A Novel Tracking Control of Mobile Robots Based on Bioinspired Neural Dynamics and Unscented Kalman Filter

by

Zhe Xu

A Thesis presented to The University of Guelph

In partial fulfilment of requirements for the degree of Master of Applied Science in Engineering

Guelph, Ontario, Canada

© Zhe Xu, August, 2019
ABSTRACT

A NOVEL TRACKING CONTROL OF MOBILE ROBOTS BASED ON BIOINSPIRED NEURAL DYNAMICS AND UNSCENTED KALMAN FILTER

Zhe Xu

University of Guelph, 2019

Advisor: Professor Simon X. Yang

Tracking control has been a vital research area for robotics. It is important to design robotic systems with smooth velocity commands and robustness to system and measurement noises. This thesis presents a novel controller based on a bioinspired neural dynamic model and the unscented Kalman filter. Typical backstepping techniques usually suffer from speed jumps, which may create discontinuities in the speed of robotic systems. This disadvantage of backstepping techniques creates difficulties in applications to real robots. The bioinspired backstepping controller is able to generate smooth velocity commands without any speed jumps. Additionally, in real world applications, the robotic system may operate in noisy environments or be equipped with imperfect sensors that result in inaccurate measurements due to the system and measurement noises. The unscented Kalman filter is capable of reducing the effects of these noises and providing accurate estimates. Therefore, the innovative contribution of this thesis is that the unscented Kalman filter is integrated with the bioinspired backstepping control, minimizing the effects of the system and measurement noises to the mobile robot. Simulation experiments demonstrate the efficiency and effectiveness of the proposed bioinspired controller with unscented Kalman filter in removing speed jumps existed in conventional backstepping controllers, and providing accurate state estimates of robotic systems.
DEDICATION

To my parents and advisor,

Thank you for raising me and training me to help me achieve one of my greatest goals in my life.
ACKNOWLEDGEMENTS

First, I would like to thank my advisor, professor Simon X. Yang. His exceptional work in control theory made this thesis possible to complete. His hard work and patience encouraged me and eventually led me into the completion of my thesis.

I am also grateful to professor Stephen Andrew Gadsden as my committee member. His valuable work in estimation theory guided the success of my thesis. Without being enlightened by him, I would not have completed my thesis.

The ARIS Lab provided a great environment, which helped me to complete my work in this lab. Therefore, I would like to thank all my colleagues for their kindness.
TABLE OF CONTENTS

Abstract ........................................................................................................................................ ii
Dedication...................................................................................................................................... iii
Acknowledgements .......................................................................................................................... iv
List of Tables ................................................................................................................................... viii
List of Figures ................................................................................................................................. ix
List of Symbols .............................................................................................................................. xi
Chapter 1  Introduction .................................................................................................................. 1
  1.1 Problem Statement ................................................................................................................... 2
  1.2 Objectives of This Thesis ......................................................................................................... 3
  1.3 Contributions of This Thesis ................................................................................................... 4
  1.4 Organization of This Thesis ................................................................................................... 5
Chapter 2  Background and Literature Review .............................................................................. 7
  2.1 Robot Control ......................................................................................................................... 8
    2.1.1 Backstepping Control ....................................................................................................... 8
    2.1.2 Sliding Mode Control ....................................................................................................... 10
    2.1.3 Linearization Control ....................................................................................................... 10
    2.1.4 Neural Networks and Fuzzy Systems Based Control ....................................................... 18
  2.2 Estimation Techniques ............................................................................................................. 21
    2.2.1 Classic Kalman Filter ....................................................................................................... 21
    2.2.2 Extended Kalman Filter .................................................................................................. 23
    2.2.3 Unscented Kalman Filter ................................................................................................. 25
### Chapter 2: Modeling of Nonholonomic Mobile Robots

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3</td>
<td>Dynamic Model of Mobile Robots</td>
<td>30</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Tracking Error and Error Dynamics</td>
<td>32</td>
</tr>
<tr>
<td>2.3.3</td>
<td>Tracking Control Objectives</td>
<td>33</td>
</tr>
<tr>
<td>2.4</td>
<td>Summary</td>
<td>34</td>
</tr>
</tbody>
</table>

### Chapter 3: The Proposed Bioinspired Controller with Unscented Kalman Filter

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>The Bioinspired Controller with Unscented Kalman Filter</td>
<td>36</td>
</tr>
<tr>
<td>3.2</td>
<td>Bioinspired Backstepping Controller</td>
<td>37</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Backstepping Control</td>
<td>38</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Bioinspired Neural Dynamics</td>
<td>39</td>
</tr>
<tr>
<td>3.3</td>
<td>Bioinspired Controller with Unscented Kalman Filter</td>
<td>42</td>
</tr>
<tr>
<td>3.4</td>
<td>Stability Analysis</td>
<td>44</td>
</tr>
<tr>
<td>3.5</td>
<td>Summary</td>
<td>47</td>
</tr>
</tbody>
</table>

### Chapter 4: Simulation Results

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Tracking Straight Paths</td>
<td>49</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Straight Path Tracking without Noises</td>
<td>49</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Straight Path Tracking with Unscented Kalman Filter</td>
<td>55</td>
</tr>
<tr>
<td>4.2</td>
<td>Tracking Circular Paths</td>
<td>60</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Circular Path Tracking without Noises</td>
<td>61</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Circular Path Tracking with Unscented Kalman Filter</td>
<td>66</td>
</tr>
<tr>
<td>4.3</td>
<td>Tracking Elliptical Paths</td>
<td>69</td>
</tr>
<tr>
<td>4.4</td>
<td>Summary and Discussion</td>
<td>72</td>
</tr>
</tbody>
</table>

### Chapter 5: Conclusions and Future Works

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
<td>Summary and Discussion</td>
<td>74</td>
</tr>
</tbody>
</table>
5.1 Conclusions .................................................................................................................................................. 74

5.2 Future Works .............................................................................................................................................. 75

References ........................................................................................................................................................... 78
LIST OF TABLES

Table 4.1. Speed overshoot comparison between backstepping and bioinspired controllers..53
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Wheeled mobile robot configuration</td>
<td>11</td>
</tr>
<tr>
<td>2.2</td>
<td>Optimal fuzzy system architecture (Redrawn from Sun et al., 2018)</td>
<td>20</td>
</tr>
<tr>
<td>2.3</td>
<td>Distribution of sigma point set for unscented Kalman filter in 2D space</td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>(From Haykin, 2001)</td>
<td></td>
</tr>
<tr>
<td>2.4</td>
<td>Dynamic model of a nonholonomic mobile robot (Redrawn from Yang, 2012)</td>
<td>30</td>
</tr>
<tr>
<td>2.5</td>
<td>Error dynamics of a nonholonomic mobile robot</td>
<td>35</td>
</tr>
<tr>
<td>3.1</td>
<td>System setup for bioinspired backstepping controller with unscented Kalman</td>
<td>37</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparison of straight path (y=4) tracking.</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>(a) Position; (b) Angular Velocity; (c) Linear velocity</td>
<td></td>
</tr>
<tr>
<td>4.2</td>
<td>Comparison of straight path (y=5) tracking.</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>(a) Position; (b) Angular Velocity; (c) Linear velocity</td>
<td></td>
</tr>
<tr>
<td>4.3</td>
<td>Comparison of straight path (y=6) tracking.</td>
<td>54</td>
</tr>
<tr>
<td></td>
<td>(a) Position; (b) Angular Velocity; (c) Linear velocity</td>
<td></td>
</tr>
<tr>
<td>4.4</td>
<td>Comparison of desired state and actual measurement.</td>
<td>56</td>
</tr>
<tr>
<td></td>
<td>(a) Position; (b) Angular velocity; (c) Linear velocity</td>
<td></td>
</tr>
<tr>
<td>4.5</td>
<td>Comparison of straight path tracking with/without filter.</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>(a) Position; (b) Angular velocity; (c) Linear velocity</td>
<td></td>
</tr>
<tr>
<td>4.6</td>
<td>Measurement with variance of $10^{-3}$ measurement noise.</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>(a) Position; (b) Angular velocity; (c) Linear velocity</td>
<td></td>
</tr>
</tbody>
</table>
4.7 Comparison of circular path \((x^2+y^2=1)\) tracking. (a) Position; (b) Angular velocity; (c) Linear velocity. ................................................................. 63

4.8 Comparison of circular path \((x^2+y^2=2.25)\) tracking. (a) Position; (b) Angular velocity; (c) Linear velocity. ................................................................. 64

4.9 Circular path tracking with different initial state \((-0.5,1,0)\). (a) Position; (b) Angular velocity; (c) Linear velocity. ................................................................. 65

4.10 Comparison of desired state and actual measurement. (a) Position; (b) Angular velocity; (c) Linear velocity................................................................. 67

4.11 Measurement with variance of \(2 \times 10^{-4}\) measurement noise. (a) Position; (b) Angular velocity; (c) Linear velocity................................................................. 68

4.12 Elliptical path \((x^2+1.5y^2=2.25)\) tracking. (a) Position; (b) Angular velocity; (c) Linear velocity. ................................................................. 70

4.13 Elliptical path \((x^2+1.5y^2=2.25)\) tracking. (a) Position; (b) Angular velocity; (c) Linear velocity. ................................................................. 71

4.14 Tracking errors for elliptical path. (a) Linear Velocity (b) Angular velocity ............... 73

5.1 Straight path tracking with variance of \(10^{-4}\) system and measurement noises. (a) Position; (b) Angular velocity; (c) Linear velocity................................................................. 76
# LIST OF SYMBOLS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>Passive decay rate of the shunting model</td>
</tr>
<tr>
<td>$a, b$</td>
<td>Distances of the front and rear axles to the mass center of the vehicle</td>
</tr>
<tr>
<td>$B$</td>
<td>Upper bound of the neural activity</td>
</tr>
<tr>
<td>$C_1, C_2, C_3$</td>
<td>Gain parameters of the bioinspired controller</td>
</tr>
<tr>
<td>$C_m$</td>
<td>Membrane capacitance</td>
</tr>
<tr>
<td>$D$</td>
<td>Lower bound of the neural activity</td>
</tr>
<tr>
<td>$E$</td>
<td>A $3 \times 1$ error vector $[e_X, e_Y, e_{\theta}]^T$</td>
</tr>
<tr>
<td>$E_p, E_{Na}, E_k$</td>
<td>Nernst potentials for potassium, sodium ions and passive leak current</td>
</tr>
<tr>
<td>$e_X, e_Y, e_{\theta}$</td>
<td>Tracking error in inertial frame</td>
</tr>
<tr>
<td>$e_x, e_y, e_{\psi}$</td>
<td>Tracking error in body fixed frame</td>
</tr>
<tr>
<td>$F$</td>
<td>System matrix of the robotic system</td>
</tr>
<tr>
<td>$F_{yf}, F_{yr}$</td>
<td>Forces acting on the front and rear driving wheels</td>
</tr>
<tr>
<td>$f(.)$</td>
<td>Nonlinear system model</td>
</tr>
<tr>
<td>$f'(.)$</td>
<td>Threshold function</td>
</tr>
<tr>
<td>$g_P, g_{Na}, g_K$</td>
<td>Conductance of potassium, sodium, and passive channels</td>
</tr>
<tr>
<td>$\hat{g}(.)$</td>
<td>Inhibitory nonlinear input</td>
</tr>
<tr>
<td>$H$</td>
<td>Measurement matrix of the robotic system</td>
</tr>
<tr>
<td>$h$</td>
<td>Nonlinear system measurement model</td>
</tr>
<tr>
<td>$I$</td>
<td>Identity matrix</td>
</tr>
<tr>
<td>$I_{zz}$</td>
<td>Inertial in yaw movement of the mobile robot</td>
</tr>
</tbody>
</table>
\( i, k \) \hspace{1cm} \text{Time step index}

