DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MODELING AND MOTION CONTROL OF A WHEELED MOBILE ROBOT USING MULTIPLE SENSORS

by

Halit Ege CEYHUN

July, 2023

İZMİR

MODELING AND MOTION CONTROL OF A WHEELED MOBILE ROBOT USING MULTIPLE SENSORS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for Master of Science in
Department of Mechatronics Engineering, Mechatronics Engineering Program

by

Halit Ege CEYHUN

July, 2023

İZMİR

THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"MODELING AND MOTION CONTROL OF A WHEELED MOBILE ROBOT USING MULTIPLE SENSORS"** completed by **HALİT EGE CEYHUN** under supervision of **ASSOC.PROF.DR. AYTAÇ GÖREN** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

.....

Assoc.Prof.Dr. Aytaç GÖREN

Supervisor

.....

Prof.Dr. Zeki KIRAL

Jury Member

.....

Assoc.Prof.Dr. S. Murat BAĞDATLI

Jury Member

Prof. Dr. Okan FISTIKOĞLU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENT

I would like to thank deeply my thesis advisor Assoc. Prof. Dr. Aytaç GÖREN who guided me during my thesis work and also helped me with the career decision making process.

I would like to thank my family who provided their full support along any point of my education.

Halit Ege CEYHUN

MODELING AND MOTION CONTROL OF A WHEELED MOBILE ROBOT USING MULTIPLE SENSORS

ABSTRACT

This thesis work tackles a well-known problem of trajectory tracking of wheeled mobile robots (WMR) from different perspectives. Kinematic bicycle model is taken as the actor of the study. This model is firstly introduced to the reader with its constraints and controllability conditions. Then, tracking controllers are considered. These controllers are Linear Quadratic Regulator (LQR), a Lyapunov-based controller and Dynamic Feedback Linearization with non-interacting control property. Controllers are applied to the kinematic bicycle model differently (error dynamics, extended coordinates). Moreover, obstacle avoidance is considered and Pure Pursuit Algorithm is employed as a geometric obstacle avoidance mechanism. Similarly, Pure Pursuit Algorithm is utilized for local controllers to approach the desired trajectory when initialized away. Obtained hybrid controllers are then exposed to measurements with noise. By utilizing an Extended Kalman Filter (EKF), performances are evaluated. The results are then compared to a Landmark-based measurement fusion. Finally, these hybrid controllers are simulated for 3 different cases to find the best.

Keywords: Linear Quadratic Regulator, Wheeled Mobile Robot, Kinematic Bicycle Model, Lyapunov, Dynamic Feedback Linearization, Pure Pursuit Algorithm, Trajectory Tracking, Obstacle Avoidance, Extended Kalman Filter, Hybrid Control

ÇOK ALGILAYICILI TEKERLEKLİ ROBOTUN MODELLEMESİ VE HAREKET KONTROLÜ

ÖZ

Bu tez çalışması iyi bilinen bir problem olan tekerlekli robotların güzergah izlemesini farklı perspektiflerden yaklaşarak konu almaktadır. Kinematik bisiklet modeli bu çalışmanın aktörü olarak ele alınmıştır. Bu model ilk olarak okuyucuya kısıtlar ve kontrol edilebilirlik şartlarıyla birlikte sunulmuştur. Devamında takip kontrolcüleri ele alınmıştır. Bu kontrolcüler Linear Quadratic Regulator (LQR), bir Lyapunov temelli kontrolcü ve etkileşimsiz özellikli dinamik geribeslemeli doğrusallaştırmadır. Kontrolcüler kinematik bisiklet modele farklı şekillerde uygulanmıştır(hata dinamikleri, genişletilmiş koordinatlar). Bununla birlikte, engellerden kaçınma göz önünde bulundurulmuş ve Pure Pursuit Algoritması geometrik bir engelden kaçınma mekanizması olarak burada devreye alınmıştır. Buna benzer olarak, Pure Pursuit Algoritması lokal kontrolcüler için istenen güzergatan uzakta başlatıldığında yaklaşma için de kullanılmıştır. Elde edilen hibrit kontrolcüler ölçüm gürültüsüne maruz bırakılmıştır. Extended Kalman Filtresi (EKF) kullanılarak performanslar değerlendirilmiştir. Devamında konuma bağlı ölçüm füzyonu dahil edilerek sonuçlar karşılaştırılmıştır. Sonuç olarak, bu hibrit kontrolcüler 3 farklı senaryo için simüle edilmiş ve en iyi sonucu veren araştırılmıştır.

Anahtar Kelimeler: Linear Quadratic Regulator, Tekerlekli Robot, Kinematik Bisiklet Modeli, Lyapunov, Dinamik Geribeslemeli Doğrusallaştırma, Pure Pursuit Algoritması, Güzergah İzleme, Engellerden Kaçınma, Extended Kalman Filtresi, Hibrit Kontrol

CONTENTS

Page

THESIS EXAMINATION RESULT FORM ii
ACKNOWLEDGMENT iii
ABSTRACT iv
ÖZ v
CONTENTS vi
LIST OF FIGURES ix
LIST OF TABLES xiii

CHAPTER ONE INTRODUCTION 1

1.1 State of the Art 1
1.2 Defining the Problem 2
1.3 Aim of This Study 3

CHAPTER TWO KINEMATIC MODELING 4

2.1 Modeling 4
2.2 Kinematic Bicycle Model 4

CHAPTER THREE CONTROL APPROACHES 11

3.1 Control of Kinematic Bicycle Model 11
3.2 Reference Generation 11
3.3 Local Control Approaches 15
3.3.1 Coordinate Transformation and Expression of Error System 15
3.3.2 Linear Quadratic Regulator 18
3.3.3 Lyapunov Theory-Based Control 21
3.3 Global Control Approach 24
3.4.1 Dynamic Input-Output Feedback Linearizing Control 24
3.5 Geometric Path Tracking Approach 47
3.5.1 Pure Pursuit Algorithm 47

CHAPTER FOUR SENSORS AND FILTERING 52

4.1 Sensors and Measurement 52
4.1.1 State Measurements 52
4.1.2 External Measurements 54
4.2 Filtering 55
4.2.1 Sinusoidal Noise Canceling 56
4.2.2 Statistical Filtering 56

CHAPTER FIVE SIMULATIONS AND RESULTS 65

5.1 LQR Trajectory Tracking 65
5.2 Lyapunov-Based Trajectory Tracking 70
5.3 Dynamic Feedback Linearizing Control 72
5.4 Pure Pursuit Tracking 78
5.5 Pure Pursuit Obstacle Avoidance 80
5.6 Tracking Controller Performances Under Noisy Measurement and Filtering 83
5.7 Hybrid Motion Control of Kinematic Bicycle Model 95
5.7.1 Case 1: Model Starting Away from Trajectory and Avoiding a Static Point Obstacle 95
5.7.2 Case 2: Model Starting Away from Trajectory and Avoiding Moving Point Obstacles 100
5.7.3 Case 3: Model Starting on a Linear Trajectory and Avoiding Moving Point Obstacles Under Noisy Measurement 104

CHAPTER SIX CONCLUSION 113

APPENDICES 114

Appendix 1: LQR Control Scheme 114
Appendix 2: Lyapunov-Based Control Scheme 115
Appendix 3: Dynamic Feedback Linearizing Control Scheme 116
Appendix 4: Pure Pursuit Algorithm Block Scheme 117

Appendix 5: Lyapunov-Based Technique with Extended Kalman Filter Scheme 118
Appendix 6: LQR Technique with Extended Kalman Filter Scheme 119
Appendix 7: Dynamic Feedback Linearizing Control with Extended Kalman Filter Scheme 120
Appendix 8: Lyapunov-Based Hybrid Control Scheme 121
Appendix 9: LQR Hybrid Control Scheme 122
Appendix 10: Dynamic Feedback Linearizing Hybrid Control Scheme 123

REFERENCES 124

LIST OF FIGURES

Page

Figure 2.1 Kinematic bicycle model coordinates 6

Figure 3.1 Kinematic bicycle model moving along an arc 13

Figure 3.2 Control scheme explanation 24

Figure 3.3 Extended system creation 29

Figure 3.4 Creation of extended system for second step 33

Figure 3.5 Two chains of integrators 46

Figure 3.6 Pure Pursuit Algorithm scheme 48

Figure 3.7 Pure Pursuit Algorithm-based control law 51

Figure 4.1 Measurement representation 55

Figure 4.2 State observer block scheme 57

Figure 4.3 Discrete time estimation scheme for Kalman Filtering 60

Figure 4.4 Probability distributions 61

Figure 4.5 Probability distribution functions with additional measurement 62

Figure 5.1 Tracking performance of LQR technique 68

Figure 5.2 Tracking error of LQR technique on both cartesian coordinates 69

Figure 5.3 Tracking performance of Lyapunov-based technique 70

Figure 5.4 Tracking error of Lyapunov-based technique on both cartesian coordinates 71

Figure 5.5 Tracking performance of dynamic feedback linearization in both coordinate systems 74

Figure 5.6 Tracking error of dynamic feedback linearization on both coordinates ... 75

Figure 5.7 Convergence and tracking performance ... 77

Figure 5.8 Pure Pursuit Algorithm tracking lookahead distance 2 m ... 79

Figure 5.9 Effect of lookahead distance selection ... 80

Figure 5.10 Dangerous zone defined by an obstacle ... 82

Figure 5.11 Additive measurement noise on feedback arm ... 84

Figure 5.12 Tracking performance of LQR and Lyapunov-based controllers under noisy measurement and Extended Kalman Filter application ... 85

Figure 5.13 Measured and estimated states of Lyapunov-based Controller ... 87

Figure 5.14 Measured and estimated states of LQR controller ... 87

Figure 5.15 Tracking performance of LQR and Lyapunov-based controllers under noisy measurement and Extended Kalman Filter application with landmark-based localization ... 89

Figure 5.16 Measured and estimated cartesian coordinates of Lyapunov-based controller ... 90

Figure 5.17 Measured and estimated cartesian coordinates of LQR controller ... 90

Figure 5.18 Tracking performance under noisy measurement with Extended Kalman Filter ... 91

Figure 5.19 Measured and estimated states of dynamic feedback linearizing controller ... 92

Figure 5.20 Measured and estimated cartesian coordinates of dynamic feedback linearizing controller with landmark-based localization ... 93

Figure 5.21 Tracking performance of dynamic feedback linearizing controller under noisy measurement and landmark-based localization with Extended Kalman Filter ... 94

Figure 5.22 Tracking and obstacle avoidance comparison of hybrid controllers 98

Figure 5.23 Multiple moving obstacle environment decision making subsystem 101

Figure 5.24 Tracking and obstacle avoidance comparison of hybrid controllers in dynamic environment 102

Figure 5.25 Obstacle avoidance and tracking performances comparison under noisy measurement and Extended Kalman Filter 106

Figure 5.26 Filter performance of cartesian coordinates of dynamic feedback linearizing control approach 108

Figure 5.27 Cartesian coordinates estimation error over time for dynamic feedback linearizing control 108

Figure 5.28 Obstacle avoidance and tracking performances comparison under noisy measurement and Extended Kalman Filter with landmark-based localization 109

Figure 5.29 Filter performance of cartesian coordinates of dynamic feedback linearizing control approach for EKF+landmarks scenario 111

Figure 5.30 Cartesian coordinates estimation error over time for dynamic feedback linearizing control for EKF+landmarks scenario 112

Figure A.1 LQR control scheme Simulink screen 114

Figure A.2 Lyapunov-based control scheme Simulink screen 115

Figure A.3 Dynamic feedback linearizing control scheme Simulink screen 116

Figure A.4 Pure Pursuit Algorithm Simulink screen 117

Figure A.5 Lyapunov-based technique with Extended Kalman Filter Simulink screen 118

Figure A.6 LQR technique with Extended Kalman Filter Simulink screen 119

Figure A.7 Dynamic feedback linearizing control with Extended Kalman Filter Simulink screen 120

Figure A.8 Lyapunov-based hybrid control with Pure Pursuit Algorithm Simulink screen..... 121

Figure A.9 LQR hybrid control with Pure Pursuit Algorithm Simulink screen 122

Figure A.10 Dynamic feedback linearizing hybrid control with Pure Pursuit Algorithm Simulink screen 123

LIST OF TABLES

Page

Table 5.1 Performance Evaluation of LQR Technique 69

Table 5.2 Performance Evaluation of Lyapunov-Based Technique 71

Table 5.3 Performance Evaluation of Dynamic Feedback Linearizing Control 74

Table 5.4 Effect of R Value Selection 76

Table 5.5 Comparison of Performance Under Noise and Extended Kalman Filter 86

Table 5.6 Comparison of performance under noise and Extended Kalman Filter with landmark-based localization 89

Table 5.7 Effect of performance for landmark-based localization 93

Table 5.8 Comparison of performance for hybrid controllers 99

Table 5.9 Comparison of performance for hybrid controllers in dynamic environment 103

Table 5.10 Tracking comparison and obstacle avoidance performance of case 3 with EKF 107

Table 5.11 Tracking comparison and obstacle avoidance performance for case 3 with EKF+landmarks 110

CHAPTER ONE
INTRODUCTION

Autonomous navigation of wheeled mobile robots and vehicles is one of the emerging technology fields in today's world. Thanks to the rapidly growing and evolving interest on this research area, technologic advancements are achieved easily in a competitive manner. These considerations give a good amount of materials for the interested individuals, investors or industries in both companies and academia. Many subtopics of autonomous navigation can be a research focus. Some of them can be given as, trajectory tracking, setpoint regulation, sensor measurement, sensor fusion and data filtering, energy optimization, perception, etc. However, one subtopic stands before others. Safety considerations are very important for these technologies and therefore decide if a designed system is deployable or not. Moreover, designing an autonomously navigating robotic or vehicle system is a challenging and highly responsible task. In this work, motion control of a kinematic model is discussed and a realistic, deployable control technique is searched to accomplish a trajectory tracking task.

