t_f-t_0)} X_{\text{initial}} - X_{\text{final}} \right), \quad \forall t \in [t_0, t_f]$$

where

$$W(t_f, t_0) = \int_{t_0}^{t_f} e^{A(t_f-t)} B B^\top e^{A^\top(t_f-t)} dt.$$