\( K \) \hspace{1cm} \text{Kalman gain}

\( m \) \hspace{1cm} \text{Mass of the mobile robot}

\( P \) \hspace{1cm} \text{Predicted corresponding state error covariance matrix}

\( P_c \) \hspace{1cm} \text{Current mobile robot posture}

\( P_d \) \hspace{1cm} \text{Desired mobile robot posture}

\( P_{zz}, P_{xz} \) \hspace{1cm} \text{Measurement covariance and cross covariance}

\( Q \) \hspace{1cm} \text{Current corresponding state error covariance matrix}

\( S \) \hspace{1cm} \text{Sliding Manifold}

\( T_e \) \hspace{1cm} \text{Transformation matrix}

\( W_0 \) \hspace{1cm} \text{First corresponding weight}

\( X_0 \) \hspace{1cm} \text{First corresponding sigma point}

\( X_c, X_d \) \hspace{1cm} \text{Actual and desired postures}

\( (x_c, y_c, \theta_c) \) \hspace{1cm} \text{Current coordinates of the mobile robot in inertial frame}

\( (x_d, y_d, \theta_d) \) \hspace{1cm} \text{Desired coordinates of the mobile robot in inertial frame}

\( \hat{x} \) \hspace{1cm} \text{State estimate}

\( \hat{y} \) \hspace{1cm} \text{Measurement error}

\( \hat{Z} \) \hspace{1cm} \text{Measurement of the nonlinear system}

\( z_k \) \hspace{1cm} \text{Measurement matrix}

\( \alpha \) \hspace{1cm} \text{Steering angle of the front wheel driven robot}

\( \beta \) \hspace{1cm} \text{Sliding slip displacement of the mobile robot}

\( \gamma \) \hspace{1cm} \text{Yaw displacement of the mobile robot}
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta$</td>
<td>Steering angle of the mobile robot</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>System noise</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Measurement noise</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Continuous function for sliding mode control</td>
</tr>
<tr>
<td>$v_c$</td>
<td>Current linear velocity</td>
</tr>
<tr>
<td>$v_d$</td>
<td>Desired linear velocity</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Angular velocity of the front wheel driven mobile robot</td>
</tr>
<tr>
<td>$\omega_c$</td>
<td>Current angular velocity</td>
</tr>
<tr>
<td>$\omega_d$</td>
<td>Desired angular velocity</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

The term “robot” was firstly officially defined in 1920 by a Czech writer whose name is Karel Capek. This term was originally a name from a Slavic language as ‘robota’, which means forced labor in English (Hemal, 2011). Since then, this area has drawn great research attention, many scientists and engineers have been working in robotic research areas for their entire life. Robots have stepped into people’s daily life, as it has been a vital component of modern society. For example, robot arms have been implemented in manufacturing factories to reduce the labor costs and unmanned ground robots have been used for the distribution of parts in the assembly line. In fact, this area is still one of the most advanced topics in the world, as each year there are many problems yet to be solved.

In academia, robotics is an interdisciplinary study. It applies both engineering and computer science knowledge in its design, manufacturing, and operation. The first mobile robot was built by W. Grey Walter to study the behaviors of the robot (Hemal, 2011). Since then, the scientific area started experiencing a boom of robotic technology. There are many branches in robotic research areas, such as trajectory tracking, estimation theory, and simultaneous localization and mapping. The main challenges in robotic research include signal disturbance, the effectiveness and efficiency of tracking, and accuracy of the feedback.
This thesis mainly discusses the trajectory tracking control for mobile robots, because trajectory tracking has been one of the most popular topics in robotic control. Although there are multiple approaches existing for robotic systems, they are either hard to implement or there are difficulties in current methods, which still need improvements to make them work.

Mobile robots are mainly divided into two different categories, holonomic mobile robots and non-holonomic mobile robots. The difference between these two types of mobile robots is that the non-holonomic robotic systems have extra differential constraints over the holonomic robotic systems, such that the parameters may be returned to its original values, but the system may not return to its original states. Although both types of mobile robots have 3 degrees of freedom, non-holonomic mobile robots are subjected to a kinematic constraint, which does not exist in holonomic mobile robots (Fu et al., 2018).

Most mobile vehicles are non-holonomic. The best example of a non-holonomic mobile system is the cars people are driving on the road, thus, it is crucial to do research in this area since it affects people’s daily life. This thesis mainly focuses on trajectory tracking for non-holonomic mobile robots. Though this problem has been investigated deeply by many researchers in recent years, there is still much progress that can be done in order to make the tracking more efficient.

1.1 Problem Statement

For the trajectory tracking technique, although there are many approaches available, there are still many gaps in current methods. For example, typical backstepping techniques usually suffer from speed jump and large speed overshoot issues, which may create discontinuities in robotic
systems. These speed jump and overshoot issues make backstepping methods hard to implement on real robots.

In real applications, where robots might face noisy environments that create large system and measurement noises, the noises have huge impacts on the accuracy of trajectory tracking. If a robotic system is affected by a large amount of noises, the robotic system will fail to yield accurate measurements with no doubt. If the current state of a robotic system cannot be measured accurately, the system will follow an unpredictable path due to the incorrect readings. In addition, if an operator is not able to collect accurate readings, the desired path of the mobile robot might be set inaccurately due to the misjudgment of the current situation.

1.2 Objectives of This Thesis

The purpose of this thesis is to develop a controller that is robust to uncertain noises while dealing with the speed jump and overshoot issues occurring in backstepping techniques. It is important to make robotic systems with smooth velocity commands, which typical backstepping methods cannot achieve due to its natural set up. In addition, control methods must be robust to uncertain noises to ensure consistent performance, for which the unscented Kalman filter is an effective tool. Therefore, the objectives of this thesis are divided into two parts:

1. To test the existing bioinspired method for overcoming the speed jump and overshoot issues and generating smooth velocity commands.
2. To Integrate the unscented Kalman filter into the bioinspired control method to minimize the effects of measurement and system noises from the controller.

The goal of implementation of these two control techniques provides is to create smooth velocity commands in noisy environments, with increased accuracy of mobile robot control in real world applications.

1.3 Contributions of This Thesis

This thesis implements an existing bioinspired backstepping controller (Yang et al., 2012) and it has been combined with unscented Kalman filter to minimize the effects of uncertain noises toward estimates. Overall, this thesis has done various works in trajectory tracking control, which can be summarized as follows:

1. A deeper analysis of the reasons is presented why backstepping tracking control is not feasible when large tracking error occurs.

2. The relations between the linear velocity overshoot for both tracking control and backstepping tracking control are investigated. The results indicate that speed overshoot has a linear relation with tracking error.

3. Propose a bioinspired backstepping controller with unscented Kalman filter is proposed, which resolves the speed jump and overshoot issues and minimizes the effects of system and measurement noises in trajectory tracking.
4. Multiple simulations are conducted to demonstrate the efficiency and effectiveness of the proposed novel backstepping controller.

1.4 Organization of This Thesis

There are five chapters in this thesis. The organization of this thesis is presented as follows:

Chapter 1 identifies the problems that are going to be addressed in this thesis and provides the reasons why tracking control is important for robotic systems. The novel contributions of this thesis are summarized in Chapter 1.

Chapter 2 provides the background and literature review of the existing works. There are four most commonly used robot control strategies that are discussed in Section 2.1. Then, three existing Kalman filters are presented in Section 2.2 from the view of estimation theory. After that, the modeling of nonholonomic mobile robot dynamics are demonstrated, which is used in the design of the proposed bioinspired controller. Finally, a summary is provided to illustrate why this thesis chooses to design the proposed bioinspired controller based on backstepping technique and unscented Kalman filter.

Chapter 3 demonstrates how the bioinspired neural dynamics and unscented Kalman filter are combined with typical backstepping controller. The design process of the proposed controller has also been detailed in Chapter 3.

Chapter 4 presents the results of multiple simulations to show the efficiency and effectiveness of the proposed controller in terms of avoiding speed jump, speed overshoot, and
providing reliable estimates. The proposed controller is used to track a linear path and a circular path to show its flexibility, additionally, several discussions explain why the bioinspired neural dynamics and unscented Kalman filter are capable of avoiding speed jumps and overshoots issues and providing accurate estimates regardless of the measurement and system noises.

Chapter 5 provides several concluding remarks on the proposed bioinspired controller with unscented Kalman filter. In addition, there are some limitations that are also discussed, and several future works are provided as well.
Chapter 2

Background and Literature Review

The research on mobile robotics has been on-going for many years because of its vast applications in military purposes and civilian usage (Sun et al., 2013). The developments of control techniques, especially for mobile robots, are not as difficult as years before due to the invention of multiple software. Therefore, there are many different control strategies that have been developed in recent years, and these works can be divided into four different categories: sliding control, backstepping control, linearization control, and neural networks and fuzzy systems based control. This chapter reviews the works done previously and compares the advantages and disadvantages of each control method. In addition, the estimation strategies are equally important as control methods under of the larger picture of control theory. Thus, three estimation techniques, which are classic Kalman filter, extended Kalman filter, and unscented Kalman filter are discussed in this section as well. Each estimation technique also has pros and cons, however, the reason why this thesis chooses to use unscented Kalman filter is based on multiple factors, which are discussed in this section. Finally, the dynamic model of the nonholonomic system (Yuan, 2001) is provided to demonstrate how the nonholonomic mobile robot works, because this thesis is focusing on implementing the existing bioinspired backstepping controller and unscented Kalman filter on a nonholonomic mobile robot, This novel controller was heavily inspired by these previous works and eventually led to the completion of this thesis.
2.1 Robot Control

This section discusses the advantages and disadvantages of linearization control, sliding mode control, neural networks and fuzzy systems based control, and backstepping control. Additionally, there are several recent studies of these methods, which are discussed and reviewed. These four advanced control strategies are the core approaches in modern robotic control theory. Thus, it is important to identify the characteristics of each method because there is no method that is superior to the other one, as each of them is suitable in certain conditions.

2.1.1 Backstepping Control

The backstepping technique was first proposed by Kokotovic (1992), it was intended to control certain nonlinear dynamic systems. This method contains a recursive structure that gives a feedback to the controller and adjusts the systems’ state and eventually terminates when the systems reach their desired state. The stability of this method can be proven by Lyapunov stability theory. The basic structure of backstepping technique for a three degrees of freedom mobile robot can be illustrated as (Fierro & Lewis, 1997)

\[ e_p = T_e (P_d - P_c) \]  \hspace{1cm} (2.13)

\[ E = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos \theta_c & \sin \theta_c & 0 \\ -\sin \theta_c & \cos \theta_c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d - x_c \\ y_d - y_c \\ \theta_d - \theta_c \end{bmatrix} \]  \hspace{1cm} (2.14)

where \( e_p \) is the tracking error in the inertia frame, \( T_e \) is the transformation matrix, and \( P_d \) and \( P_c \) are respectively the desired posture and measured posture in the body-fixed frame. Variables \( e_x \),
$e_X$, and $e_\theta$ are the errors in $X$ and $Y$ axis, and rotational angle $\theta$ in lateral direction, respectively. Variables $x_d$, $y_d$, and $\theta_d$ are the components of the desired posture in the inertia frame and $x_c$, $y_c$, and $\theta_c$ are components of the measured posture in inertia frame. The velocity commands of the backstepping controller based on error dynamics (2.14) is written as

$$V_c = \left[ \begin{array}{c} \omega_r + C_2 v_r e_\theta \\ C_1 e_\theta \end{array} \right]$$

where $V_c$ is the velocity command of the mobile robot in a 2D Cartesian workspace, and $C_1$, $C_2$, and $C_3$ are positive constants. Equations (2.13) to (2.16) are the fundamental design of the backstepping technique for control systems in a 2D Cartesian workspace and this method and its variations have been used in many applications (Depature et al., 2018; Kwan & Lewis, 2000; Yang et al., 2012; Zhang & Li, 2018).