1.1 State of the Art

Wheeled mobile robots can be divided into subfamilies of kinematic models. Unicycles and bicycle kinematic models are well known families. Bicycle kinematic models are also divided into subcategories by reference point selection. These models can also represent conventional vehicles. Therefore, a model selection and studying control techniques on them, gives an insight on a scalable result. However, dynamic models are also widely employed with many parameters to be identified.

For control techniques, there are also subcategories and different approaches. Depending on model selection, longitudinal motion controllers, lateral motion controllers, geometric controllers can be designed. Some of the mainly applied techniques can be listed as linear controllers, nonlinear controllers, Model Predictive

Control (MPC), Stanley Controller, Pure Pursuit Algorithm, cost defined optimal controllers, Lyapunov-based controllers, etc. All of them have advantageous and disadvantageous sides when compared.

In terms of sensing, many sensors and transducers are used for measuring position, speed, acceleration, orientation of the subject itself. On the other hand, range and bearing measurements are also used for the tasks. Some of the attached sensors are encoders, GPS, Inertial Measurement Units (IMUs), LIDARs, SONARs, cameras.

As another aspect, filtering of measurements is vital for trustworthiness. Reliable measurements allow engineers to come up with acceptable control techniques for autonomous navigation and motion control of robots or vehicles. Mainly filtering processes can be divided into two as linear and nonlinear filtering. Some of the examples are considered as moving average filtering, bandpass filters, Kalman Filters, Particle Filters, Extended/Unscented Kalman Filters.

So, there are many aspects which are already considered and studied. Therefore, it is necessary to do a deep research and problem to be worked on must be well defined. To be more clear, definition of the problem and aim of this study is detailed in next sections.

1.2 Defining the Problem

As stated above, trajectory tracking is a highly sensitive and detailed mission. Autonomous wheeled mobile robots or vehicles can cause damage on environment and living creatures in case of undesired motions. Therefore, a co-existing space with humans, animals or other features must be well analyzed. Since human health is more important than any device, a hierarchy of importance is well defined. However, these high-level technology devices are expensive and as a result it is necessary to avoid any damage for the robot or vehicle as well. A preexisting trajectory in a known space must be designed by considering worst case scenarios. Regular trajectory of a city bus can be a good example for a designed trajectory. Even if the city bus is assumed to be using

one lane, it must follow a desired speed input, must stay in the lane and avoid any obstacles (pedestrians, other cars) with smallest possible deviation from center of the lane or by generating and following an alternative collision free path. Since all control techniques have different advantages and disadvantages, hybrid approach is a valid idea.

Under the light of previous statements, definition of the problem can be proposed as “performing trajectory tracking with minimum error and successful obstacle avoidance via a hybrid or a pure control technique”.

1.3 Aim of This Study

This study aims to pick kinematic bicycle model with rear wheel contact point reference as the model. Then, perform different control techniques to find their strengths and flaws. By this way it is possible to distinguish between possible fields of usage of control techniques. Next step is to compare these techniques under noisy measurements and come up with filtering operation. Moreover, a hybrid or pure technique is the desired output for this work. As a result, a realistic result is found and presented.

Simulations are presented mostly in a comparative way. All simulations are performed in MATLAB/Simulink environment.

CHAPTER TWO
KINEMATIC MODELING

2.1 Modeling

Kinematic modeling of bicycle type robots or vehicles is a widely investigated topic. These models are practical in case of low-speed, low-acceleration motions or preferred in cases where parametric unknowns of the subject are numerous. Reasons behind are evident and can be pointed out easily.

When high-speed moving robots or vehicles take a turn, they will encounter with a centrifugal force which can cause the subject to deviate from its original orientation. Besides, instantaneous high torque inputs to any wheel structure will create an additional motion where pure rolling condition of wheels are violated.

For detailed models with high-speed working conditions, it is possible to use dynamic models. However, as stated above, number of unknown parameters will decide if this model is employable or not. Even with a robust design for system dynamics, a high number of uncertain or unknown parameters might drive the system to instability. Contrary to this, kinematic bicycle models only need a single parameter for system design. This will be explored deeply in forthcoming works in this thesis.

2.2 Kinematic Bicycle Model

Kinematic bicycle model is representative of a large group of conventional vehicles and robots. In this work, statically balanced kinematic bicycle model is considered. Therefore, out of plane motion is not studied. To mathematically define a bicycle kinematic type robot, first step is to define wheels and their rolling constraints.

Types of wheels can be found in many resources of literature. For kinematic bicycle model there are 2 types of wheels. One steerable or orientable wheel at the front end

and another standard or fixed wheel at the rear end. As mentioned before, single parameter of this model is the wheelbase. Which is the length between two wheels' contact points to the ground.

These wheels are assumed to be only rolling. Therefore, only one direction of motion is allowed for wheels, others are constrained. To be more clear, wheel centers are moving in a single directional way and it is along wheel's orientation. Moreover, this can be considered as a planar vector. Since all structure is moving together, generalized coordinates of the model can be expressed by position of wheels and their orientation. Equation (2.1) shows the state vector of kinematic bicycle model.

$$s = \begin{bmatrix} x_{front} \\ y_{front} \\ x_{rear} \\ y_{rear} \\ \theta \\ \varphi \end{bmatrix} \tag{2.1}$$

Where θ is orientation angle of the structure and φ is the steering angle. However, 6-dimensional state vector s is redundant and can be expressed with less dimensions.

Since wheelbase is the single constant parameter of the structure, it is possible to rewrite both wheel contact points with a simple calculation. In this work, wheelbase is shown as l. Equations (2.2) and (2.3) show wheel coordinates relationship by wheelbase.

$$x_{front} = x_{rear} + l\cos(\theta) \tag{2.2}$$

$$y_{front} = y_{rear} + l\sin(\theta) \tag{2.3}$$

Figure 2.1 shows the dimensionally reduced version of bicycle kinematic model.

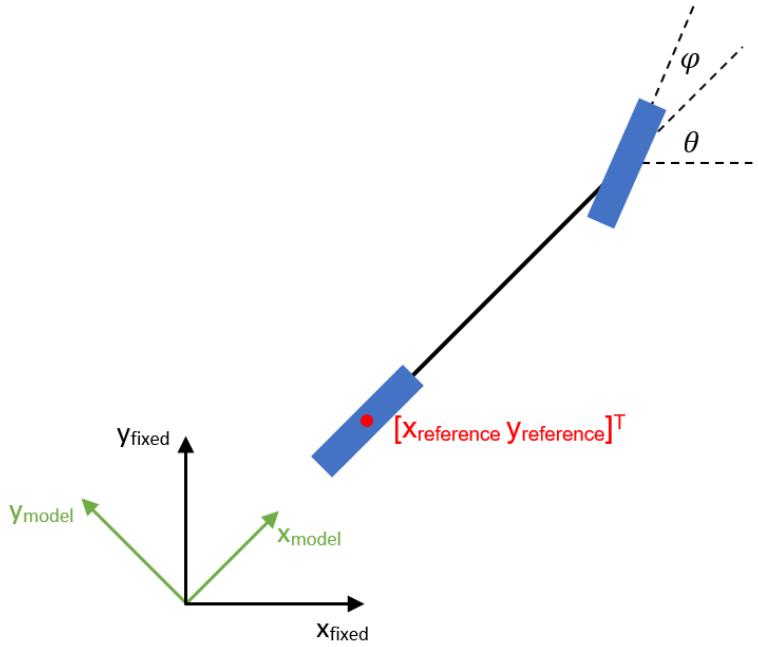

Figure 2.1 Kinematic bicycle model coordinates

As stated above, only one contact point is sufficient to express both wheels thanks to the constant wheelbase length. To express the motion of this structure, it is necessary to use reference frames. A fixed reference frame is considered as the inertial frame and it is not moving in time. Another reference frame is a rotating one and for this thesis work it is the rotated version of the fixed reference frame by the orientation angle of the kinematic bicycle model.

This work takes rear wheel as the reference point. Other reference point choices can be made. Different choice of reference points results different kinematic models. Some common reference point choices are rear wheel contact with the ground, front wheel contact with the ground, center of mass.

State vector now can be expressed with 4 dimensions. However, rolling constraints must be checked and non-holonomy condition must be investigated. Time derivation of wheel equations can be seen in Equations (2.4) and (2.5).

$$\dot{x}_{front} = \dot{x}_{rear} - l\dot{\theta}\sin\theta \tag{2.4}$$

$$\dot{y}_{front} = \dot{y}_{rear} + l\dot{\theta}\cos\theta \tag{2.5}$$

By this way rolling constraints can be expressed for both wheels and grouped in a matrix form. Following equations will use x and y coordinates for rear wheel contact point to the ground and subindex “rear” will be dropped.

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0 \tag{2.6}$$

$$\dot{x}_{front}\sin(\theta+\varphi) - \dot{y}_{front}\cos(\theta+\varphi) = 0 \tag{2.7}$$

By injecting Equations (2.4) and (2.5) to front wheel rolling constraint as shown in Equation (2.7), a meaningful form is obtained in Equation (2.8).

$$\left(\dot{x} - l\dot{\theta}\sin\theta\right)\sin(\theta+\varphi) - \left(\dot{y} + l\dot{\theta}\cos\theta\right)\cos(\theta+\varphi) = 0 \tag{2.8}$$

By reordering the terms and doing cancellations, a simpler expression is rewritten in Equation (2.9).

$$\dot{x}\sin(\theta+\varphi) - \dot{y}\cos(\theta+\varphi) - l\cos\varphi = 0 \tag{2.9}$$

So that both constraints can be grouped in matrix form as shown in Equation (2.10).

$$\begin{bmatrix} sin(\theta) & -cos(\theta) & 0 & 0 \\ sin(\theta+\varphi) & -cos(\theta+\varphi) & -lcos(\varphi) & 0 \end{bmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi} \end{pmatrix} = 0 \qquad (2.10)$$

This matrix expression is in Pfaffian form which suggests a kinematic constraint on the motion and gives the number of control vector fields to be determined. In this case 4-dimensional system has 2 constraints. Therefore, 2 input vector fields will be introduced. Note that this result also states the non-holonomy. Because solution of this kinematic constraint for null space and integration doesn't represent a geometric constraint. Hence, the model is non-holonomic and can reach any point of configuration space as long as it is free.

Linear speed of the rear wheel is taken as first input and angular velocity on steering angle is taken as the second input. Inputs are shown in Equations (2.11) and (2.12).

$$v = \sqrt{\dot{x}^2 + \dot{y}^2} \qquad (2.11)$$

$$w = \dot{\varphi} \qquad (2.12)$$

This choice satisfies the constraint matrix and therefore kinematic model can be written in its final form as stated in Equation (2.13).

$$\dot{s} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\varphi} \end{bmatrix} = \begin{pmatrix} cos(\theta) \\ sin(\theta) \\ tan(\varphi)/l \\ 0 \end{pmatrix} v + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} w \qquad (2.13)$$

Controllability conditions must be proven before moving on to the controller design and simulation part. This proof is also a double check of non-holonomy conditions. To do so, controllability of nonlinear systems is studied and results are evaluated.