Backstepping technique is relatively easy to implement compared to sliding mode, linearization, neural networks, and fuzzy logic control. In addition, this method is the most commonly used method in trajectory tracking problems. Therefore, the reason why this thesis chooses to use this method is because there are plenty of resources available in the literature, and this backstepping method is compatible with many other strategies, such as Kalman Filter and its variations.

Comparing between backstepping control and sliding mode control, backstepping control does not have chattering problems whereas chattering issues cannot be completely eliminated from sliding mode control. However, backstepping technique is more sensitive to uncertainties,
therefore, one of the main purposes of this thesis is to increase the robustness of the backstepping technique.

Binh et al., (2019) proposed an adaptive backstepping controller, which they claimed reaches asymptotically stable without worrying about uncertainties. The adaptive term they added on their controller basically cancelled uncertainties caused by weight changes. Although they increased robustness overall based on the adaptive term they added, they did not perform their simulation results on a real robot. Thus, it can be inferred that their controller would only work in an ideal condition and their studies did not include the effects of uncertain noises on the robot and its measurement. Therefore, this thesis is going to mainly focus on dealing with noises in the control system.

Cui et al., (2016) proposed a backstepping controller with unscented Kalman filter, which their results showed that the backstepping technique is able to track their trajectory, whereas unscented Kalman filter provides accurate estimates. However, they also did a test where a certain tracking error in the desired trajectory was introduced, and the results were as expected as the speed overshoot occurred, which made their proposed controller only able to stay in a simulation stage, because the speed overshoot might disable the robot due to potential acceleration constraints.

2.1.2 Sliding Mode Control

Sliding mode control is a well-developed method in the robotic research area, it has been introduced for many years, which can be traced back to at least the 1980s (Slotine & Sastry, 1983). The system slides between two sliding manifolds, which forces the system to switch between two
states and eventually leads the system to stability. Therefore, these kinds of systems have chattering problems because the system basically slides between two different states. In addition, sliding mode uses discontinuous control signals to force the systems chatter between two states, which means the system, when implemented in robotic systems, may have difficulties such as the loss of internal energy, vibration of the system, and even damage to the equipment used. Thus, to implement this method into real applications, people should proceed with extreme caution.

The general functions of the sliding mode control for trajectory tracking can be described as follows (Mechlih, 1993). A model of a front wheel driving mobile robot is shown in Figure 2.1.
Assuming the posture of the front wheel driven nonholonomic mobile robot is designed as a function of the steering angle and angular speed of the front wheel,

\[ \dot{P} = \phi(\alpha, \omega) \]  

where \( P \) is the posture of the mobile robot, \( \alpha \) and \( \omega \) are the steering angle and angular speed of the front wheel, respectively. Then, the general control law is designed as

\[
\alpha_c(P_c, t) = \begin{cases} 
\alpha_d(P_d, t) + k & \text{if } S < 0 \\
\alpha_d(P_d, t) - k & \text{if } S \geq 0
\end{cases}
\]  

(2.9)

where \( P_c \) and \( P_d \) are the current and desired postures of the mobile robot, respectively; \( \alpha_c \) and \( \alpha_d \) are respectively the current and desired steering angle of the mobile robot; \( k \) is the switching angle factor; \( t \) is the time iteration. The sliding surface \( S \) is defined as

\[ S = y - g(x) \]  

(2.10)

where \( g(x) \) generates the desired posture in a 2-dimensional Cartesian workspace \((x, y)\). In addition, when the error between current and desired postures of the mobile robot is zero, \( S \) would be zero as well. Then, the switching term of the sliding mode control can be written as

\[ \alpha(P_c, t) = \alpha_d(P_d, t) + k(e_\theta)\text{sgn}(S) \]  

(2.11)

where \( e_\theta \) is the angle error between the desired posture and actual posture of the mobile robot. The sign function, \( \text{sgn}(S) \), is defined as
\[
\text{sgn}(S) = \begin{cases} 
+1 & \text{if } S \geq 0 \\
-1 & \text{if } S < 0 
\end{cases}
\] (2.12)

As can be seen from (2.12), the sign function is the switching term of the sliding mode control. It is not hard to observe why the sliding mode control has chattering problems, because the sign function would only provide +1 or -1 to the system and there is nothing in between. Therefore, the mobile robot control system will slide between two different modes as the system reaches stability.

Despite all the disadvantages the sliding mode control has, there are some advantages as well. One of the biggest advantages for sliding mode is that this method is robust to disturbance. The reason why sliding mode has such high robustness is because the control is sliding between two different states, so the system does not need to be precise, and the system is not sensitive to the change of parameters. Due to its strong robustness, many applications have taken advantage of this method and implemented it into real life applications.

Zhu and Sun (2013) proposed a hybrid control strategy for an unmanned underwater vehicle, in which they used a bioinspired neural dynamics model along with a sliding mode torque control for the thrusts. They were aware that the chattering issue for the sliding mode control was caused by the discontinuous term, therefore, they added an adaptive term (Soylu et al., 2008) into the control law to replace the sign function. The results showed that they have overcome the chattering issue and implemented their results on a real underwater robot from their university.

There is another group of researchers who proposed a chattering free sliding mode control
in tracking for underactuated hovercraft (Fu et al., 2018). Based on the observation from their design of surge and yaw force control, the fundamental problem that is embedded in sliding mode control, the sign function, still remained untouched. Although their results may have less of a chattering issue because they brought a radial basis function into the sliding mode control, they did not touch the sign function at all. Therefore, the chattering issue was not completely solved by them yet.

Several others were trying to solve the chattering problems as well (Sayed et al., 2012), they proposed a strategy by using two different sliding manifolds to increase the robustness of the system. Although this strategy made the system become more robust to system model changes, the fundamental problem, which was the chattering problem from the sign function in sliding mode control, remained unsolved.

Sliding mode control is quite robust to disturbance, which has many applications on surface vessels. For example, Ashrafiuon et al. (2008) proposed a controller, which used sliding mode to control the sway and surge movements of a surface vessel and implemented their controller on a real robot. The chattering issue still occurred, however, in their case, making the controller more robust to disturbances was more important for surface vessels. Additionally, they claimed that their results were affected by the experimental set ups due to the uneven thrusters power and voltage across the circuits. If they had used filters, their readings might be more accurate and make their results more reliable. The estimation strategies, like filters, which are discussed and simulated in this thesis show their efficiency. Additionally, their experimental results showed that the voltage
across the circuits was not stable and oscillated, which might cause the system to fail because unstable voltage may create resonance along with sliding mode controller, which is not a safe operating condition.

### 2.1.3 Linearization Control

The linearization control technique is another common technique in control theory. This method was conducted by transferring the nonlinearities in the system into linear terms, namely, the input-output linear control system. For a nonlinear system to be able to linearize and perform effectively, the initial error between actual state and desired state cannot be large. This technique converts the nonlinear relations between input and output into linear relations by using methods such as the decoupling matrix or transformation matrix. It is obvious that if systems are complicated and the initial errors are large the linearization method will generate even larger errors, because linearization is just an approximation of nonlinear functions.

The input-output linearization based approaches have been used in trajectory tracking in many studies (Chwa, 2010; d’Andrea-Novel et al., 1995; Sun et al., 2018). One of the typical linearization approaches of trajectory tracking is introduced by Sun et al. (2018). They proposed a path-tracking steering controller for mobile vehicles. The sideslip $\dot{\beta}$ and yaw $\dot{\gamma}$ rates they introduced are described as

$$\dot{\beta} = \frac{F_{yf} \cos(\delta) + F_{yr}}{m \mu_x} - \gamma$$

(2.1)
\[
\dot{\gamma} = \frac{aF_{yf}\cos(\delta) - bF_{yr}}{I_{zz}}
\]

(2.2)

where \(m\) is the mass of the mobile robot and \(I_{zz}\) is the inertia in yaw movement. Variables \(F_{yf}\) and \(F_{yr}\) are the force of the front and rear axles, respectively. Parameters \(a\) and \(b\) are the distances of the front and rear axles to the mass center of the mobile robot, respectively. By observing (2.1) and (2.2), it is obvious that the dynamic model of the robot is nonlinear. Therefore, Sun proposed a string of suppositions. Firstly, they assumed the steering angle is fixed at each time iteration, which is shown as

\[
\Delta\delta_a(k + i) = \Delta\delta_{Np}, \quad i = 0, ..., N_p - 1,
\]

(2.3)

where \(\Delta\delta_a(k + i)\) is the desired steering angle at \(k + i\)th iteration and \(\Delta\delta_{Np}\) is the corresponding fixed iteration. Then, assuming the vehicle tracks its desired path at iteration \(N_p - 1\), then the steering angle is rewritten as

\[
\Delta\delta_a(k + i) = \frac{\Delta\delta(k) - \Delta\delta_a(k + N_p - 1)}{N_p}
\]

(2.4)

\[
\delta_a(k + i) = \delta(k) + i\Delta\delta_{Np}.
\]

(2.5)

Then, by combining (2.1), (2.2), and (2.5), the nonlinear model of the vehicle is linearized, which is written as

\[
\dot{\beta} = \frac{F_{yf}(k + i)\cos(\delta_a(k + i)) + F_{yr}(k + i)}{m\mu_x} - \gamma(k + i)
\]

(2.6)
\[ \dot{\gamma} = \frac{aF_{yf}(k + i) \cos(\delta_a(k + i)) - bF_{yr}(k + i)}{I_{zz}}. \] (2.7)

The vehicle’s dynamic model that they proposed was linearized, however, they assumed that the steering angle is fixed at each time iteration, which means this method cannot take large tracking errors otherwise the system would fail.

There are still some advantages of the linearization method, such that this method is relatively easy compared to other approaches. In most cases, people only need to derive the decoupling matrix or transformation matrix for the nonlinear function to be able to get this method to work. In addition, there are a couple of research studies to overcome the disadvantage of linearization methods such as having a nonlinear feedback with linear control (Menton et al., 1991).

Liu et al. (2010) proposed a linearization method for an unmanned helicopter in altitude tracking. They noticed that the linearized model is just an estimate of the nonlinear model. Thus, they used a strategy called variance constrained trajectory linearization control to increase the robustness of their system. From their results, it can be seen that their proposed controller is robust to disturbance. However, the disturbance they brought into their system is relatively small, therefore, it did not have much effect on the linearized system. Thus, they still did not solve the problem as mentioned that linearization method cannot deal with large initial error.

Qiu et al. (2019) tried to use linearization control to solve the tracking control problems. They proposed an adaptive trajectory linearization control for the surface vessels, which they
argued improved linearization with modeling uncertainties. Based on the results they had, one can observe that their algorithm is complicated and rather difficult to implement on a real surface vessel. In addition, there are multiple factors that can affect the tracking efficiency of the surface vessel, such as waves and winds. Therefore, linearization control is not actually suitable for surface vessels.

### 2.1.4 Neural Networks and Fuzzy Systems Based Control

Neural networks based control has become one of the most popular topics in control theory and in other areas. This method requires online learning, and is therefore expensive and complex to practice. The advantage of the neural network control is that it cannot provide the relations for complex uncertain nonlinear systems, in which mathematical explanations are difficult to describe.