A well-known approach is to create an accessibility distribution matrix with Lie brackets obtained by input vector fields. Since dimension of this model is 4, accessibility distribution matrix must have rank equal to 4 for this model to be controllable. Lie derivative operations can be seen in Equations (2.14) and (2.15).

$$[f_1, f_2] = \frac{\partial f_2}{\partial s} f_1 - \frac{\partial f_1}{\partial s} f_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ \tan(\varphi) \\ \hline l \\ 0 \end{pmatrix}$$
$$- \begin{pmatrix} 0 & 0 & -\sin(\theta) & 0 \\ 0 & 0 & cos(\theta) & 0 \\ 0 & 0 & 0 & \frac{1}{l \cos^2 \varphi} \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\frac{1}{l \cos^2 \varphi} \\ 0 \end{pmatrix} = f_3 \qquad (2.14)$$

Third vector field is found directly by using first two input vector fields. A fourth one is searched as in Equation (2.15) by including newly found vector field to the Lie derivation.

$$[f_1, f_3] = \frac{\partial f_3}{\partial s} f_1 - \frac{\partial f_1}{\partial s} f_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{2 \sin \varphi}{l \cos^3 \varphi} \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \cos(\theta) \\ \sin(\theta) \\ \tan(\varphi) \\ \hline l \\ 0 \end{pmatrix}$$
$$- \begin{pmatrix} 0 & 0 & -\sin(\theta) & 0 \\ 0 & 0 & cos(\theta) & 0 \\ 0 & 0 & 0 & \frac{1}{l \cos^2 \varphi} \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ -\frac{1}{l \cos^2 \varphi} \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{-\sin(\theta)}{l \cos^2 \varphi} \\ \frac{cos(\theta)}{l \cos^2 \varphi} \\ 0 \\ 0 \end{pmatrix} = f_4 \qquad (2.15)$$

Since 4 vectors are found, accessibility distribution matrix rank condition is investigated in Equation (2.16).

$$rank\begin{bmatrix} \cos(\theta) & 0 & 0 & \frac{-\sin(\theta)}{l\cos^2\varphi} \\ \sin(\theta) & 0 & 0 & \frac{cos(\theta)}{l\cos^2\varphi} \\ \frac{\tan(\varphi)}{l} & 0 & -\frac{1}{l\cos^2\varphi} & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = 4 \qquad (2.16)$$

Full rank condition is satisfied. Thus, it can be said that the system is controllable via these vector fields in 4-dimensional state space. Lie bracket operation guarantees reachability of the whole configuration space. Therefore, constraints are not global on this model. They are only constraints on instantaneous evolution of states in time. In forthcoming sections, different types of control techniques are studied and results are shared.

CHAPTER THREE
CONTROL APPROACHES

3.1 Control of Kinematic Bicycle Model

This section includes different control approaches to the kinematic bicycle model. This topic is a focus point of autonomous wheeled mobile robotics. Various techniques are applied to this model and can be seen in literature. However, all suggested control techniques have advantageous and disadvantageous sides. While some are insensitive to noise some others are better at tracking a desired input. In this section, two local approaches, one global approach and a geometrical tracking approach are presented. To accomplish a reference trajectory tracking task, generation of reference inputs must be presented.

3.2 Reference Generation

Reference state vector is necessary to define a trajectory containing information on reference inputs. By this way it is possible to define a reference for a conventional feedback scheme of a control system. Equation (3.1) shows the reference state vector for this task.

$$s_{reference} = \begin{pmatrix} x_{reference} \\ y_{reference} \\ \theta_{reference} \\ \varphi_{reference} \end{pmatrix} \tag{3.1}$$

Thanks to differential flatness and constant single parameter of the model, derivatives of the states and reference inputs can be expressed. Equations (3.2), (3.3), (3.4) and (3.5) show evolution of reference states in time.

$$\dot{x}_{reference} = v_{reference} \cos(\theta_{reference}) \tag{3.2}$$

$$\dot{y}_{reference} = v_{reference} \sin(\theta_{reference}) \tag{3.3}$$

$$\dot{\theta}_{reference} = \frac{v_{reference} tan(\varphi_{reference})}{l} \tag{3.4}$$

$$\dot{\varphi}_{reference} = w_{reference} \tag{3.5}$$

Unlike many other models, this derivation operation is not sufficient for reference generation because reference steering angle cannot be obtained directly by derivative of given x and y coordinates of the reference trajectory. However, linear velocity calculation can be made by deriving only given reference trajectory coordinates. This can be seen easily in Equation (3.6).

$$v_{reference} = \sqrt{\dot{x}_{reference}{}^2 + \dot{y}_{reference}{}^2} \tag{3.6}$$

Calculation of reference steering angle can be made through calculation of the curvature of the reference trajectory. Before giving the equation, Figure 3.1 is presented for clarification.

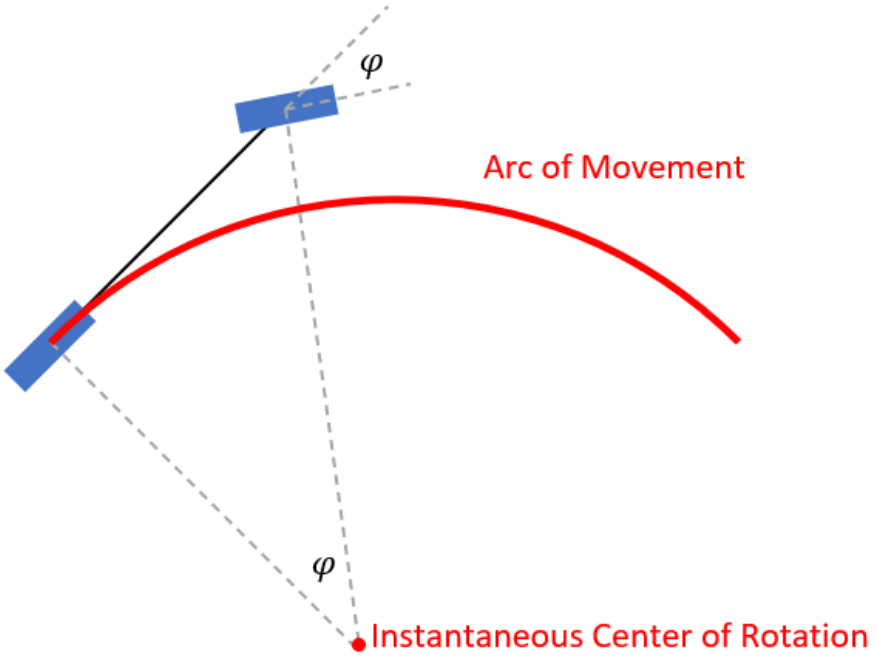

Figure 3.1 Kinematic bicycle model moving along an arc

In Figure 3.1 the steering angle is shown. This concept can be explained through kinematic bicycle model's geometry. To find this structure's instantaneous center of rotation, two lines must be drawn from centers of wheels. These lines are along perpendicular direction of wheel orientations. Instantaneous center of rotation of that time instant is at the meeting point of these two lines. Clearly, whole body is rotating around that singular point at that time instant because of previously explained constraint of wheel movement. Since wheels only move along their orientation, there cannot exist any movement along their perpendicular directions. Therefore, following any arc with a certain radius directly decides necessary steering angle.

Steering angle is a function of the rear wheel's distance to the instantaneous center of rotation. It is known that this distance is inverse of the curvature which calculated at the same point. Equation (3.7) shows calculation of curvature for any given trajectory by only using derivatives of cartesian coordinates of the same trajectory. Radius of the trajectory can be calculated by Equation (3.8).

$$\kappa_{reference} = \frac{\ddot{y}_{reference}\dot{x}_{reference} - \ddot{x}_{reference}\dot{y}_{reference}}{\left(\dot{x}_{reference}{}^2 + \dot{y}_{reference}{}^2\right)^{3/2}} \tag{3.7}$$

$$R_{reference} = \frac{1}{\kappa_{reference}} \tag{3.8}$$

Since curvature of the trajectory can be obtained by derivation of the trajectory coordinates directly, also distance between rear wheel and instantaneous center of rotation is obtained. Finally, the reference steering angle to follow an arc at a given time instant can be calculated by Equation (3.9).

$$\varphi_{reference} = \tan^{-1}\left(\frac{l}{R_{reference}}\right) \tag{3.9}$$

Hence, the derivation of reference steering angle will give the second input as the rate of change of steering angle. On the other hand, reference orientation for the given trajectory can be easily calculated by Equation (3.10).

$$\theta_{reference} = \tan^{-1}\left(\frac{\dot{y}_{reference}}{\dot{x}_{reference}}\right) \tag{3.10}$$

As a result, all of the reference states and their evolutions with respect to time are calculated. These results will be used in simulations as references.

3.3 Local Control Approaches

This part of the thesis work gives intuition on two different approaches for controlling kinematic bicycle model through focusing alternative dynamics of the system. Approaches are considered local because of the made assumptions.

3.3.1 Coordinate Transformation and Expression of Error System

Applying coordinate transformation for control methods is a common approach for many robotic and mechatronic systems. Given system for this work is highly nonlinear and can be considered hard to control. Direct linearization through Jacobian method is not representing the dynamics of this system. Therefore, feedback linearization and coordinate transformations are necessary to control this system. This results another problem to be solved. Linearized system will have different coordinates to be controlled. Different techniques result different new systems. A common approach is to use an error system to stabilize and get real inputs of the original system. In this work these coordinate transformations and realization of error systems are shown stepwise.

In order to express the error system, it is necessary to have measurement for each state or a good estimation through observers for unmeasured or noisy states. All states are considered available for this part. Error vector can be expressed as in Equation (3.11).

$$e = \begin{bmatrix} x_{reference} - x \\ y_{reference} - y \\ \theta_{reference} - \theta \\ \varphi_{reference} - \varphi \end{bmatrix} \tag{3.11}$$

Cartesian errors are expressed in rotated coordinate system. Rotation is done by orientation angle. However, this transformation is only applied to planar system. Steering angle must be kept the same since it is already a relative angle to the orientation angle. Equation (3.12) shows this transformation operation.

$$\begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{pmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 & 0 \\ -\sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x_{reference} - x \\ y_{reference} - y \\ \theta_{reference} - \theta \\ \varphi_{reference} - \varphi \end{pmatrix} \tag{3.12}$$

Since this new state vector is defined, its time derivative is also necessary to form a state space representation of the error system. Time derivatives of these new states can be seen in Equations (3.13), (3.14), (3.15) and (3.16).

$$\dot{e}_1 = v_{reference} cos(e_3) - v + \frac{v \tan(\varphi)}{l} e_2 \tag{3.13}$$

$$\dot{e}_2 = v_{reference} sin(e_3) - \frac{v \tan(\varphi)}{l} e_1 \tag{3.14}$$

$$\dot{e}_3 = \frac{v_{reference} tan(\varphi_{reference})}{l} - \frac{v \tan(\varphi)}{l} \tag{3.15}$$

$$\dot{e}_4 = \dot{\varphi}_{reference} - \dot{\varphi} \tag{3.16}$$

However, this system is still highly nonlinear. Following steps explain definition of artificial inputs for this system and other assumptions which are made for different control techniques.

Firstly, following artificial inputs in Equations (3.17), (3.18) and (3.19) are selected.

$$u_1 = v_{reference} cos(e_3) - v \tag{3.17}$$

$$u_2 = \dot{e}_3 \tag{3.18}$$

$$u_3 = \dot{e}_4 \tag{3.19}$$

Then small angle assumption is made and is shown in Equation (3.20), (3.21) and (3.22). It assumes equivalence of trigonometric functions while given angle is close to zero. This reasoning is meaningful around zero error for tracking.

$$\lim_{\varepsilon \to 0} \sin(\varepsilon) = \varepsilon \tag{3.20}$$

$$\lim_{\varepsilon \to 0} \cos(\varepsilon) = 1 \tag{3.21}$$

$$\lim_{\varepsilon \to 0} \tan(\varepsilon) = \varepsilon \tag{3.22}$$

So that system defined in previously shown new state vector can be rewritten in a matrix form. New form can be seen in Equation (3.23).

$$\begin{bmatrix}\dot{e}_1\\ \dot{e}_2\\ \dot{e}_3\\ \dot{e}_4\end{bmatrix} = \begin{bmatrix}0 & \frac{v_r tan(\varphi_r)}{l} & 0 & 0\\ -\frac{v_r tan(\varphi_r)}{l} & 0 & v_r & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}e_1\\ e_2\\ e_3\\ e_4\end{bmatrix} + \begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}u_1\\ u_2\\ u_3\end{bmatrix} \quad (3.23)$$

At this point, considerations for linear time invariant systems become important. Equation (3.23) shows a perfect form for separated system variables and inputs. However, terms inside drift matrix can be time dependent for reference trajectories. Reference speed and reference steering angle must be predetermined to have constant terms in this matrix. Doing this operation results a linear time invariant structure but these values of reference speed and reference steering angle must be constant for all time intervals. As a preliminary result, one can say that this approach will be successful for constant speed and constant steering angle trajectories with no stops. In next sections a circular trajectory is selected for simulations and tracking result are shown.

3.3.2 Linear Quadratic Regulator

Linear Quadratic Regulator (LQR) is one of the most popular optimal control techniques. Optimal control uses a quadratic cost function to find optimal control gains for a time invariant system. Standard formulation of LQR cost index for minimization can be seen in Equation (3.24). In this work, continuous time, infinite time horizon version of LQR is taken into account.

$$J = \int (x^T Q x + u^T R u) dt \quad (3.24)$$

In this formulation x and u are vectors of system states and inputs. These states and inputs are penalized separately by diagonal matrices of Q and R. For previously

expressed error system, this formulation can be shown as a function of new state vector. Equation (3.25) shows this formulation.

$$J = \int (e^T Q e + u^T R u) dt \tag{3.25}$$

As long as the reference velocity and reference steering angle values are constant, it is possible to utilize this formulation.

Form of Q and R matrices can be seen in Equations (3.26) and (3.27).

$$Q = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{bmatrix} \tag{3.26}$$

$$R = \begin{bmatrix} r_1 & 0 & 0 \\ 0 & r_2 & 0 \\ 0 & 0 & r_3 \end{bmatrix} \tag{3.27}$$

Where diagonal terms shown by q and r are weights of each state and input. These weights specify the importance of corresponding state or input. Where Q is symmetric positive semidefinite and R is symmetric positive definite.

Determination of corresponding control gain is the second step after deciding these weights. As stated above, LQR can find the optimal control gain for a linear time invariant system. Equation (3.28) shows the form of a linear time invariant system and Equation (3.29) shows the static feedback structure of a linear time invariant system.

$$\dot{x} = Ax + Bu \tag{3.28}$$

$$u = -Kx \tag{3.29}$$

The resulting system has eigenvalues determined by the new drift matrix defined by this feedback. Equation (3.30) shows new state space form.

$$\dot{x} = (A - BK)x \tag{3.30}$$

For newly found error system, this form takes the structure of Equation (3.31) for linear time invariance conditions.

$$\dot{e} = (A - BK)e \tag{3.31}$$

Determination of gain matrix K is done by solving Riccati Equation. This equation uses Q and R matrices to find the optimal gains and collects them in K matrix. Equations (3.32) and (3.33) show calculation of K matrix.

$$A^T P + PA - PBR^{-1}B^T P + Q = 0 \tag{3.32}$$

$$K = R^{-1}B^T P \tag{3.33}$$

As a result, a static gain matrix of K can be determined and stability analysis can be done for new drift matrix. This analysis is done in next sections.

On the other hand, this feedback action gives a feedback action in matrix form. Open form of Equation (3.29) in error system coordinates has three artificial inputs of

the error system. However, it is necessary to obtain original inputs for linear speed and rate of change of steering angle. This consideration is clarified in next sections.

3.3.3 Lyapunov Theory-Based Control

Lyapunov Theory is a nonlinear control approach which investigate the stability conditions of a given system. Unlike LQR technique, a linear time invariant structure is not necessary. However, a storage function is necessary and stability is defined by characteristics of the same function. In this work, Lyapunov's direct method is used and this method suggests conditions in Equations (3.34) and (3.35) for stability. These equations take x as states and V(x) as storage function.

$$V(x) \geq 0 \tag{3.34}$$

$$\dot{V}(x) \leq 0 \tag{3.35}$$

These conditions consider that x has a stable origin and the storage function is equal to zero only at origin. On the other hand, for stability, time derivative of that storage function must be negative or equal to zero at any place including the origin. However, if strictly negative time derivative is present, asymptotic stability is present but locally. Global asymptotic stability can be proven by an additional condition of radial boundedness.

In this study, a storage function is constructed by releasing the small angle assumption on error system. In Equations (3.36), (3.37), (3.38) and (3.39) it is possible to see the expression of this error system. These equations have the same structure after reordering the terms in Equations (3.13), (3.14), (3.15) and (3.16) by adding the terms in Equations (3.17), (3.18) and (3.19).

$$\dot{e}_1 = u_1 + \left(\frac{v_r tan(\varphi_d)}{l} - u_2\right) e_2 \tag{3.36}$$

$$\dot{e}_2 = v_r sin(e_3) - \left(\frac{v_r tan(\varphi_d)}{l} - u_2\right) e_1 \tag{3.37}$$

$$\dot{e}_3 = u_2 \tag{3.38}$$

$$\dot{e}_4 = u_3 \tag{3.39}$$

For this given new system of errors, a storage function is designed and Lyapunov criteria are checked for stability and form of inputs. Equations (3.40) and (3.41) show the storage function and its time derivative.

$$V(e) = \frac{1}{2}({e_1}^2 + {e_2}^2 + {e_4}^2) + \frac{1 - cos(e_3)}{k_2} \tag{3.40}$$

$$\dot{V}(e) = e_1\dot{e}_1 + e_2\dot{e}_2 + e_4\dot{e}_4 + \frac{sin(e_3)}{k_2}\dot{e}_3 \tag{3.41}$$

First condition of Lyapunov stability analysis is satisfied for whole state space if k_2 is a positive constant. This can be said thanks to a quadratic form and limit values of trigonometric functions. However, for second condition, a feedback design is necessary. Equations (3.42), (3.43) and (3.44) shows applied inputs to this system.

$$u_1 = -k_1 e_1 \tag{3.42}$$

$$u_2 = -k_2 v_r e_2 \tag{3.43}$$

$$u_3 = -k_3 e_4 \tag{3.44}$$

If these feedback terms are injected to system dynamics as shown in Equations (3.36), (3.37), (3.38) and (3.39), derivative of the selected storage function can be expressed differently. This operation is shown in Equation (3.45). Then, cancellations are performed and a simpler form is obtained in Equation (3.46).

$$\dot{V} = -k_1 e_1^2 + e_2 v_r sin(e_3) + \frac{sin(e_3)}{k_2}(-k_2 v_r e_2) - k_3 e_4^2 \tag{3.45}$$

$$\dot{V} = -k_1 e_1^2 - k_3 e_4^2 \tag{3.46}$$

Obtained function is satisfying second condition of Lyapunov stability if k_1 and k_3 are positive. Therefore, this design can be considered locally asymptotically stable. Globally asymptotical functions can be defined easily. So that this storage function can become a globally asymptotically stable Lyapunov function.

This can be done through the radial unboundedness property of functions. This property needs the time derivative of Lyapunov candidate function strictly negative at any point of state space other than origin. Then checks if there exists a limit value of the Lyapunov candidate function. If there doesn't exist any limit of storage function as norm of the state variables go to infinity, then Lyapunov function can be considered radially unbounded. Equations (3.47) and (3.48) show radial unboundedness condition for error system shown previously.

$$\dot{V}(e) < 0, \forall e \in \mathbb{R}^4 \backslash [0\ 0\ 0\ 0]^T \tag{3.47}$$

$$\lim_{\|e\|\to\infty} V(e) \to \infty \tag{3.48}$$

Equations (3.40) and (3.46) satisfy radial unboundedness condition. Therefore, this control design is considered globally asymptotically stable.

However, for this design two coordinate changes are used. These changes must be reversed to get real speed and rate of change of steering angle. This consideration is clarified in next sections.

3.3 Global Control Approach

Instead of creating an error system, a direct application of global controllers is a possible approach. This approach depends on cancellation of undesired dynamics of the system and assigning desired ones. Dynamic feedback linearization through coordinate transformation is a strong method to perform this operation.

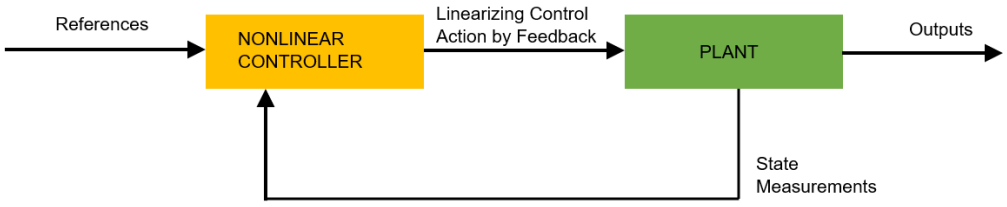

Figure 3.2 Control scheme explanation

3.4.1 Dynamic Input-Output Feedback Linearizing Control

This technique aims to come up with a feedback action cancelling nonlinearities in system and introduce a linear system dynamics which can be used for control easily. Form of the controller contains a cancelling part and a new input term. Since this

approach is a long and complicated control technique, the steps must be clearly explained.

Steps:

1. Write system in state space form
2. Check relative degree
3. Construct decoupling matrix
4. Check invertibility of decoupling matrix
5. Construct feedback action with finalized inputs
6. Compute necessary coordinate transformation
7. Control linear part and use inputs in new feedback action

In these 7 steps, the kinematic bicycle model is controlled by this approach.

3.4.1.1 Step 1: Write System in State Space Form

System was already shown in state space form in Equation (2.13). However, for ease of calculations and traceability, it is rewritten as in Equations (3.49), (3.50), (3.51) and (3.52).

$$\dot{x} = v\cos(\theta) = \dot{x}_1 \tag{3.49}$$

$$\dot{y} = v\sin(\theta) = \dot{x}_2 \tag{3.50}$$

$$\dot{\theta} = v\frac{\tan(\varphi)}{l} = \dot{x}_3 \tag{3.51}$$

$$\dot{\varphi} = w = \dot{x}_4 \tag{3.52}$$

This system is in general form of an input affine nonlinear system as shown in Equations (3.53) and (3.54).

$$\dot{x} = f(x) + g(x)u \tag{3.53}$$

$$y = h(x) \tag{3.54}$$

With following definitions shown in Equations (3.55), (3.56), (3.57), (3.58) and (3.59) it is possible to move on to the next step.

$$f(x) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{3.55}$$

$$g_1(x) = \begin{pmatrix} \cos(x_3) \\ \sin(x_3) \\ \dfrac{\tan(x_4)}{l} \\ 0 \end{pmatrix} \tag{3.56}$$

$$g_2(x) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.57}$$

$$h_1(x) = x_1 \tag{3.58}$$

$$h_2(x) = x_2 \tag{3.59}$$

Where system is defined in a different form. Since computational load of this method is big, outputs are taken only cartesian coordinates of the model. Therefore, it is directly concentrated on trajectory tracking purpose.

3.4.1.2 Step 2: Check Relative Degree

Lie derivation is used again to determine nonzero contribution of input-output pairs. Equation (3.60) shows the main form of this operation.

$$L_{g(x)}h(x) = \left[\frac{\partial h(x)}{\partial x}\right]^T g(x) \tag{3.60}$$

This operation is applied to expressions shown in Equations (3.56), (3.57), (3.58) and (3.59). Results can be seen in Equations (3.61), (3.62), (3.63) and (3.64).

$$L_{g_1}h_1 = \begin{pmatrix}1 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}\cos(x_3)\\ \sin(x_3)\\ \frac{\tan(x_4)}{l}\\ 0\end{pmatrix} = \cos(x_3) \tag{3.61}$$

$$L_{g_2}h_1 = \begin{pmatrix}1 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}0\\ 0\\ 0\\ 1\end{pmatrix} = 0 \tag{3.62}$$

$$L_{g_1}h_2 = \begin{pmatrix}0 & 1 & 0 & 0\end{pmatrix}\begin{pmatrix}\cos(x_3)\\ \sin(x_3)\\ \frac{\tan(x_4)}{l}\\ 0\end{pmatrix} = \sin(x_3) \tag{3.63}$$

$$L_{g_2}h_2 = (0 \quad 1 \quad 0 \quad 0)\begin{pmatrix}0\\0\\0\\1\end{pmatrix} = 0 \tag{3.64}$$

For both input-output pairs, nonzero component appears at the first Lie derivation. Therefore, relative degree is equal to 1 for this case. Equation (3.65) shows relative degrees.

$$r_1 = r_2 = 1 \tag{3.65}$$

3.4.1.3 Step 3: Construct Decoupling Matrix

Decoupling matrix shows input-output relations of all channels. Equation (3.66) represents decoupling matrix form for two inputs and two outputs case.

$$H(x) = \begin{bmatrix} L_{g_1}{L_f}^{r_1-1}h_1 & L_{g_2}{L_f}^{r_1-1}h_1 \\ L_{g_1}{L_f}^{r_2-1}h_2 & L_{g_2}{L_f}^{r_2-1}h_2 \end{bmatrix} \tag{3.66}$$

Where H(x) is the decoupling matrix. This matrix must be invertible in order to perform dynamic feedback linearizing control. Therefore, singularity conditions must be checked. For kinematic bicycle model, decoupling matrix is constructed by using Equations (3.61), (3.62), (3.63), (3.64), (3.65) and (3.66). The result is shown in Equation (3.67).

$$H(x) = \begin{pmatrix} \cos(x_3) & 0 \\ \sin(x_3) & 0 \end{pmatrix} \tag{3.67}$$

However, decoupling matrix is singular. Therefore, it cannot be used in dynamic feedback linearization. Hence, by adding and integrator on the first channel it is possible to extend the system. Moreover, it is necessary to perform previous steps until getting an invertible decoupling matrix. Figure 3.3 shows related block scheme.