In some cases, where accurate system dynamic models are impossible to obtain due to disturbances or environmental noises, neural networks based control may be used to obtain system dynamic models. Therefore, this neural network approach has many applications, such as applying adaptive neural networks to obtain dynamic models for remotely operated vehicles (Chu et al., 2017). Another research team also used a neural network in tracking control for a surface vehicle with unknown dynamics (Pan et al., 2013). They combined known system model and used a neural network to get unknown dynamics for their controller to work. Based on the observation of their paper, it can be concluded that the learning process would be long and require large amount of training to get a satisfactory output.

Li et al. (2015) introduced an adaptive neural network tracking design, which they used for
tracking on an uncertain nonlinear system with unknown time delay. The results they had were computationally strenuous due to the amount of calculations, and it required a lot of time to optimize the efficiency of the neural network. Thus, although they achieved the objective they were supposed to get, the process of getting the results was time consuming.

Although there are various types of neural networks that can be used in tracking control, the basic structures of neural network models that are used in control systems usually have one input layer, which contains multiple input neurons. Then, the inputs feed into multiple hidden layers, to weigh each input parameters to get the optimized value and combined results in the output layer. Since it simulates the human neural networks, the neural networks need to be trained to have it know what the optimized value is just like people need to have enough experience to make the right decision. Therefore, the more samples for the neural networks to train, the better the outcome from the neural network is. However, if a neural network been over or under trained, there be overfitting or underfitting issues. The overfitting problems usually happen because the neural networks are over-trained, the neural networks start to remember the results, and becomes less robust to unknown dynamics. In contrast, if a neural network has underfitting issues, this indicates that the neural network is trained less than what it needs to, and the output would not be efficient to be used in other applications yet.

Fuzzy logic based approach is another alternative method for trajectory tracking of robot control. This approach has drawn a great amount of attention from many researchers. However,
this method requires the setup of fuzzy based rules, which means this approach is hard to come up with fuzzy rules that are suitable for relevant conditions.

Sun et al. (2018) proposed an optimized fuzzy control algorithm for a three dimensional autonomous underwater vehicle, and the design of their controller can be seen from Figure 2.1. The fuzzy rules were computationally complicated, however, the overall performance in terms of path planning for their controller was satisfactory. To come up with such fuzzy rules would be time consuming and the logic for fuzzy rules has to be perfect.

![Figure 2.2. Optimal fuzzy system architecture. (Redrawn from Sun et al., 2018)](image)

There is another person who proposed a fuzzy adaptive tracking control for mobile robot (Chwa, 2012), and he used a fuzzy based approach for the robot system, in addition, with the application of the adaptive term on fuzzy controller, the results showed the proposed controller was able to track desired paths. However, his fuzzy rules were built based on multiple assumptions, which made his controller not able to be implemented on a real robot and only stayed in a simulation stage.
2.2 Estimation Techniques

This section focuses on the background and approaches of estimation strategies. The estimation strategies are crucial to robotic motion planning and trajectory optimization because it provide accurate estimates, which are crucial for tracking. The most commonly used estimation strategies are Kalman filter and its variations. Therefore, this section reviews some of the Kalman filters and illustrates how these filters work.

2.2.1 Classic Kalman Filter

Kalman filter was named by its inventor, Rudolf Kalman (1960). Kalman filter provides the optimized estimates for linear problems. It is used to filter the unwanted system and measurement noises. These noises are usually considered as Gaussian, which can be described as

\[ P(\lambda_k) \sim N(0, Q_k) \] and \[ P(\mu_k) \sim N(0, R_k) \], where \( \lambda_k \) is the system noise and \( \mu_k \) is the measurement noise. The typical linear system model can be defined as

\[
\begin{align*}
x_{k+1} &= Fx_k + Gu_k + \lambda_k \quad k = 1, 2, 3 \ldots n \\
z_{k+1} &= Hx_{k+1} + \mu_{k+1}
\end{align*}
\] (2.17)

where \( F \) is the state transition matrix for state \( x_k \), parameter \( G \) is the input matrix, \( \lambda_k \) is the system noise and \( u_k \) is the system input. Parameter \( z_{k+1} \) is the state measurement, \( H \) is the measurement matrix and \( \mu_{k+1} \) is the measurement noise, \( k \) is the \( k \)-th time iteration. Then the Classic Kalman filter at its predicting stage can be described as

\[
\hat{x}_{k+1|k} = F\hat{x}_{k|k} + Gu_k
\] (2.19)
\[ P_{k+1|k} = F P_{k|k} F^T + Q_k \]  

(2.20)

where \( \hat{x}_{k+1|k} \) is the priori state estimate and \( \hat{x}_{k|k} \) is the previous state estimate. Parameters \( P_{k+1|k}, P_{k|k}, \) and \( Q_k \) are predicted corresponding state error covariance matrix and current corresponding state error covariance matrix, and system noise covariance, respectively. Then, the system goes to its updating stage, which its functions may be defined as follows

\[ K_{k+1} = P_{k+1|k} H^T \left[ H P_{k+1|k} H^T + R_{k+1} \right]^{-1} \]  

(2.21)

\[ \hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \left[ z_{k+1|k} - H \hat{x}_{k+1|k} \right] \]  

(2.22)

where \( K_{k+1}, R_{k+1} \) and \( \hat{x}_{k+1|k+1} \) are the Kalman gain, measurement noise covariance, and its updated estimates, respectively. The entire \( [H P_{k+1|k} H^T + R_{k+1}]^{-1} \) is called the innovation covariance, usually represented by \( S_{k+1} \). Finally, a posteriori state error covariance shall be derived as

\[ P_{k+1|k+1} = [I - K_{k+1} H] P_{k+1|k} \]  

(2.23)

where \( P_{k+1|k+1} \) is the state error covariance and \( I \) is the identity matrix.

The functions above illustrate how the original Kalman filter works, it yields statistically optimal results for a well-defined linear system in Gaussian noise. However, most systems in nature are nonlinear, and the Kalman filter cannot provide accurate estimates for nonlinear
systems. Therefore, based on the Kalman filter, a number of variant Kalman filters are introduced to deal with the nonlinear systems, which is discussed in section 2.2.2 and 2.2.3.

Chi et al. (2014) introduced an extreme leaning machine based adaptive Kalman filter. Their model was able to track a ball, however, due to the limitations of the Kalman filter, they also assume the ball was moving linearly, which is quite hard to achieve in real situations. Thus, their results are highly limited in experimental stage and not testable in real situations, and one can easily understand why they did not implement their results on a real robot.

2.2.2 Extended Kalman Filter

Extended Kalman filter is an extension of Kalman filter. The benefit of this method is that this approach can be used in nonlinear systems. The way the extended Kalman filter works is that this filter linearizes the nonlinear system using the Jacobian matrix, the linearized system goes to the same processes as classic Kalman filter does. In order to derive the linearized system out of the nonlinear system, the partial derivatives are needed to calculate linearized system matrix $F$ and measurement matrix $H$, which can be calculated as (Slotine & Li, 1991)

$$F_k = \frac{\partial f}{\partial x} |_{\hat{x}_k|k, u_k} \quad k = 0, 1, 2 \ldots n \quad (2.24)$$

$$H_{k+1} = \frac{\partial h}{\partial x} |_{\hat{x}_{k+1}|k'} \quad (2.25)$$
Equations (2.24) and (2.25) linearize the nonlinear system and its measurement function in regard to current state, respectively. Then, the system goes through its predicting stage, which is defined as

\[ \hat{x}_{k+1|k} = f(\hat{x}_k|k, u_k) \quad (2.26) \]

\[ P_{k+1|k} = F_k P_k|k F_k^T + Q_k \quad (2.27) \]

where \( f \) is the nonlinear system model, \( \hat{x}_{k+1|k} \) and \( P_{k+1|k} \) are state estimate and corresponding state error covariance matrix, respectively. After these two parameters are obtained, there are few more parameters need to be calculated as

\[ \hat{y}_{k+1} = z_{k+1} - h(\hat{x}_{k+1|k}) \quad (2.28) \]

\[ S_{k+1} = H_{k+1} P_{k+1|k} H_{k+1}^T + R_{k+1} \quad (2.29) \]

where \( \hat{y}_{k+1} \) is the measurement error, which is calculated from system’s nonlinear measurement model \( h \). Variable \( S_{k+1} \) represents the measurement error covariance, which was calculated by using the linearized system matrix and its predicted corresponding matrix. After that, the system goes to its updating stage, which may be defined as

\[ K_{k+1} = P_{k+1|k} H_{k+1}^T S_{k+1}^{-1} \quad (2.30) \]

\[ \hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1} \hat{y}_{k+1} \quad (2.31) \]
where $K_{k+1}$ is the Kalman gain, and $\hat{x}_{k+1|k+1}$ is the updated estimates. Finally, the state error covariance $P_{k+1|k+1}$ can be calculated as

$$P_{k+1|k+1} = [I - K_{k+1}H_{k+1}]P_{k+1|k}$$

(2.32)

The extended Kalman filter shares similar structure to the classic Kalman filter, where the main difference is the additional linearization of the nonlinear system to make the filter work, therefore, it has become a common method to estimate nonlinear systems.

Wu and Sun (2010) used extended Kalman filter to detect and track in an unknown environment, based on their paper, it can be observed that the linearization process took a large amount of work to make the extended Kalman filter applicable. It is, therefore, not an optimized method to use, as if they had used unscented Kalman filter, the calculations they might have had should be less than what they did have since unscented Kalman filter does not require linearization.

### 2.2.3 Unscented Kalman Filter

The unscented Kalman filter was introduced by Julier et al. (2000). The unscented Kalman filter uses an unscented transform, which is basically a deterministic sampling technique. In Figure 2.4, $2n+1$ sampling points, where $n$ is the number of measurements of interest, are weighted and then transmitted through the nonlinear functions, creating an approximate solution to the mean and covariance of the desired estimates (Gadsden & Habibi, 2013). This approach yields similar results to the extended Kalman filter, however, one of the best advantages of this filter is that it does not require linearization of the nonlinear system for the filter to work. Therefore, this approach is more
feasible in multiple conditions. As this thesis mentioned in Section 2.1.2, the disadvantage of linearization is that it cannot take large initial error, specifically in Kalman filter, if a system model is extremely complicated, unscented Kalman filter may outperformed extended Kalman filter since it does not require linearization of the nonlinear system. The unscented Kalman filter has been widely used in many area and research (e.g. Attari et al., 2017; Gadsden et al., 2012)

Figure 2.3. Distribution of sigma point set for unscented Kalman filter in 2D space (From Haykin, 2001)
The unscented Kalman filter would be defined as follows (Julier & Uhlmann, 2004). Firstly, multiple sampling points named sigma points are generated. Then, \(2n+1\) sigma points are used to approximate \(n\)-dimensional random variable \(x_k\) with mean \(\hat{x}_{k|k}\) and covariance \(P_{k|k}\). The initial sigma point can be calculated as

\[
X_{0,k|k} = \hat{x}_{k|k} \tag{2.33}
\]

\[
W_0 = \frac{\kappa}{n + \kappa} \tag{2.34}
\]

where \(k=0, 1, 2\ldots n\), and \(X_{0,k|k}\) and \(W_0\) are the first corresponding sigma point and weight, respectively. Parameter \(n\) is the number of measurements and \(\kappa\) is a design value, is usually significantly less than 1. Then, sigma points 2 to \(n+1\) may be presented as

\[
X_{i,k|k} = \hat{x}_{k|k} + (\sqrt{(n + \kappa)P_{k|k}})_i \tag{2.35}
\]

\[
W_i = \frac{1}{2(n + \kappa)} \tag{2.36}
\]

where \(X_{i,k|k}\) and \(W_i\) are the corresponding sigma point and weight for \(i\)-th sigma point. The final \(n+2\) to \(2n+1\) sigma points may be calculated as

\[
X_{i+n,k|k} = \hat{x}_{k|k} - (\sqrt{(n + \kappa)P_{k|k}})_i \tag{2.37}
\]

\[
W_{i+n} = \frac{1}{2(n + \kappa)} \tag{2.38}
\]
where \( \left( \sqrt{(n + \kappa)P_{k|k}} \right)_i \) is the square root of \((n + \kappa)P_{k|k}\) of the \(i\)-th row or column of the matrix, and \(W_i\) is the weight associated with \(i\)-th sigma point (Julier et al., 2000). The state estimate \(\hat{x}_{k+1|k}\) can be predicted by propagating these sigma points through the system model along with its corresponding weight, which are shown as

\[
\hat{x}_{i,k+1|k} = f(\hat{x}_{i,k|k}, u_k) \quad (2.39)
\]
\[
\hat{x}_{k+1|k} = \sum_{i=0}^{2n} W_i \hat{x}_{i,k+1|k} \quad (2.40)
\]