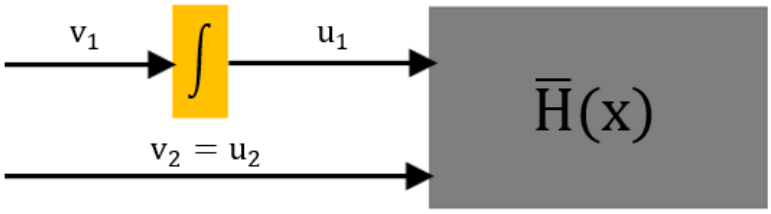

Figure 3.3 Extended system creation

By adding one integrator, a new state space relation is obtained. New system has different inputs. Equations (3.68), (3.69), (3.70), (3.71) and (3.72) represent new state space system.

$$\dot{x} = x_5 \cos(x_3) = \dot{x}_1 \tag{3.68}$$

$$\dot{y} = x_5 \sin(x_3) = \dot{x}_2 \tag{3.69}$$

$$\dot{\theta} = x_5 \frac{\tan(x_4)}{l} = \dot{x}_3 \tag{3.70}$$

$$\dot{\varphi} = w = v_2 = \dot{x}_4 \tag{3.71}$$

$$\dot{x}_5 = v_1 \tag{3.72}$$

For new set of inputs, Equations (3.55), (3.56), (3.57), (3.58) and (3.59) become as shown in Equations (3.73), (3.74), (3.75), (3.76) and (3.77).

$$\bar{f}(x) = \begin{pmatrix} x_5 \cos(x_3) \\ x_5 \sin(x_3) \\ x_5 \dfrac{\tan(x_4)}{l} \\ 0 \\ 0 \end{pmatrix} \tag{3.73}$$

$$\bar{g}_1(x) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.74}$$

$$\bar{g}_2(x) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \tag{3.75}$$

$$\bar{h}_1(x) = x_1 \tag{3.76}$$

$$\bar{h}_2(x) = x_2 \tag{3.77}$$

Same operation to find nonzero contribution of input-output pairs are done for this new system as well. Same order of operations is followed to find invertible decoupling matrix. Equations (3.78), (3.79), (3.80) and (3.81) show Lie derivation operations for extended system.

$$L_{\bar{g}_1}\bar{h}_1 = \begin{pmatrix}1 & 0 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}0\\0\\0\\0\\1\end{pmatrix} = 0 \tag{3.78}$$

$$L_{\bar{g}_2}\bar{h}_1 = \begin{pmatrix}1 & 0 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}0\\0\\0\\1\\0\end{pmatrix} = 0 \tag{3.79}$$

$$L_{\bar{g}_1}\bar{h}_2 = \begin{pmatrix}0 & 1 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}0\\0\\0\\0\\1\end{pmatrix} = 0 \tag{3.80}$$

$$L_{\bar{g}_2}\bar{h}_2 = \begin{pmatrix}0 & 1 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}0\\0\\0\\1\\0\end{pmatrix} = 0 \tag{3.81}$$

First Lie derivation stage does not represent a nonzero input-output pair. Therefore, Lie derivation must be applied until a nonzero pair appear. Second step of Lie derivation is expressed in Equation (3.82).

$$L_{\bar{g}(x)}L_{\bar{f}(x)}\bar{h}(x) = \left[\frac{\partial L_{\bar{f}(x)}\bar{h}(x)}{\partial x}\right]^T \bar{g}(x) \tag{3.82}$$

Application of the second step of Lie derivation is done in two parts. First part can be seen in Equations (3.83) and (3.84).

$$L_{\bar{f}}\bar{h}_1 = (1 \quad 0 \quad 0 \quad 0 \quad 0)\begin{pmatrix} x_5\cos(x_3) \\ x_5\sin(x_3) \\ x_5\dfrac{\tan(x_4)}{l} \\ 0 \\ 0 \end{pmatrix} = x_5\cos(x_3) \tag{3.83}$$

$$L_{\bar{f}}\bar{h}_2 = (0 \quad 1 \quad 0 \quad 0 \quad 0)\begin{pmatrix} x_5\cos(x_3) \\ x_5\sin(x_3) \\ x_5\dfrac{\tan(x_4)}{l} \\ 0 \\ 0 \end{pmatrix} = x_5\sin(x_3) \tag{3.84}$$

Second part uses results of Equations (3.83) and (3.84) and apply another Lie derivation with input vector fields. These operations are shown in Equations (3.85), (3.86), (3.87) and (3.88).

$$L_{\bar{g}_1}L_{\bar{f}}\bar{h}_1 = (0 \quad 0 \quad -x_5\sin(x_3) \quad 0 \quad \cos(x_3))\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \cos(x_3) \tag{3.85}$$

$$L_{\bar{g}_2}L_{\bar{f}}\bar{h}_1 = (0 \quad 0 \quad -x_5\sin(x_3) \quad 0 \quad \cos(x_3))\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = 0 \tag{3.86}$$

$$L_{\bar{g}_1}L_{\bar{f}}\bar{h}_2 = (0 \quad 0 \quad x_5\cos(x_3) \quad 0 \quad \sin(x_3))\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \sin(x_3) \tag{3.87}$$

$$L_{\bar{g}_2} L_{\bar{f}} \bar{h}_2 = (0 \quad 0 \quad x_5 \cos(x_3) \quad 0 \quad \sin(x_3)) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = 0 \tag{3.88}$$

This operation for extended system gives new relative degree for input-output relationship. Relative degrees are presented in Equation (3.89).

$$\bar{r}_1 = \bar{r}_2 = 2 \tag{3.89}$$

Therefore, extended system has a new decoupling matrix. This matrix can be seen in Equation (3.90).

$$H(x) = \begin{pmatrix} \cos(x_3) & 0 \\ \sin(x_3) & 0 \end{pmatrix} \tag{3.90}$$

However, this resultant decoupling matrix is the same as the previous one. Therefore, same operations are performed again. This approach is done by adding another integrator on the first input channel. By this way lagging input becomes apparent. New form of representation can be seen in Figure 3.4.

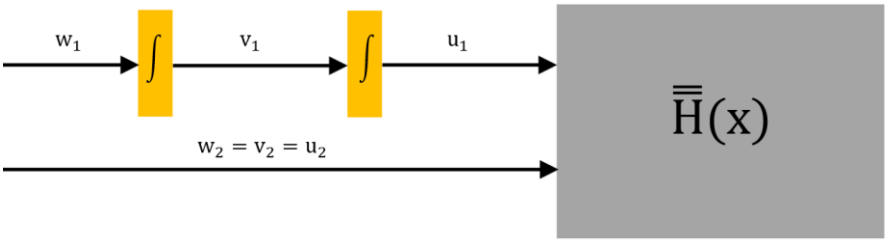

Figure 3.4 Creation of extended system for second step

Since another integrator is added to the system, number of states will increase by one as it did for the first extension. State space representation of the new extended system is presented in Equations (3.91), (3.92), (3.93), (3.94), (3.95) and (3.96).

$$\dot{x} = x_5 \cos(x_3) = \dot{x}_1 \tag{3.91}$$

$$\dot{y} = x_5 \sin(x_3) = \dot{x}_2 \tag{3.92}$$

$$\dot{\theta} = x_5 \frac{\tan(x_4)}{l} = \dot{x}_3 \tag{3.93}$$

$$\dot{\varphi} = w = v_2 = w_2 = \dot{x}_4 \tag{3.94}$$

$$\dot{x}_5 = x_6 \tag{3.95}$$

$$\dot{x}_6 = w_1 \tag{3.96}$$

For the second extension of the system, new inputs are presented as previously shown in first extension. State space system is shown in vectoral form in Equations (3.97), (3.98), (3.99), (3.100) and (3.101).

$$\bar{\bar{f}}(x) = \begin{pmatrix} x_5 \cos(x_3) \\ x_5 \sin(x_3) \\ x_5 \dfrac{\tan(x_4)}{l} \\ 0 \\ x_6 \\ 0 \end{pmatrix} \tag{3.97}$$

$$\bar{\bar{g}}_1(x) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{3.98}$$

$$\bar{\bar{g}}_2(x) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \tag{3.99}$$

$$\bar{\bar{h}}_1(x) = x_1 \tag{3.100}$$

$$\bar{\bar{h}}_2(x) = x_2 \tag{3.101}$$

As it is shown in first extension, same steps of Lie derivations are applied for the second extension of the system. First step of Lie derivations are shown in Equations (3.102), (3.103), (3.104) and (3.105).

$$L_{\bar{\bar{g}}_1}\bar{\bar{h}}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = 0 \tag{3.102}$$

$$L_{\bar{\bar{g}}_2}\bar{\bar{h}}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = 0 \tag{3.103}$$

$$L_{\bar{\bar{g}}_1}\bar{\bar{h}}_2 = \begin{pmatrix}0 & 1 & 0 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}0\\0\\0\\0\\0\\1\end{pmatrix} = 0 \tag{3.104}$$

$$L_{\bar{\bar{g}}_2}\bar{\bar{h}}_2 = \begin{pmatrix}0 & 1 & 0 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}0\\0\\0\\1\\0\\0\end{pmatrix} = 0 \tag{3.105}$$

It is evident that Lie derivation operation must continue for another step like shown in Equation (3.82). First part of second step Lie derivation can be seen in Equations (3.106) and (3.107).

$$L_{\bar{\bar{f}}}\bar{\bar{h}}_1 = \begin{pmatrix}1 & 0 & 0 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}x_5\cos(x_3)\\x_5\sin(x_3)\\x_5\dfrac{\tan(x_4)}{l}\\0\\x_6\\0\end{pmatrix} = x_5\cos(x_3) \tag{3.106}$$

$$L_{\bar{\bar{f}}}\bar{\bar{h}}_2 = \begin{pmatrix}0 & 1 & 0 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}x_5\cos(x_3)\\x_5\sin(x_3)\\x_5\dfrac{\tan(x_4)}{l}\\0\\x_6\\0\end{pmatrix} = x_5\sin(x_3) \tag{3.107}$$

Lie derivation is continued for second part of second step. Equations (3.108), (3.109), (3.110) and (3.111) show these operations.

$$L_{\bar{\bar{g}}_1} L_{\bar{\bar{f}}} \bar{\bar{h}}_1 = (0 \quad 0 \quad -x_5 \sin(x_3) \quad 0 \quad 0 \quad \cos(x_3)) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = 0 \qquad (3.108)$$

$$L_{\bar{\bar{g}}_2} L_{\bar{\bar{f}}} \bar{\bar{h}}_1 = (0 \quad 0 \quad -x_5 \sin(x_3) \quad 0 \quad 0 \quad \cos(x_3)) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = 0 \qquad (3.109)$$

$$L_{\bar{\bar{g}}_1} L_{\bar{\bar{f}}} \bar{\bar{h}}_2 = (0 \quad 0 \quad x_5 \cos(x_3) \quad 0 \quad 0 \quad \sin(x_3)) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = 0 \qquad (3.110)$$

$$L_{\bar{\bar{g}}_2} L_{\bar{\bar{f}}} \bar{\bar{h}}_2 = (0 \quad 0 \quad x_5 \cos(x_3) \quad 0 \quad 0 \quad \sin(x_3)) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = 0 \qquad (3.111)$$

However, there are no apparent nonzero result of these operations. Therefore, a third step of Lie derivation is necessary. Form of third Lie derivation is shown in Equation (3.112).

$$L_{\bar{\bar{g}}(x)} L_{\bar{\bar{f}}(x)} L_{\bar{\bar{f}}(x)} \bar{\bar{h}}(x) = \left[\frac{\partial L_{\bar{\bar{f}}(x)} L_{\bar{\bar{f}}(x)} \bar{\bar{h}}(x)}{\partial x} \right]^T \bar{\bar{g}}(x) \tag{3.112}$$

As shown previously, this operation can be performed part by part. Second part is shown in Equations (3.113) and (3.114).

$$L_{\bar{\bar{f}}} L_{\bar{\bar{f}}} \bar{\bar{h}}_1 = \begin{pmatrix} 0 \\ 0 \\ -x_5 \sin(x_3) \\ 0 \\ 0 \\ \cos(x_3) \end{pmatrix}^T \begin{pmatrix} x_5 \cos(x_3) \\ x_5 \sin(x_3) \\ x_5 \dfrac{\tan(x_4)}{l} \\ 0 \\ x_6 \\ 0 \end{pmatrix}$$

$$= -{x_5}^2 \sin(x_3) \frac{\tan(x_4)}{l} + x_6 \cos(x_3) \tag{3.113}$$

$$L_{\bar{\bar{f}}} L_{\bar{\bar{f}}} \bar{\bar{h}}_2 = \begin{pmatrix} 0 \\ 0 \\ -x_5 \cos(x_3) \\ 0 \\ 0 \\ \sin(x_3) \end{pmatrix}^T \begin{pmatrix} x_5 \cos(x_3) \\ x_5 \sin(x_3) \\ x_5 \dfrac{\tan(x_4)}{l} \\ 0 \\ x_6 \\ 0 \end{pmatrix}$$

$$= {x_5}^2 \cos(x_3) \frac{\tan(x_4)}{l} + x_6 \sin(x_3) \tag{3.114}$$

After these calculations, a third step is applied for both input vector fields. Third step can be seen in Equations (3.115), (3.116), (3.117) and (3.118).

$$L_{\bar{\bar{g}}_1} L_{\bar{\bar{f}}}{}^2 \bar{\bar{h}}_1 = \begin{pmatrix} 0 \\ 0 \\ -x_5{}^2 \cos(x_3) \dfrac{\tan(x_4)}{l} - x_6 \sin(x_3) \\ -x_5{}^2 \sin(x_3) \dfrac{1}{l \cos^2(x_4)} \\ -2x_5 \sin(x_3) \dfrac{\tan(x_4)}{l} \\ \cos(x_3) \end{pmatrix}^T \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \cos(x_3) \qquad (3.115)$$

$$L_{\bar{\bar{g}}_2} L_{\bar{\bar{f}}}{}^2 \bar{\bar{h}}_1 = \begin{pmatrix} 0 \\ 0 \\ -x_5{}^2 \cos(x_3) \dfrac{\tan(x_4)}{l} - x_6 \sin(x_3) \\ -x_5{}^2 \sin(x_3) \dfrac{1}{l \cos^2(x_4)} \\ -2x_5 \sin(x_3) \dfrac{\tan(x_4)}{l} \\ \cos(x_3) \end{pmatrix}^T \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$= -x_5{}^2 \sin(x_3) \frac{1}{l \cos^2(x_4)} \qquad (3.116)$$

$$L_{\bar{\bar{g}}_1} L_{\bar{\bar{f}}}{}^2 \bar{\bar{h}}_2 = \begin{pmatrix} 0 \\ 0 \\ -x_5{}^2 \sin(x_3) \dfrac{\tan(x_4)}{l} + x_6 \cos(x_3) \\ x_5{}^2 \cos(x_3) \dfrac{1}{l \cos^2(x_4)} \\ 2x_5 \cos(x_3) \dfrac{\tan(x_4)}{l} \\ \sin(x_3) \end{pmatrix}^T \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \sin(x_3) \qquad (3.117)$$

$$L_{\bar{\bar{g}}_2} L_{\bar{\bar{f}}}{}^2 \bar{\bar{h}}_2 = \begin{pmatrix} 0 \\ 0 \\ -x_5{}^2 \sin(x_3) \dfrac{\tan(x_4)}{l} + x_6 \cos(x_3) \\ x_5{}^2 \cos(x_3) \dfrac{1}{l \cos^2(x_4)} \\ 2x_5 \cos(x_3) \dfrac{\tan(x_4)}{l} \\ \sin(x_3) \end{pmatrix}^T \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$
$$= x_5{}^2 \cos(x_3) \frac{1}{l \cos^2(x_4)} \tag{3.118}$$

As a result, relative degrees are found and expressed in Equation (3.119).

$$\bar{\bar{r}}_1 = \bar{\bar{r}}_2 = 3 \tag{3.119}$$

3.4.1.4 Step 4: Check Invertibility of Decoupling Matrix

Under the light of previous calculations, decoupling matrix is obtained. Equation (3.120) shows the decoupling matrix. Invertibility is checked through determinant operation of decoupling matrix and expressed in Equation (3.121).

$$\bar{\bar{H}}(x) = \begin{pmatrix} L_{\bar{\bar{g}}_1} L_{\bar{\bar{f}}}{}^{\bar{\bar{r}}_1 - 1} \bar{\bar{h}}_1 & L_{\bar{\bar{g}}_2} L_{\bar{\bar{f}}}{}^{\bar{\bar{r}}_1 - 1} \bar{\bar{h}}_1 \\ L_{\bar{\bar{g}}_1} L_{\bar{\bar{f}}}{}^{\bar{\bar{r}}_2 - 1} \bar{\bar{h}}_2 & L_{\bar{\bar{g}}_2} L_{\bar{\bar{f}}}{}^{\bar{\bar{r}}_2 - 1} \bar{\bar{h}}_2 \end{pmatrix}$$
$$= \begin{pmatrix} \cos(x_3) & -x_5{}^2 \sin(x_3) \dfrac{1}{l \cos^2(x_4)} \\ \sin(x_3) & x_5{}^2 \cos(x_3) \dfrac{1}{l \cos^2(x_4)} \end{pmatrix} \tag{3.120}$$

$$det\left(\bar{\bar{H}}(x)\right) = \frac{x_5{}^2}{l \cos^2(x_4)} \tag{3.121}$$

Invertibility of decoupling matrix is guaranteed for 6-dimensional system for following conditions expressed in Equations (3.122) and (3.123).

$$x_5 \neq 0 \tag{3.122}$$

$$x_4 \neq k\frac{\pi}{2} for k \in \mathbb{Z} \tag{3.123}$$

As long as these conditions are not violated, decoupling matrix will be nonsingular. Therefore, a desired structure is obtained. Note that $x_3 = k\frac{\pi}{2}, k \in \mathbb{Z}$ is also a critical point. However, even at these points none of the rows or columns are equal to zero. Therefore, a noninteracting control form can be seen.

3.4.1.5 Step 5: Construct Feedback Action with Finalized Inputs

Form of the feedback action contains inverted input-output relation which is defined previously by decoupling matrix. This contribution cancels undesired nonlinearities in terms of control. Then a linear action is introduced to control the feedback linearized system. This form of feedback is shown in Equation (3.124).

$$W(x) = \bar{\bar{H}}^{-1}(x)[-Q(x) + U_{linear}(x)] \tag{3.124}$$

This form contains another term which is shown with Q(x). This term can be computed again with Lie derivation procedure. Form of Q(x) is shown in Equation (3.125).

$$Q(x) = \begin{pmatrix} L_{\bar{\bar{f}}}{}^{\bar{\bar{r}}_1}\bar{\bar{h}}_1 \\ L_{\bar{\bar{f}}}{}^{\bar{\bar{r}}_2}\bar{\bar{h}}_2 \end{pmatrix} = \begin{pmatrix} L_{\bar{\bar{f}}}{}^{3}\bar{\bar{h}}_1 \\ L_{\bar{\bar{f}}}{}^{3}\bar{\bar{h}}_2 \end{pmatrix} \tag{3.125}$$

Necessary calculations of Q(x) terms are shown in Equations (3.126) and (3.127). It has 3 parts of calculations however first two parts were performed and expressed in Equations (3.113) and (3.114). Equations (3.126) and (3.127) show the results for calculated Q(x) matrix elements.

$$L_{\bar{\bar{f}}}{}^{3}\bar{\bar{h}}_1 = \begin{pmatrix} 0 \\ 0 \\ -{x_5}^2\cos(x_3)\dfrac{\tan(x_4)}{l} - x_6\sin(x_3) \\ -{x_5}^2\sin(x_3)\dfrac{1}{l\cos^2(x_4)} \\ -2x_5\sin(x_3)\dfrac{\tan(x_4)}{l} \\ \cos(x_3) \end{pmatrix}^T \begin{pmatrix} x_5\cos(x_3) \\ x_5\sin(x_3) \\ x_5\dfrac{\tan(x_4)}{l} \\ 0 \\ x_6 \\ 0 \end{pmatrix}$$

$$= x_5\frac{\tan(x_4)}{l}\left(-{x_5}^2\cos(x_3)\frac{\tan(x_4)}{l} - x_6\sin(x_3)\right) - \frac{2x_5x_6\sin(x_3)\tan(x_4)}{l} \tag{3.126}$$

$$L_{\bar{\bar{f}}}{}^{3}\bar{\bar{h}}_2 = \begin{pmatrix} 0 \\ 0 \\ -{x_5}^2\sin(x_3)\dfrac{\tan(x_4)}{l} + x_6\cos(x_3) \\ {x_5}^2\cos(x_3)\dfrac{1}{l\cos^2(x_4)} \\ 2x_5\cos(x_3)\dfrac{\tan(x_4)}{l} \\ \sin(x_3) \end{pmatrix}^T \begin{pmatrix} x_5\cos(x_3) \\ x_5\sin(x_3) \\ x_5\dfrac{\tan(x_4)}{l} \\ 0 \\ x_6 \\ 0 \end{pmatrix}$$

$$= x_5\frac{\tan(x_4)}{l}\left(-{x_5}^2\sin(x_3)\frac{\tan(x_4)}{l} + x_6\cos(x_3)\right) + \frac{2x_5x_6\sin(x_3)\tan(x_4)}{l} \tag{3.127}$$

Resulting Q(x) vector can be seen in Equation (3.128).

$$Q(x) = \begin{pmatrix} -\frac{{x_5}^3 \tan^2(x_4)\cos(x_3)}{l^2} - \frac{3x_5x_6\tan(x_4)\sin(x_3)}{l} \\ -\frac{{x_5}^3 \tan^2(x_4)\sin(x_3)}{l^2} + \frac{3x_5x_6\tan(x_4)\cos(x_3)}{l} \end{pmatrix} \quad (3.128)$$

Last term to be defined is linear control inputs. It is a vector with two elements for two input channels. Form of this vector is shown in Equation (3.129).

$$U_{linear}(x) = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \quad (3.129)$$

Complete form of new input vector W(x) defined in Equation (3.124) is rewritten in Equation (3.130).

$$\begin{pmatrix} \frac{{x_5}^3 \tan^2(x_4)}{l^2} + u_1\cos(x_3) + u_2\sin(x_3) \\ -\frac{\cos(x_4)}{{x_5}^2}(3x_5x_6\sin(x_4) + lu_1\cos(x_4)\sin(x_3) - lu_2\cos(x_4)\cos(x_3)) \end{pmatrix} \quad (3.130)$$

3.4.1.6 Step 6: Compute Necessary Coordinate Transformation

In this step a coordinate transformation is computed in order to get a chained form of linear system representation. Equation (3.131) defines the form of coordinate transformation for 6-dimensional state space. Since sum of relative degrees is equal to the number of states, any hidden dynamics is not present. Therefore, it is possible to obtain an exact linearization through feedback. This result also guarantees that no unobservable dynamics is contributing this system.

$$\xi(x) = \begin{pmatrix} \bar{\bar{h}}_1 \\ L_{\bar{\bar{f}}} \bar{\bar{h}}_1 \\ L_{\bar{\bar{f}}}{}^2 \bar{\bar{h}}_1 \\ \bar{\bar{h}}_2 \\ L_{\bar{\bar{f}}} \bar{\bar{h}}_2 \\ L_{\bar{\bar{f}}}{}^2 \bar{\bar{h}}_2 \end{pmatrix} \tag{3.131}$$

For the considered system this coordinate transformation is expressed in Equation (3.132). Note that Lie derivations shown in Equation (3.131) are calculated in previous steps.

$$\xi(x) = \begin{pmatrix} x_1 \\ x_5 \cos(x_3) \\ -x_5{}^2 \sin(x_3) \dfrac{\tan(x_4)}{l} + x_6 \cos(x_3) \\ x_2 \\ x_5 \sin(x_3) \\ x_5{}^2 \cos(x_3) \dfrac{\tan(x_4)}{l} + x_6 \sin(x_3) \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \\ z_6 \end{pmatrix} \tag{3.132}$$

This transformation separates the system into two subsystems. These two subsystems are proven to be in chained form by simply checking their time derivatives.