From (2.39) and (2.40), the predicted state error covariance \(P_{k+1|k}\) is obtained through

\[
P_{k+1|k} = \sum_{i=0}^{2n} W_i (\hat{x}_{i,k+1|k} - \hat{x}_{k+1|k}) (\hat{x}_{i,k+1|k} - \hat{x}_{k+1|k})^T. \quad (2.41)
\]

These sigma points are then processed through the nonlinear measurement model to obtain the predicted measurement, which is defined as

\[
\hat{z}_{i,k+1|k} = h(\hat{x}_{i,k+1|k}, u_k) \quad (2.42)
\]
\[
\hat{z}_{k+1|k} = \sum_{i=0}^{2n} W_i \hat{z}_{i,k+1|k} \quad (2.43)
\]

where \(\hat{z}_{i,k+1|k}\) is the \(i\)-th measurement and \(\hat{z}_{k+1|k}\) is the predicted measurement. Then, the measurement covariance \(P_{zz,k+1|k}\), and the cross-covariance \(P_{xz,k+1|k}\) are calculated as
\[
P_{zz,k+1|k} = \sum_{i=0}^{2n} W_i (\hat{\theta}_{i,k+1|k} - \dot{z}_{k+1|k}) (\hat{\theta}_{i,k+1|k} - \dot{z}_{k+1|k})^T \quad (2.44)
\]

\[
P_{xz,k+1|k} = \sum_{i=0}^{2n} W_i (\hat{x}_{i,k+1|k} - \hat{\theta}_{k+1|k}) (\hat{x}_{i,k+1|k} - \hat{\theta}_{k+1|k})^T \quad (2.45)
\]

Furthermore, the Kalman gain, \( K_{k+1} \), may be calculated and then used to update the state estimate and state error covariance based on (2.44) and (2.45)

\[
K_{k+1} = P_{xz,k+1|k}^{-1} P_{zz,k+1|k} \quad (2.46)
\]

\[
\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}(z_{k+1} - \dot{z}_{k+1|k}) \quad (2.47)
\]

\[
P_{k+1|k+1} = P_{k+1|k} - K_{k+1} P_{zz,k+1|k} K_{k+1}^T \quad (2.48)
\]

Rajabi and Hossein (2018) proposed a controller based on sliding mode control using unscented Kalman filter. They used their controller for the trajectory tracking of a ball screw driven shake table. Although the results by using unscented Kalman filter did not have many differences when compared to the results using extended Kalman filter, the unscented Kalman filter did not require linearization, which means unscented Kalman filter can give more accurate results if a larger tracking error occurs, which they did not show in their paper.

2.3 Modeling of Nonholonomic Mobile Robots

The precision of the localization system of the mobile robot in the inertial frame is the essential requirement for the robot tracking system to work. In addition, the accuracy of the
measurements is equally important for the control system to be able to adjust itself. Current research focuses on improving the efficiency of the control strategies without considering the effects of the uncertain noises. Therefore, this thesis focusing on making an improvement on backstepping control in uncertain noises.

2.3.1 Dynamic Model of Mobile Robots

In a two-dimensional Cartesian workspace, the localization of the mobile robot can be described in inertia frame of the coordinates $X$ and $Y$, and the center point of the mobile robot $C$.

Figure 2.4. is a typical dynamic model of a nonholonomic mobile robot. The $\{X, O, Y\}$ is the inertia frame, and $\{D, C, L\}$ is the body-fixed frame. Variables $D$ and $L$ are the driving and lateral direction respectively, $C$ is the center of the mobile robot, $P = [X_c, Y_c, \theta_c]^T$ denotes the
current robot state and orientation angle. Although there are three degrees of freedom for the mobile robot, because the mobile robot is a nonholonomic system, the robot is subjected to a kinematic constraint, which limits its dynamic movement, and is defined as

\[ \dot{y}_c \cos \theta_c - \dot{x}_c \sin \theta_c = 0. \] (2.49)

The velocity between inertial frame, \( \dot{P}_c = [\dot{x}_c, \dot{y}_c, \dot{\theta}_c]^T \), and body fixed frame \( v = [v_c, \omega_c]^T \) is obtained through a Jacobian matrix as

\[
\begin{bmatrix}
\dot{x}_c \\
\dot{y}_c \\
\dot{\theta}_c
\end{bmatrix} = \dot{P}_c = \begin{bmatrix}
cos \theta_c & 0 \\
\sin \theta_c & 0 \\
0 & 1
\end{bmatrix} \begin{bmatrix}
v_c \\
\omega_c
\end{bmatrix}.
\] (2.50)

where \( v_c \) and \( \omega_c \) are the linear velocity and angular velocity of the mobile robot at its current state. The velocities are obtained through two independent driving wheels. The angular velocities for two driving wheels are expressed as

\[
\omega_L = \frac{1}{R} \left( v_c - \frac{D_\omega}{2} \omega_c \right)
\] (2.51)

\[
\omega_R = \frac{1}{R} \left( v_c + \frac{D_\omega}{2} \omega_c \right)
\] (2.52)

where \( \omega_L \) and \( \omega_R \) are the angular velocities for left and right driving wheels, respectively. Parameter \( D_\omega \) is the azimuth length between two wheels. Then, the linear velocity \( v_c \) and angular velocity \( \omega_c \) can be derived from (2.51) and (2.52) as

\[
v_c = \frac{v_L + v_R}{2}
\] (2.53)
\[ \omega_c = \frac{v_L + v_R}{D\omega}. \] (2.54)

where \( v_L \) and \( v_R \) are the linear velocities for left and right driving wheels, respectively. Equations (2.49) to (2.54) show the dynamics of the nonholonomic mobile robot. This type of mobile robot can be driven by two motors or using a gearbox to control the differences between the velocity of two driving wheels. Additionally, this robot has been wildly used in many applications, such as ground vehicles for distributing parts in factories and unmanned cleaning robots. There are many different studies for this type of robot, which include obstacle avoiding problems and trajectory tracking.

2.3.2 Tracking Error and Error Dynamics

The nonholonomic mobile robot’s desired state can be obtained by a reference path in the inertia frame as \( P_d(t) = [x_d(t), y_d(t), \theta_d(t)]^T \). Then, the tracking error can be defined as follows \( E(t) = [e_X(t), e_Y(t), e_\theta(t)]^T \), where \([e_X, e_Y]\) denotes the tracking errors in \( X \) and \( Y \) axis in the body-fixed frame, respectively, and \( e_\theta(t) \) represents the tracking error in orientation. Furthermore, an equivalent tracking error in body fixed frame can be calculated through a transformation matrix as

\[
E = \begin{bmatrix} e_X \\ e_Y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta_c & \sin\theta_c & 0 \\ -\sin\theta_c & \cos\theta_c & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e_X \\ e_Y \\ e_\theta \end{bmatrix} \] (2.55)
where \( e_x = x_d - x_c \), \( e_y = y_d - y_c \), and \( e_\theta = \theta_d - \theta_c \). These three values represent the errors in the inertia frame and the error dynamics, which can be seen in Figure 3.2. Additionally, the error dynamics is derived from (2.55) as

\[
\begin{bmatrix}
\dot{e}_x \\
\dot{e}_y \\
\dot{e}_\theta
\end{bmatrix} = 
\begin{bmatrix}
\omega_c e_y - u_c + v_d \cos e_\theta \\
-\omega_c e_x + v_d \sin e_\theta \\
\omega_d - \omega_c
\end{bmatrix}.
\] (2.56)

### 2.3.3 Tracking Control Objectives

The basic goal of the controller, as discussed in this thesis, is to complete the robot navigation tasks by building a relationship between observed information and velocity commands. The controller can be designed through the following processes. Assuming the desired robot state is known and defined as \( P_d(t) = [x_d(t), y_d(t), \theta_d(t)]^T \). Then, \( v_d \) and \( \omega_d \) are calculated as

\[
v_d = \sqrt{\dot{x}_d^2 + \dot{y}_d^2} \] (2.57)

\[
\omega_d = \frac{\ddot{y}_d \dot{x}_d - \ddot{x}_d \dot{y}_d}{\dot{x}_d^2 + \dot{y}_d^2} \] (2.58)

where \( v_d \) and \( \omega_d \) are the desired linear and angular velocity, respectively.

Knowing \( v_d \) and \( \omega_d \), a control law is then can be designed with linear and angular velocity, \( u_c \) and \( \omega_c \), based on desired state \( P_d(t) \). This control law, which is discussed in Chapter
3, drives the robot from actual state, $P_c(t) = [x_c(t), y_c(t), \theta_c(t)]^T$, to track its desired state.

As time approaches infinity, the tracking error is eventually reduced to zero, which can be expressed as

$$\lim_{t \to \infty} P_c(t) - P_d(t) = 0$$  \hspace{1cm} (2.59)

where $P_c(t)$ and $P_d(t)$ are the current state and desired state of mobile robot at time $t$. Equations (2.55) to (2.59) show how to model the nonholonomic mobile robot and calculate the error between desired state and current state. Then, this system is used in Chapter 4 to perform the trajectory control.

2.4 Summary

There are many different types of control techniques and estimation strategies available currently, all of which have pros and cons. Backstepping method is one of the easiest tracking techniques compared to others, although it suffers from large speed jump and overshoot issues. Additionally, unscented Kalman filter is relatively easy to apply and more compatible to control techniques as well. Therefore, backstepping tracking control and unscented Kalman filter are chosen in this thesis to track a desired trajectory with uncertain noises being presented for mobile robots. The existing speed jumps and overshoot issues in typical backstepping tracking controls, which is solved in the literature and this thesis with an application of bioinspired neural dynamics, by replacing the term that causes the speed jump and overshoot problems in backstepping control. In addition, the unscented Kalman filter gives extra robustness to the backstepping control strategy.
and provides accurate estimates. The results are shown and discussed in the following sections, which meets the intended expectations for this thesis.

Figure 2.5. Error dynamics of a nonholonomic mobile robot
Chapter 3

The Proposed Bioinspired Controller with Unscented Kalman Filter

This chapter proposes a novel tracking control based on backstepping technique and unscented Kalman filter. In order to avoid the speed jump and overshoot issues from the backstepping technique, an existing bioinspired neural dynamics method called shunting model is introduced to the control systems. In addition, to prevent the effects of the system and measurement noises, the unscented Kalman filter is brought into the system. The unscented Kalman filter is supposed to provide accurate estimates without being heavily affected by the noises.

3.1 The Bioinspired Controller with Unscented Kalman Filter

The unscented Kalman filter is integrated with the existing bioinspired backstepping controller to form a novel controller that can avoid speed jump and overshoot issues and provide accurate estimates. The entire system’s set up is shown in Figure 3.1. The desired posture of the mobile robot is assumed to be known. In addition, the measured posture of the mobile robot is also assumed to be known, however, the measured posture of the mobile robot is subjected with system and measurement noises, which are considered as zero-mean Gaussian.
The error in the inertia frame is obtained by \( P_d - \hat{P}_c \), which goes to the transformation matrix to calculate the body fixed frame error vector \( E_p \). After that, \( E_p \) and desired linear and angular velocity, \( v_d \) and \( \omega_d \), are fed into the path tracker to generate the speed commands, \( v_c \) and \( \omega_c \), for the nonholonomic mobile robot. Due to the external disruptions, Gaussian system and measurement noises are then added to the system and go through the unscented Kalman filter to filter the noises, obtaining an accurate estimate, therefore, this bioinspired backstepping controller with unscented Kalman filter should be more applicable in real world situations. Additionally, there are no perfectly accurate sensors existing in the world, as each sensor has certain tolerances due to the imperfections in manufacturing processes. Therefore, it is crucial to implement unscented Kalman filter into the bioinspired backstepping control in order to obtain accurate estimates.

### 3.2 Bioinspired Backstepping Controller

This section discusses the design procedure of the bioinspired backstepping controller. The bioinspired controller is able to generate smooth velocity commands without any velocity jump and large overshoot. This bioinspired backstepping controller has been used on a nonholonomic
mobile robot, unmanned underwater robot, and unmanned aerial robot (e.g. Sun et al., 2014; Wang, 2015; Yang et al., 2012).

3.2.1 Backstepping Control

The typical backstepping control law from a three degrees of freedom nonholonomic system is presented as

\[
v_c = C_1 e_x + v_d \cos \theta \\
\omega_c = \omega_d + C_2 v_d e_y + C_3 v_d \sin \theta
\]  

(3.1) \hspace{3cm} (3.2)

where \( C_1, C_2, \) and \( C_3 \) are the parameters, \( e_x, e_y \) and \( e_\theta \) are the errors in driving and lateral directions and error in tracking angle, respectively. It can be concluded from the observation of many papers (e.g. Fierro & Lewis, 1995; Tsai & Wang, 2004; Q. Zhang, et al., 1999), that the sudden velocity overshoot and jump from backstepping technique is caused by the sudden change in tracking error. The reason why speed jump and overshoot issues happen is due to the \( C_1 e_x \) term in (3.1), where \( C_1 \) is a positive constant, and \( e_x \) is the tracking error in driving direction. If a tracking error in driving direction occurs, the linear velocity \( v_c \) will be increased by \( C_1 e_x \) amount in a short period of time. However, the set up of the robot might not be able to make the mobile robots reach the desired acceleration due to the lack of the motor power. The larger the tracking error is, the higher the speed jump and overshoot which occur. Therefore, \( C_1 e_x \) is the fundamental problem why the backstepping technique has had difficulty tracking a desired trajectory for mobile robots. It is, therefore, one’s priority to eliminate the \( C_1 e_x \) term.
3.2.2 Bioinspired Neural Dynamics

The backstepping controller is able to track a desired trajectory alone. However, there are still limitations of this method, as it cannot take large tracking errors, which may lead to speed jump and overshoot that create discontinuities in robotic systems. Furthermore, the backstepping control may completely fail in certain situations, such as speed jumps at beginning, due to the infinitely torque requirement at initial stage. The shunting model can resolve the speed jump and overshoot problems in backstepping controllers, therefore, this thesis integrates the shunting model with backstepping controller to form a bioinspired backstepping controller.

Grossberg (1988) was the first person that developed a bioinspired neural dynamics model, which illustrates the adaptive behavior of individuals. This neural dynamic model was inspired from the membrane model, which was proposed by Hodgkin and Huxley (1990) for a patch membrane using electrical elements. The dynamics of the voltage across the membrane is described as

\[ C_m \frac{dV_m}{dt} = -(E_p + V_m)g_P + (E_{Na} - V_m)g_{Na} - (E_k + V_m)g_K \]  

(3.3)

where \( C_m \) is the membrane capacitance, the parameters \( E_p, E_{Na} \) and \( E_k \) are the Nernst potentials for Potassium and sodium ions, and the passive leak current in the membrane, respectively. The parameters \( g_P, g_{Na} \) and \( g_K \) are respectively the conductance of the potassium, sodium, and passive channels.
This neural dynamic model was then modified by Öğmen and Gagné (1990). They modified this membrane model by assuming $C_m = 1$ and then substituting $x_i = E_p + V_m$, $A = g_p$, $B = E_{Na} + E_p$, $D = E_k - E_p$, $S_i^+ = g_{Na}$, and $S_i^- = g_K$, then a shunting equation is derived as follows

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)S_i^+(t) - (D + x_i)S_i^-(t) \quad (3.4)$$

where $x_i$ is the neural activity of the $i$-th neuron, $A$ is the passive decay rate, and $B$ and $D$ are the upper and lower bounds of the neural activity, respectively. The variable $S_i^+$ is the excitatory input and $S_i^-$ is the inhibitory input, respectively. This bioinspired model has many applications in control theory (e.g. Ni, et al., 2017; Zhu, et al., 2015).

This shunting model can solve the speed jump and overshoot issues that occur in backstepping technique. Therefore, $e_D$ from backstepping technique from (3.1) is replaced by $u_s$, which is a function derived from shunting neural models. The dynamic behavior $u_s$ should make the control command be a smooth function with respect to position errors. In addition, the output of the shunting model changes smoothly if there is a sharp change of the input. The shunting model’s passive decay rate is able to adjust the system’s transient response, and the output from the shunting model is constrained to a bounded interval $[B, -D]$.

Thus, the tracking control law for the linear velocity $v_c$ and angular velocity $\omega_c$ becomes

$$v_c = u_s + v_d \cos \theta \quad (3.5)$$
\[ \omega_c = \omega_d + C_2 v_d \gamma + C_3 v_d \sin \theta \]  

(3.6)

where \( C_2 \) and \( C_3 \) are positive parameters and \( v_s \) is obtained through the shunting model in driving direction, which is written as

\[
\frac{dv_s}{dt} = -Av_s + (B - v_s)\hat{f}(e_X) - (D + v_s)\hat{g}(e_X) \tag{3.7}
\]

where \( A \) is the passive decay rate, \( B \) and \( D \) are upper and lower bounds of the velocity, respectively. In addition, the linear above threshold function is defined as

\[
\hat{f}(e_X) = \begin{cases} 
  e_X & \text{if } e_D > 0 \\
  0 & \text{if } e_D \leq 0
\end{cases} \tag{3.8}
\]

and the inhibitory nonlinear input is defined as

\[
\hat{g}(e_X) = \begin{cases} 
  -e_X & \text{if } e_D < 0 \\
  0 & \text{if } e_D \geq 0
\end{cases} \tag{3.9}
\]

The variable \( v_s \) can be derived from (3.7) to (3.9) and it is written as

\[
v_s(t + 1) = v_s(t) + \frac{dv_s}{dt} \Delta t. \tag{3.10}
\]

For the shunting model, \( A \) effects the convergent rate. As \( A \) gets larger, the convergent rate is faster. In addition, \( B \) and \( D \) represent the upper and lower bound of the neural activity, the smaller the \( B \) and \( D \) set, the faster the convergent rate is. Based on the time response of the first order system, the desired velocity of the nonholonomic robot is given as
where \( v_0 \) and \( \tau \) are the desired velocity and time constant. This shows the mobile robot should increase its velocity until it reaches its desired value exponentially.

\[
v_d(t) = v_0 \left(1 - e^{-\frac{t}{\tau}}\right)
\]

3.3 Bioinspired Controller with Unscented Kalman Filter

The existing bioinspired backstepping controller alone may be sufficient in an ideal situation, however, in real environments the controller might operate in noisy environments and even be equipped with faulty devices. The original design of the bioinspired backstepping controller did not consider the effect of uncertain noises on the controller. It is, therefore, important to introduce estimation strategy into the control theory. The innovation of this thesis is that the existing bioinspired controller is combined with unscented Kalman filter, which should give extra robustness to the controller. The unscented Kalman filter is a nonlinear filter, which is a powerful tool of providing accurate estimates regardless of system and measurement noises. In this particular case, one may choose to apply unscented Kalman filter in the controller based on two factors. First, it is more compatible with most control techniques, second, it does not require linearization of the system matrix, which is a hard task for backstepping control.

This section designs an unscented Kalman filter, which is suitable for this bioinspired backstepping controller. Firstly, the noises that affect tracking control are system noise and
measurement noise, which are treated as Gaussian. The system noise is $P(\lambda_k) \sim N(0, Q_k)$ and, and measurement noise is $P(\mu_k) \sim N(0, R_k)$. Then the system may be defined as

\[
\hat{P}_k = f(P_k, e_k) + \lambda_k \quad (3.13)
\]

\[
\hat{Z}_k = h(\hat{P}_k) + \mu_k \quad (3.14)
\]

where $k = 0, 1, 2, \ldots, n$, $\lambda_k$ and $\mu_k$ are the system and measurement noises at $k$-th time iteration, respectively, and both noises are considered as Gaussian; $e_k$ is the tracking error from the desired state to actual state, which is $E(t) = [e_x(t), e_y(t), e_\theta(t)]^T$; $f(P_k, e_k)$ is the actual posture, $P_c = [x_c, y_c, \theta_c]^T$, of the mobile robot; $f$ is the shunting model based backstepping controller, which is defined through (3.5) to (3.10); and $\hat{P}_k$ and $\hat{Z}_k$ are the actual and measurement states with noises, respectively. Then, these values are propagated through the unscented Kalman filter, which is defined through (2.33) to (2.48), to obtain its updated state estimate, state error covariance, and Kalman gain. The coordinates and the rotating angle for the mobile robot in inertial frame is assumed to be known and they are calculated by a real-time multisensory fusion by a robot localization algorithm. Because there are three measurements $x_c, y_c$, and $\theta_c$ from the systems, the number of sigma points generated from unscented Kalman is seven, and these points are distributed in a 3D Cartesian workspace. Then, these seven sigma points along with their weight functions go through the unscented Kalman filter and generate optimal results.

By using unscented Kalman filter, this bioinspired backstepping controller is supposed to provide accurate estimates with system and measurement noises being presented in the bioinspired
backstepping controller. Therefore, the proposed controller is more suitable in real world applications compared to just using a controller without any filter.

3.4 Stability Analysis

As mentioned in Section 2.14, the stability of the backstepping technique can be guaranteed by Lyapunov stability theory. This controller is stable based on Lyapunov theory and the tracking errors will eventually reduce to zero. In addition, function \((B - v_s)f(e_X)\) in shunting model makes the output of the shunting model stays below its upper bound \(B\), whereas function \((D + v_s)g(e_X)\) let the output of the shunting model stay above its lower bound \(-D\). As a result, the output of the shunting model stays between a bounded interval \([B, -D]\).