Existence of first chain of integrators is shown in Equations (3.133), (3.134) and (3.135). Calculations are done by taking time derivatives of newly defined coordinates and comparing with state space system shown in Equations (3.91), (3.92), (3.93), (3.94), (3.95) and (3.96).

$$\dot{z}_1 = \dot{x}_1 = x_5 \cos x_3 = \dot{z}_2 \tag{3.133}$$

$$\dot{z}_2 = \dot{x}_5 \cos(x_3) - x_5 \dot{x}_3 \sin(x_3) = x_6 cos(x_3) - \frac{{x_5}^2 sin(x_3) \tan(x_4)}{l} = \dot{z}_3 \quad (3.134)$$

$$\begin{aligned} \dot{z}_3 &= \dot{x}_6 \cos(x_3) - x_6 \dot{x}_3 \sin(x_3) - \frac{2x_5 \dot{x}_5 \sin(x_3) \tan(x_4)}{l} \\ &\quad - \frac{{x_5}^2}{l}\left(\dot{x}_3 \cos(x_3) \tan(x_4) + \frac{\dot{x}_4 \sin(x_3)}{\cos^2(x_4)}\right) \\ &= u_1 \cos^2(x_3) + u_1 \sin^2(x_3) + u_2 cos(x_3) \sin(x_3) - u_2 cos(x_3) \sin(x_3) \\ &= u_1 \end{aligned} \quad (3.135)$$

First chain of integrators are realized by shown calculations. In other words, half of the extended system can be controlled linearly by designed feedback action.

Existence of second chain of integrators is shown in Equations (3.136), (3.137) and (3.138). Calculations are done in the same manner with first chain.

$$\dot{z}_4 = \dot{x}_2 = x_5 \sin x_3 = \dot{z}_5 \quad (3.136)$$

$$\dot{z}_5 = \dot{x}_5 \sin(x_3) + x_5 \dot{x}_3 \cos(x_3) = x_6 \sin(x_3) + {x_5}^2 \cos(x_3) \frac{\tan(x_4)}{l} = \dot{z}_6 \quad (3.137)$$

$$\begin{aligned} \dot{z}_6 &= \frac{2x_5 \dot{x}_5 \cos(x_3) \tan(x_4)}{l} + \frac{{x_5}^2}{l}\left(-\dot{x}_3 \sin(x_3) \tan(x_4) + \frac{\dot{x}_4 \cos(x_3)}{\cos^2(x_4)}\right) \\ &\quad + \dot{x}_6 \sin(x_3) + x_6 \dot{x}_3 \cos(x_3) \\ &= u_2(\sin^2(x_3) + \cos^2(x_3)) + \frac{{x_5}^3 \sin(x_3)}{l^2}\left(\frac{1}{\cos^2(x_4)} - 1 - \tan^2(x_4)\right) = u_2 \end{aligned} \quad (3.138)$$

Second chain of integrators are obtained and a linear form can be realized for both chains. Next section clarifies this procedure.

3.4.1.7 Step 7: Control Linear Part and Use Inputs in New Feedback Action

Two separate chains of integrators are realized by the designed feedback for 6-dimensional state space representation. Block scheme form of transformed system can be seen in Figure 3.5.

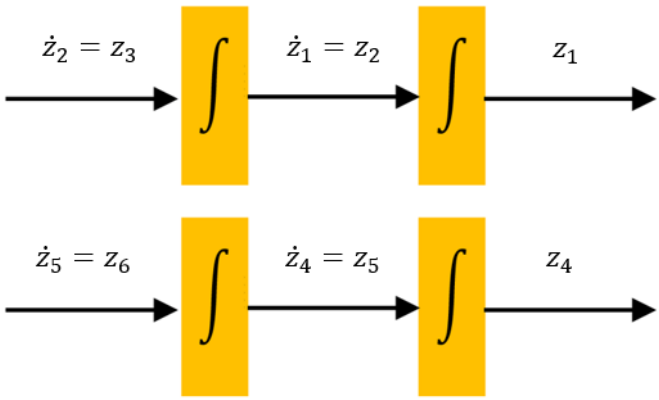

Figure 3.5 Two chains of integrators

Since this form is obtained, linearized subsystem must be controlled linearly. State space representation of this form is shown in Equations (3.139), (3.140), (3.141) and (3.142).

$$\begin{bmatrix}\dot{z}_1\\ \dot{z}_2\\ \dot{z}_3\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}z_1\\ z_2\\ z_3\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 1\end{bmatrix}u_1 \tag{3.139}$$

$$\begin{bmatrix}\dot{z}_4\\ \dot{z}_5\\ \dot{z}_6\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}z_4\\ z_5\\ z_6\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 1\end{bmatrix}u_2 \tag{3.140}$$

$$y_1 = z_1 = x_1 \tag{3.141}$$

$$y_2 = z_4 = x_2 \tag{3.142}$$

Controllability check is done through Equation (3.143).

$$det(\chi) = det([B \quad \vdots \quad AB \quad \vdots \quad A^{n-1}B]) = det\left(\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}\right) = -1 \neq 0 \tag{3.143}$$

Since controllability of linearized system is proven, any linear control approach can be used. However, initial conditions of every state in both coordinate system must be calculated.

3.5 Geometric Path Tracking Approach

Geometric path tracking algorithms are algorithms which use geometric approaches to track a previously given path. Difference of a path and a trajectory lies on time dependence. While a trajectory is defining time dependent inputs, a path is defining a set of points in the space. Therefore, a geometric approach is a logical and natural way for a path tracking task. In this work, Pure Pursuit Algorithm is used. This algorithm is widely studied and tests are done in various environments and competitions.

3.5.1 Pure Pursuit Algorithm

Pure Pursuit Algorithm uses rear wheel contact point of kinematic bicycle model unlike Stanley Algorithm using front wheel contact point to the ground. Therefore, this approach is selected for this work. The preliminary knowledge on path is necessary for generation of waypoints. These waypoints are considered as goals or setpoints of

the kinematic bicycle model. Figure 3.6 shows the geometric approach of Pure Pursuit Algorithm.

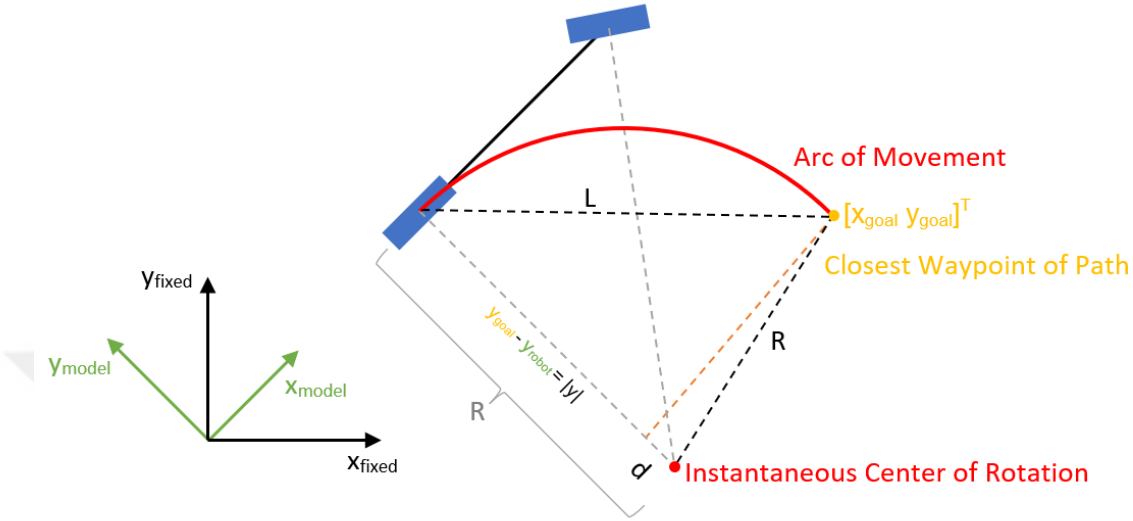

Figure 3.6 Pure Pursuit Algorithm scheme

This algorithm has one parameter called as the lookahead distance. In Figure 3.6 it is showed by L. Since this model will be rotating around an instantaneous center of rotation at every time instant, it is possible to define arcs connecting current position and a waypoint. By constraining a rotation movement for an arc with starting point at rear wheel reference and ending at a point L meters away, it is possible to find one unique arc. For this algorithm cartesian coordinates of the model is expressed in rotated reference frame by orientation angle in planar space. Equations (3.144), (3.145), (3.146) and (3.147) shows the calculations of R as a function of L.

$$R = |y| + d \tag{3.144}$$

$$d^2 + x^2 = R^2 \tag{3.145}$$

$$x^2 + y^2 = L^2 \tag{3.146}$$

$$R = \frac{L^2}{2|y|} \tag{3.147}$$

So that a unique arc is defined by its radius. Pure Pursuit Algorithm proposes a calculation of orientation angle error and it is calculated by using a trigonometric formulation. It is shown in Equation (3.148).

$$\alpha = \tan^{-1}\left(\frac{y_{goal} - y_{model}}{x_{goal} - x_{model}}\right) \tag{3.148}$$

This algorithm also proposes a calculation of necessary steering angle of kinematic bicycle model by this angle. This calculation is written in Equation (3.149).

$$\varphi_{reference} = \tan^{-1}\left(\frac{2l\sin(\alpha)}{L}\right) \tag{3.149}$$

However, if any point at lookahead distance is absent, then algorithm takes the closest point or middle point of two closest points as waypoint. Next subsection shows the control design for Pure Pursuit Algorithm. Introduced control design slightly differs from widely used proportional controllers.

3.5.1.1 Pure Pursuit Algorithm-Based Control

This part takes a feedforward and a feedback action of control together for input generation of kinematic bicycle model. This control technique considers only the waypoints which are ahead of the model. Proposed control law can be seen in Equations (3.150) and (3.151).

$$v = v_{desired} \tag{3.150}$$

$$w = w_{desired} + k_w(\varphi_{desired} - \varphi) \tag{3.151}$$

This form includes one control gain for rate of change of steering angle. Two feedforward terms are present for both inputs. Based on the task, desired final values can be introduced. Other terms which are multiplied by control gains will decay to zero while kinematic bicycle model gets closer to the waypoint. This decaying angular error can be seen in equation (3.151). These considerations are important for final point of a path. This control approach can be considered as a proportional control. Figure 3.7 clarifies this approach.

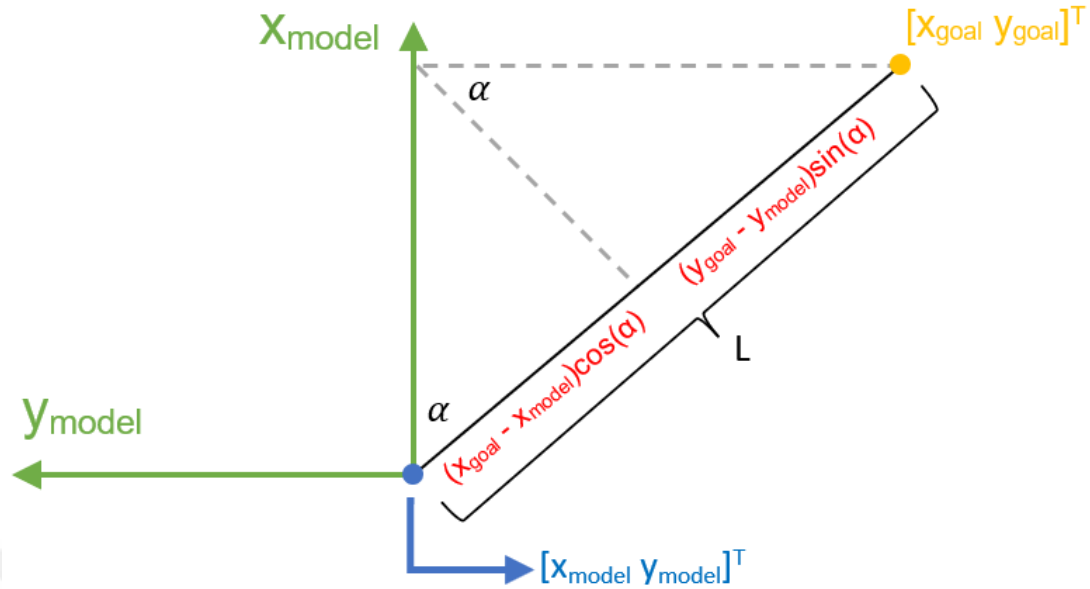

Figure 3.7 Pure Pursuit Algorithm-based control law

As a result, Pure Pursuit Algorithm and a related proportional control law is introduced. Only parameter of this approach is the lookahead distance of the model. Therefore, this approach can be realized by deciding this single parameter. Moreover, this selection is made by sensor selection. GPS, LIDAR, SONAR are utilized sensors for realization. Range and bearing measurement must be well designed.

CHAPTER FOUR
SENSORS AND FILTERING

4.1 Sensors and Measurement

Like any other electromechanical system, robots or vehicles using kinematic bicycle model obtain the necessary information on its states through measurement units. These units consist of sensors or transducers depending on measurement signal type. This topic is a popular research area in academia and industry because of its importance in real world applications. Therefore, a growing interest is already present and pushing industries of processors and sensory devices forward.