To prove the system is stable, a Lyapunov candidate function is selected as

\[
V(t) = \frac{1}{2}(e_X^2 + e_Y^2) + \frac{1}{C_2}(1 - \cos e_\theta) + \frac{1}{2B}v_s^2 \tag{3.15}
\]

where \(v_s\) is defined in (3.7). Therefore, one can find that \(\frac{1}{2}(e_X^2 + e_Y^2), \frac{1}{C_2}(1 - \cos e_\theta)\) and \(\frac{1}{2B}v_s^2\) is greater than zero. The \(V(t)\) equals to zero only if \(e_X, e_Y, e_\theta\), and \(v_s\) are all equal to zero. Then, based on (3.15) and (3.5) to (3.7), the Lyapunov candidate function are taken to a time derivative, which becomes

\[
\dot{V}(t) = \dot{e}_X e_X + \dot{e}_Y e_Y + \frac{1}{C_2}\dot{e}_\theta \sin e_\theta + \frac{1}{B}v_s v_s. \tag{3.16}
\]

This function is then converted to
\( \dot{V}(t) = (\omega_c e_Y - v_c + v_d \cos \theta) e_X + (-\omega_c e_X + v_d \sin \theta) e_Y + \frac{1}{C_2} (\omega_d - \omega_c) \)

\[ + \frac{1}{B} [-Av_s + (B - v_s)f(e_X) - (D + v_s)g(e_X)]. \tag{3.17} \]

Finally, the time derivative of this function is written as

\[ \dot{V}(t) = -v_s e_X - \frac{C_3}{C_2} v_d \sin^2 \theta e_X + \frac{1}{B} [-A - f(e_X) - g(e_X)]v_s^2 + \frac{1}{B} [Bf(e_X) - Dg(e_X)]v_s. \tag{3.18} \]

From (3.18), choose \( B = D \), then it is rewritten as

\[ \dot{V}(t) = \frac{C_3}{C_2} v_d \sin^2 \theta e_X + \frac{1}{B} [-A - f(e_X) - g(e_X)]v_s^2 + [f(e_X) - g(e_X)]v_s. \tag{3.19} \]

Additionally, based on (3.8) and (3.9), it indicates that if \( e_X \geq 0 \), then \( f(e_X) = e_X \) and \( g(e_X) = 0 \). It is, therefore, one can find that

\[ [f(e_X) - g(e_X) - e_X]v_s = [e_X - 0 - e_X] = 0. \tag{3.20} \]

In the other case, if \( e_X \leq 0 \), then \( g(e_X) = -e_X \) and \( f(e_X) = 0 \). Thus, one can calculate that

\[ [f(e_X) - g(e_X) - e_X]v_s = [0 - (-e_X) - e_X] = 0. \tag{3.21} \]

Based on (3.17), (3.18), and (3.15) is rewritten as
\[
\dot{V}(t) = -\frac{C_2}{C_2} v_d \sin^2 e_\theta + \frac{1}{B} \left[-A - f(e_X) - g(e_X)\right] v_s^2.
\] 

(3.22)

Parameter \(C_2, C_3\), and desired forward velocity are all positive constants, then the first part
\[-\frac{C_2}{C_2} v_d \sin^2 e_\theta\] from (3.19) is smaller than or equal to zero. In addition, from (3.8) and (3.9), \(f(e_X)\) and \(g(e_X)\) are always nonnegative constants, variables \(A\) and \(B\) are always greater than or equal to zero. Thus, one can conclude that the second part of (3.19), \(\frac{1}{B} \left[-A - f(e_X) - g(e_X)\right] v_s^2\), is smaller than or equal to zero. In conclusion, the derivative of the bioinspired backstepping controller can be expressed as

\[
\dot{V}(t) \leq 0.
\] 

(3.23)

Therefore, the tracking controller is stable, and it is proven by the Lyapunov stability theory. To further prove the system is asymptotically stable, there are additional procedures which need to be calculated. It can be concluded that when the time approaches to infinity, \(v_c \to v_d\) and \(\omega_c \to \omega_d\). Additionally, assuming \(v_d > 0\), it can be concluded that \(\dot{V}(t) \leq 0\) and parameters \(E_p, v_s\) are constrained. After that, based on (3.8) and (3.20), \(||E_p||, ||\dot{E}_p||\), and \(||v_s||\) are bounded. Therefore, one can find that \(||\ddot{V}(t)|| < \infty\) because \(V(t)\) is converging to a constant value. From Barbalat’s lemma, based on \(\dot{V}(t) \to 0\) as time approaches to infinity, it can be inferred that \(v_s \to 0\) as time approaches to infinity. Additionally, from (3.7), it can be calculated from shunting model, \(e_D\) converges to zero as \(v_s\) converges to zero. From Equation (3.19), one can find that as time goes to infinity \(e_\theta \to 0\). Furthermore, from (3.8) and (3.6), it can be concluded that \(\omega_c - \omega_d = 0\) as \(e_\theta \to 0\).
0, and error for the angular velocity converges to zero as time goes to infinity, therefore, \( C_2 v_d e_r \) = 0. If \( v_d > 0 \), we have \( e_r \to 0 \) as time goes to infinity, then \( E_p \) is at its equilibrium point \( E_p = 0 \). The entire system is then proven to be asymptotically stable. The error guarantees to converge to zero as time goes to infinity, which is written in the following form

\[
\lim_{t\to\infty} (X_d(t) - X_c(t)) = 0.
\] (3.24)

The unscented Kalman filter is a well-known filter and its stability has been proven in many papers (e.g. Hu et al., 2015; Xu et al., 2008), to show the unscented Kalman filter is stable and bounded proven. There are two main steps need to be done, an instrumental diagonal matrix and an extra positive definite matrix are brought into the system, then using these two matrices into the unscented Kalman filter to show the estimation error of the Kalman filter is bounded. Therefore, the design of bioinspired controller with unscented Kalman filter is stable and bounded.

3.5 Summary

This section integrated the unscented Kalman filter into the bioinspired backstepping controller for trajectory tracking control on a nonholonomic mobile robot in a two-dimensional workspace. The controller is supposed to generate smooth velocity commands itself. The innovative contribution of this thesis is that the unscented Kalman filter provides extra robustness of the controller. If the nonholonomic mobile robot has system and measurement noises present, the bioinspired backstepping controller will fail to provide accurate estimates, which are crucial for the researchers to do further analysis. Therefore, the proposed bioinspired backstepping
controller with unscented Kalman filter is capable of providing accurate estimates and tracking the desired trajectory without any large velocity overshoot or velocity jump.
Chapter 4

Simulation Results

In this section, the proposed controller is applied to track straight, circular and elliptical paths. The results are then compared and discussed to show the advantages of the proposed controller.

4.1 Tracking Straight Paths

The proposed controller is used to track a simple line, $y = 4$, in the Cartesian workspace. The starting state for the mobile robot is $(0, 2.5, 0)$, then, the initial error is $(0, 1.5, 0)$. The total running time is 10s and sampling time is 0.01s, respectively. The model parameters are selected as follows: $C_1 = 1$, $C_2 = 4$, $C_3 = 3$, $A = 4$, $B = 2$, $D = 2$. The desired linear and angular velocity are set as $v_d = 1$ and $\omega_d = 0$, respectively. The linear velocity $v_d$ increases exponentially at initial state, which is defined in (4.11), the parameters are chosen as $\kappa = 1$ and $\tau = 0.5$. The system and measurement noises described by $w_k$ and $v_k$ are zero-mean Gaussian with a variance of $Q = 10^{-6}$ and $R = 10^{-4}$, respectively.

4.1.1 Straight Path Tracking without Noises

This subsection does multiple comparison studies between typical backstepping controller and bioinspired backstepping controller to discuss the efficiencies in multiple aspects between
these two controllers in terms of trajectory tracking.

Based on Figure 4.1, both backstepping control and bioinspired backstepping control are able to track the desired trajectory. The speed jump and overshoot issues of the backstepping technique are the fundamentals issue that roots in backstepping control. By observing Figure 4.1(c), the speed overshoot problem has been largely reduced. In addition, two more studies are implemented with the same parameters’ settings. The trajectory is changed to $y = 5$ and $y = 6$, with the same initial starting point $(0,2.5)$, which makes the initial errors become $(0,2.5)$ and $(0,3.5)$. The results for different straight path tracking are shown in Figures 4.2 and 4.3., which inferred that the speed overshoot has been greatly reduced by using bioinspired backstepping controller.
Figure 4.1. Comparison of straight path ($y=4$) tracking. (a) Position; (b) Angular Velocity; (c) Linear velocity.
The results prove that even with larger initial tracking error, the speed overshoot of the bioinspired backstepping controller remains at a low level, whereas the backstepping controller reaches an unreasonable extreme velocity. This extreme velocity might create discontinuities for mobile robot because the mobile robot cannot reach such acceleration or extreme velocity. In
addition, based on the observation of different results from Table 1, one can find that the speed overshoots and initial tracking errors have linear relations, such that if the tracking error increase by 1, speed overshoot in backstepping control increases around 0.7 and 0.14 for the bioinspired controller. This happens because the shunting model constrains the acceleration of the controller, it is, therefore, restricting the speed overshoot.

Table 4.1. Speed overshoot comparison between backstepping and bioinspired controllers

<table>
<thead>
<tr>
<th></th>
<th>( y = 4 )</th>
<th>( y = 5 )</th>
<th>( y = 6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backstepping</td>
<td>1.461</td>
<td>2.156</td>
<td>2.971</td>
</tr>
<tr>
<td>Bioinspired</td>
<td>1.287</td>
<td>1.425</td>
<td>1.563</td>
</tr>
</tbody>
</table>

Overall, the speed overshoot issues in backstepping controller, which may cause discontinuities for the robotic system, has been largely reduced. This speed overshoot reduction would be quite useful in real applications, because mobile robots are always being subjected to a limited amount of acceleration or largest speed, which sometimes make backstepping control not practicable. These results for the shunting model applied to backstepping controller meet the initial expectations.
Figure 4.3. Comparison of straight path ($y=6$) tracking. (a) Position; (b) Angular Velocity; (c) Linear velocity.
4.1.2 Straight Path Tracking with Unscented Kalman Filter

This subsection discusses the solutions for nonholonomic mobile robot to track a straight path in a situation where it is affected by system and measurement noises. In ideal environments, the proposed bioinspired controller might be able to track the desired trajectory, however, in noisy environments with imperfect sensors, there might be noises, which will heavily affect the accuracy of the trajectory tracking. To compare the effects of the system and measurement noises on the proposed bioinspired controller, the initial system and measurement noises are set to zero-mean Gaussian with a variance of $Q=10^{-6}$ and $R=10^{-4}$, respectively, and these noises are added to the control system. The results without applying unscented Kalman filter to the system can be seen from Figure 4.4.
Figure 4.4. Comparison of desired state and actual measurement. (a) Position; (b) Angular velocity; (c) Linear velocity.
Figure 4.5. Comparison of straight path tracking with/without filter. (a) Position; (b) Angular velocity; (c) Linear velocity.
It is obvious that with the measurement and system noises being presented in the system, all the measurements taken from the mobile robot are way off compared to its desired measurements. Therefore, one can conclude that the existing bioinspired backstepping controller is not capable of trajectory tracking in a situation, which system and measurement noises are applied to the mobile robot system. The unscented Kalman filter provides reliable solutions to filter the system and measurement noises, therefore accurate estimates could be provided in return. Thus, after applying the unscented Kalman filter to the proposed bioinspired controller, the results for straight line path tracking can be observed from Figure 4.5. The red lines show that although there are still minor bounces between the desired path, the actual trajectory follows the desired path overall. The system and measurement noises have been greatly reduced. To show the effectiveness of the filter for reducing the measurement noise, two more simulations are conducted to demonstrate the power of the unscented Kalman filter in control. Zero-mean Gaussian measurement noise with variation of $R = 2 \times 10^{-4}$ and $R = 1 \times 10^{-3}$ are conducted, and the results are shown in Figure 4.5 and Figure 4.6.
Figure 4.6. Measurement with variance of $10^{-3}$ measurement noise. (a) Position; (b) Angular velocity; (c) Linear velocity.
Figure 4.5 and Figure 4.6 show that no matter how big the measurement noise is, the measurements are still accurate with the application of unscented Kalman filter, which is crucial for trajectory tracking problems because accurate estimates can allow people do the correct adjustments on the system and not get confused from the data they collect. For example, if a mobile robot lands on an alien planet, however, due to an unpredicted problem during landing procedures, the equipment is damaged, which creates large amount of measurement noise, with the application of unscented Kalman filter, the bioinspired controller with unscented Kalman filter would still provide reliable estimates that could potentially save millions of dollars in relaunching a mobile robot on an alien planet. The bioinspired backstepping controller with unscented Kalman filter, therefore, is more applicable in real situations.

4.2 Tracking Circular Paths

To show the proposed controller can track a curved path, a desired circular path \( x^2 + y^2 = 1 \) is used in the study. The desired robot path starts from an initial state of \((1,0)\), and the actual robot starts from \((0,-0.8)\). The desired forward and angular velocity are set to 1 and 1. The forward velocity \(v_d\) is increased exponentially at the initial state, and the parameters are chosen as \(\kappa = 1\) and \(\tau = 0.5\). The controller’s parameters are \(A = 4\), \(B = 3\), \(D = 2\), \(C_1 = 1\), \(C_2 = 2\), \(C_3 = 4\), sampling time is 0.01s and total running time is 10s.
4.2.1 Circular Path Tracking without Noises

This subsection performs comparison studies between typical backstepping controller and bioinspired backstepping controller in terms of tracking a circular path. There is a significant improvement for the proposed bioinspired backstepping technique compared to the typical backstepping technique. Figure 4.8(a) shows that both control strategies are able to track the desired circular path. However, the initial starting velocity for typical backstepping control starts from 1, which can be observed from Figure 4.8(b). This indicates that the required torque to drive the robot is infinity large at the starting stage, because only an infinite torque can make the robot reach an acceleration, which makes the speed of the mobile robot reach 1 instantly. Therefore, backstepping control is not feasible even in an ideal situation to track a circular trajectory.

In order to further prove the proposed bioinspired backstepping controller can avoid the infinite initial torque problem in typical backstepping controller, two more simulations are conducted, which the results are shown in Figure 4.7 and 4.8, to show the initial infinite torque requirement to drive the mobile robot is an unavoidable issue in typical backstepping control unless the starting linear velocity is 0, which happens in linear path tracking. In addition, these two simulations prove that proposed bioinspired strategy is capable of dealing with the infinite torque issue in situations with different desired trajectories. With different desired trajectory and initial state for the mobile robot, the proposed bioinspired controller is still able to track the desired path. In addition, one can observe that the initial linear velocity required for backstepping control is directly related to the error in the driving direction. For example, if the initial tracking errors are 1.5 and 2 in the driving direction, then the required initial linear velocities are $1.5C_1$ and $2C_1$, 

\[ \text{61} \]
respectively. The reason why this issue happens can be traced back to (4.5), the $C_1 e_X$ term decides the amount of the initial velocity required for the typical backstepping controller. However, in bioinspired controller, $C_1 e_X$ term is replaced by shunting model $u_s$, thus, the infinite torque issue does not happen. Furthermore, the initial velocity is changed to 1.5 instead of 1 with larger curved trajectory, this is because the larger trajectory needs the robot to reach a certain linear velocity to be able to catch the desired state at each iteration.

The speed overshoot for typical backstepping controller is not as much in linear path tracking as in circular path tracking, which is because larger initial velocity makes the mobile robot catch the desired trajectory faster. However, this is not realistic due to the infinite torque issue as mentioned before. Overall, the proposed bioinspired controller still has less speed overshoot compare to typical backstepping controller, although the results are not as obvious as linear trajectory tracking.
Figure 4.7. Comparison of circular path \((x^2+y^2=1)\) tracking. (a) Position; (b) Angular velocity; (c) Linear velocity.
Figure 4.8. Comparison of circular path (\(x^2+y^2=2.25\)) tracking. (a) Position; (b) Angular velocity; (c) Linear velocity.
Figure 4.9. Circular path tracking with different initial state \((-0.5,1,0)\). (a) Position; (b) Angular velocity; (c) Linear velocity.
4.2.2 Circular Path Tracking with Unscented Kalman Filter

This subsection compares the bioinspired controller in terms of tracking a circular path with system and measurement noises. As mentioned, these noises can be costly and therefore, the effects of these noises need to be minimized. Several simulations are conducted to show the efficiency of the unscented Kalman filter in circular trajectory tracking using bioinspired controller. Based on the results from Figure 4.10, with an application of $Q = 10^{-4}$ measurement noise and $R = 10^{-6}$ system noise, the bioinspired controller is not able to provide accurate estimates due to the disturbances from the noises. With the application of the unscented Kalman filter, it can be observed that there is a significant improvement in estimation of circular trajectory tracking. In a complicated situation, the bioinspired controller without unscented Kalman filter may not be able to track the desired circular trajectory.

In an extremely noisy area, the reasonable amount of system and measurement noises that robots might take can be considered as zero-mean Gaussian noises with variance of $2 \times 10^{-4}$ and $10^{-6}$, respectively. The results are shown in Figure 4.11., it indicates that the under an extreme case the proposed bioinspired controller with unscented Kalman filter is still able to provide accurate estimates. Therefore, there is no doubt that the unscented Kalman filter gives extra robustness to the system along with existing bioinspired backstepping controller.
Figure 4.10. Comparison of desired state and actual measurement. (a) Position; (b) Angular velocity; (c) Linear velocity.
Figure 4.11. Measurement with variance of $2 \times 10^{-4}$ measurement noise. (a) Position; (b) Angular velocity; (c) Linear velocity.
4.3 Tracking Elliptical Paths

The proposed bioinspired backstepping controller with unscented Kalman filter is used to track a desired elliptical path, and the results are shown in Figure 4.12 and 4.13. These figures indicate that the proposed controller is not able to track the desired elliptical paths, because the velocity should be constant as the system approaches stability (Fierro & Lewis, 1997). However, based on (2.57) and (2.58) the desired velocity command with no tracking errors for elliptical paths is oscillatory, therefore, the requirement for constant desired velocity commands in conventional backstepping controllers is not met.

By observing Figure 4.12 and 4.13, the angular velocity has two peaks every two seconds whereas the velocity command for elliptical path without tracking error is a sine function, which only has one peak every two seconds. The reason why this happens can be traced back to (3.2). The angular velocity command has two terms, which are $C_2v_\theta Y$ and $C_3v_\theta \sin \theta$. Both terms are tuning together, which causes the angular velocity to have two peaks instead of one. In addition, although both the linear and angular velocity commands eventually approach stability, the generated angular velocity does not converge to the angular velocity command with no tracking error. Therefore, the mobile robot is still able to track an elliptical path, but inaccurately. To further prove the bioinspired backstepping controller is able to track a desired elliptical trajectory inaccurately, another simulation is performed to track a different elliptical path ($x^2+1.5y^2=2.25$). It shows the fundamental issue for bioinspired backstepping controller is that it cannot guarantee the linear and angular velocity be a constant when the system reaches stability.
Figure 4.12. Elliptical path \((x^2+1.5y^2=2.25)\) tracking. (a) Position; (b) Angular velocity; (c) Linear velocity.
In order to analyze why the bioinspired backstepping controller is able to track the desired elliptical paths despite being inaccurate, Figure 4.14 shows the error between the measured parameters and desired parameters. It can be seen from Figure 4.14(a) that the position error for both methods are bounded. In addition, the errors between measured angular and linear velocities
are oscillated due to the $v_d \cos \theta$ and $C_3 v_d \sin \theta$ terms. It can be inferred that the tracking errors in angular and linear velocities are heavily influenced by these two terms as the linear and angular velocity tracking errors act as sine and cosine functions, respectively.

4.4 Summary and Discussion

Multiple simulations are conducted to demonstrate the effectiveness and efficiency of the bioinspired backstepping controller with unscented Kalman filter. The traditional backstepping controller has speed overshoot issues, where in certain cases, the speed overshoot issue turns to infinite starting torque problems, which makes this approach not feasible. Therefore, the shunting model was introduced and the backstepping controller was modified into the bioinspired controller, which can largely solve the issues that traditional backstepping control is facing. In addition, an unscented Kalman filter is brought into the system to repel the effects of the system and measurement noises in system estimations.

Overall, the proposed bioinspired controller with unscented Kalman filter can be used in many applications, such as trajectory tracking in an unknown dangerous environment or for mobile robots on an alien planet. In addition, one can conclude that the proposed controller achieved the goals that were expected, which are tracking a desired path and providing accurate estimates, based on the simulation results.
Figure 4.14. Tracking errors for elliptical path. (a) Linear Velocity (b) Angular velocity
Chapter 5

Conclusions and Future Works

5.1 Conclusions

In this thesis, a backstepping technique is firstly introduced. Then, the bioinspired neural dynamics model was presented to deal with the speed jumps and overshoot issues in backstepping technique. Finally, an unscented Kalman filter is brought into the system to minimize the system and measurement noises, which can potentially cause the system to provide inaccurate estimates. Both shunting model and unscented Kalman filter are successfully integrated in the backstepping controller. The existing problems in backstepping controller are rooted in its \( C_1 e_x \) term in its forward velocity function, as this term causes speed overshoot and unrealistic velocity jump problems. Therefore, the bioinspired neural dynamics method is brought in to deal with the existing problems. In addition, the zero-mean Gaussian noises were added into the system to simulate the effects of the system and measurement noises, which makes the mobile robot face a more complicated environment. The unscented Kalman filter is used to eliminate the noises.

Multiple simulations were performed to guarantee the effectiveness and efficiency of the proposed controller results, and these results are satisfactory. The proposed controller generates a smooth and continuous velocity command and handles the existing speed jump problems in backstepping technique. In addition, the system and measurement noises are filtered by unscented
Kalman filter, which gives extra robustness to the controller.

There are still many improvements could potentially be applied to the controller, such as using smooth variable structure filter, which would provide reliable estimates regardless of the system model’s change. Furthermore, to increase the robustness of the proposed controller, an adaptive term could potentially be developed to enhance the efficiency of the proposed controller when dealing with uncertainties. In addition, the proposed controller in this thesis could be tested on a real mobile robot in the future prove its applicability in real world environments.

5.2 Future Works

Although the proposed bioinspired controller with unscented Kalman filter has made a great improvement in current trajectory tracking problems, there are a couple more concerns raised during the simulations that have been conducted. When system noise increases, the actual path of the mobile robot tends to become unstable and the unscented Kalman filter is eventually going to fail. The results are shown in Figure 5.1. As the system and measurement noises were increased to $10^{-4}$, the mobile robot tends to follow its own model. Therefore, the filter is not applicable in this situation.
Figure 5.1. Straight path tracking with variance of $10^{-4}$ system and measurement noises. (a) Position; (b) Angular velocity; (c) Linear velocity.
Therefore, the limitations of this proposed controller are that the system noise should not be large, otherwise the entire system will fail because the proposed controller is then not able to track the path accurately. This happens due to the limitations of the unscented Kalman filter, which may be solved by adding an adaptive term into the proposed controller to deal with the added noise in the system. In addition, there is another filter called smooth variable structure filter that was first proposed by Habibi (2007), a professor from McMaster University. The advantage of smooth variable structure filter compared to unscented Kalman filter is that it is quite robust to system model changes. For example, if there is an uncertainty that changes the weight of the mobile robot, the unscented Kalman filter will fail whereas smooth variable structure filter will still provide accurate estimates.
REFERENCES


Approach to Tracking Control of Mobile Robots, 59(8), 3211–3220.