Sensory measurements for robots or vehicles using kinematic bicycle model are mainly divided into two subsections. These can be considered as state measurements of itself and external measurement units or alternatively known as proprioceptive and exteroceptive measurements. State measurements take measurements of steering angle, rotation angle of wheels, orientation in space, position, speed and acceleration. On the other hand, external measurement units collect information on distances to obstacles, detection of environmental features, applied external forces or torques. In next sections, selected measurement devices for state measurements and external measurements are expressed and applications are briefly explained together with filtering operations.

4.1.1 State Measurements

Since kinematic bicycle model is previously constructed in 4-dimensional space, measurement on each state is necessary. If a directly measuring sensor is not present or not applicable, an indirect measurement must be introduced by an estimation process through an observer or a proper estimator. However, necessary sensors and

transducers are present even for extended 6-dimensional system expressed for dynamic feedback linearization.

First two states are position information in plane. Therefore, their measurement is obtained through a single sensor. A GPS installation is a widely used technique. GPS uses multiple satellites to determine a position information. From all 31 satellites at least 4 of them are visible for a GPS signal receiver during any time of the day at any point of the world. Working principle relies on the atomic clocks. A GPS satellite sends a time signal to a receiver and then this receiver compares time difference of transmission. By this way it is possible to obtain position information. Nowadays, it is possible to enhance the accuracy of the position measurement obtained by GPS measurement. Therefore, it is a viable option. However, in GPS denied environments, different techniques must be used. Traveled distances can be measured through encoders or landmark-based measurement techniques can be preferred.

Orientation angle of the kinematic bicycle model is the third state to be measured. Generally, an IMU installation is satisfactory for this purpose. IMUs consist of accelerometers and gyroscopes inside their structure. Additionally, magnetometers and barometers are included for different applications. Therefore, IMUs can be used for measuring orientation of the body by its gyroscopes. Moreover, measurements provided by gyroscope and accelerometers can be integrated in time and therefore linear velocity and angular velocity can be obtained through IMU measurement data.

Finally, last state to be measured is the steering angle. Measurement of this state is relatively easy. A simple potentiometer can be used for this measurement. By using resistance change on this potentiometer, output voltage change can be measured and collected. Since set values are known for the potentiometer positions, steering angle would be measured directly.

4.1.2 External Measurements

These measurements are vital for the navigation and safety reasons. Range and bearing measurements to an obstacle or object and obstacle detection and classification are done by these sensors. Utilized measurement systems are explained below.

Range measurements are typically done by RADARs or LIDARs. RADARs use radio waves for detecting any object and measuring distance to the same object. This can be an obstacle or a wall with respect to the application field. They use echo measurement of a transmitted signal. On the other hand, LIDARs work in a different way. LIDARs are measurement systems that use lasers and by using multiple laser beams it can create a point cloud. This point cloud can be considered as 3D image of the environment around kinematic bicycle model subject. SONARs are also used for detection of objects and distance measurements. However, there is a tradeoff between these measurement techniques. While LIDARs are very expensive and sensitive to reflective surfaces, SONARs are relatively cheaper and can work in bad weather conditions. Nevertheless, LIDARs can extract information of a complete environment and speed of light ensures a faster measurement data.

Object and obstacle detection is done through a camera measurement. This is a relatively challenging task of measurement. Main objective behind this can be explained through widely used techniques for autonomous vehicles. For autonomous vehicle camera systems, a large dataset containing images of pedestrians, road signs, trees or pets are collected beforehand. Then a complex structure of deep neural networks is used for learning and classifying these objects successfully. Therefore, unlike previous sensing techniques, an additional work must be done on image processing. Additionally, when weather conditions are considered, these measurements become harder in terms of application.

Figure 4.1 shows sensory measurements of a kinematic bicycle model using robots or vehicles. For this figure a boundary box is shown as the vehicle body.

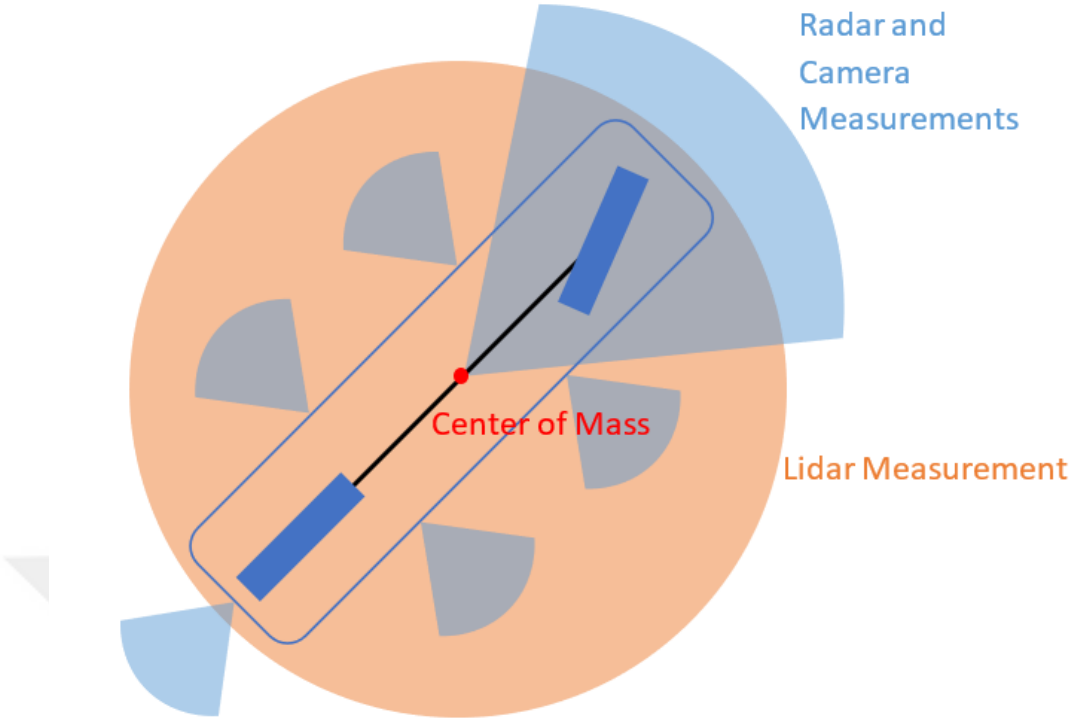

Figure 4.1 Measurement representation

Mainly long-range RADARs, LIDARs and cameras are installed on top of the robot or vehicle. In the plane, this point corresponds to center of mass. However, cameras and long-range RADARs can be mounted in front as well. Short-range RADARs are distributed around the vehicle or robot's body. IMUs and GPS devices are mainly mounted on center of mass as well.

By using multiple sensors, all states and information on environmental features can be measured. However, noise is always present for these measurements. Since the model itself is highly nonlinear and has multiple states to be measured, a sensor fusion and filtering operation is necessary. Therefore, next section is presented.

4.2 Filtering

As previously said, all measurements suffer from measurement noises. These noises can be dangerous or can cause unrealistic measurements. Therefore, filtering these

noises becomes an important task. Noises can be divided into two subgroups by their characteristics. Periodic noises can be present for the measurement system because of electrical or magnetic sources in the environment. Since these signals are present at any environment, they must be taken into account at any cases. Some hardware solutions are effective against these types of noises. However, software solutions are in general cheaper and require less work. On the other hand, random noise signals are also present at any environment. These noises require statistical approaches and therefore considered as slightly more challenging tasks.

4.2.1 Sinusoidal Noise Canceling

Sinusoidal noises can be cancelled through low-pass, high-pass or band-pass filters depending on the original signal's and noise's nature. On the other hand, by selecting an appropriate moving average filter would also smooth a high frequency periodic noisy signal.

Low-pass, high-pass and band-pass filters need a preliminary frequency analysis of a noisy signal. This can be done by utilizing Fourier Transformations. By this way it is possible to decide a convenient filter for the operation. Depending on frequency contents to be cancelled, filter type can be selected. However, this study is concentrated on statistical filtering operation.

4.2.2 Statistical Filtering

These types of filters take probability density functions and propose a good estimation of the measured states by using means and covariances of the measurement noises. This idea is detailed in next subsection.

4.2.2.1 State Estimation for Measurements

State estimation is one of the necessary applications for indirect or noisy measurements. The procedure can be explained by combining real measurements of a system and a mathematical model. Figure 4.2 shows this idea for a linear state space system and further explanations are done.

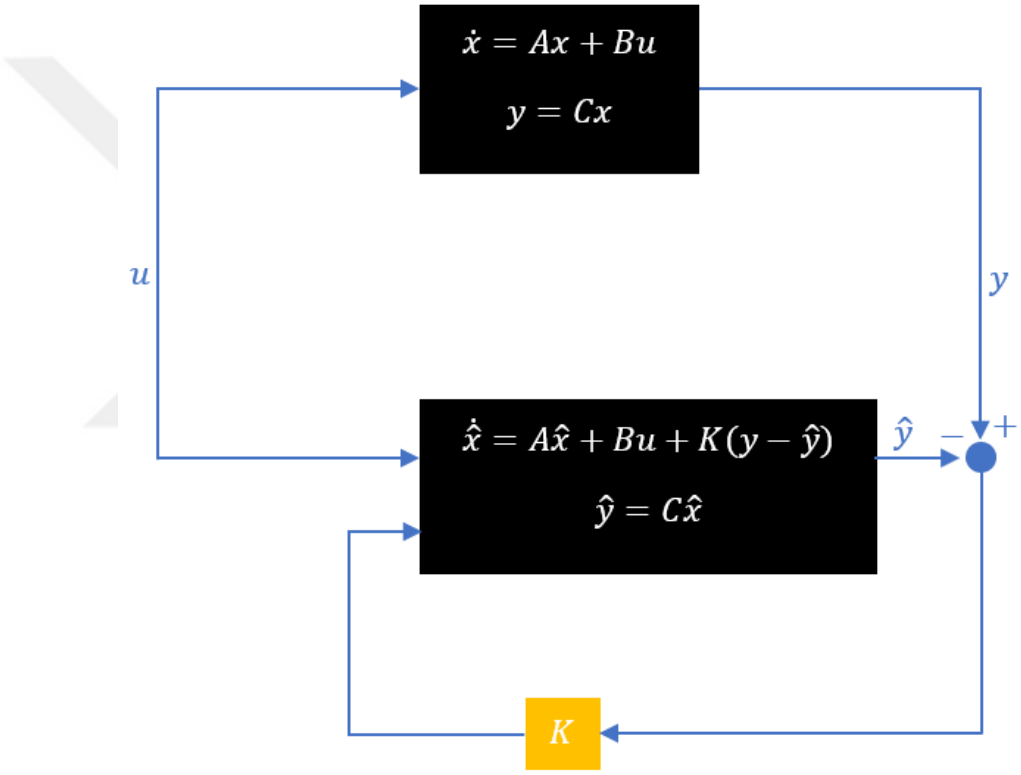

Figure 4.2 State observer block scheme

In this figure the real system and the estimator have the same inputs. The real system can be subjected to noises or uncertainties. However, the mathematical model, in this case the observer, does not have this contribution. The observer estimates the states by the mathematical calculations. By the observation gain K, the observer minimizes the error between estimated value and the measured value. This procedure is expressed in Equations (4.1), (4.2), (4.3), (4.4), (4.5), (4.6) and (4.7).

$$e = x - \hat{x} \tag{4.1}$$

$$\dot{x} = Ax + Bu \tag{4.2}$$

$$y = Cx \tag{4.3}$$

$$\dot{\hat{x}} = A\hat{x} + Bu + K(y - \hat{y}) \tag{4.4}$$

$$\hat{y} = C\hat{x} \tag{4.5}$$

$$y - \hat{y} = C(x - \hat{x}) \tag{4.6}$$

$$\dot{e} = (A - KC)e \tag{4.7}$$

Equation (4.1) defines the estimation error. Equations (4.2) and (4.3) define the state space system and equations (4.4) and (4.5) define the estimation with feedback gain K as shown in Figure 4.2. By combining these equations, it is possible to get Equations (4.6) and (4.7) where the characteristics of estimation error appears. The estimation error converges to zero if the matrix (A-KC) is negative definite. Observer gain decides the decay rate of the convergence.

After this explanation, it is possible to define the Kalman Filter and its working principles.

4.2.2.2 Kalman Filter and Extended Kalman Filter

Rudolf E. Kálmán is the founder of this type of filter. These filters are widely utilized in location and navigation systems and operation fields can be extended to other measurement systems. Accuracy and precision of the estimations under noisy or uncertain measurements and processes become satisfactory and reliable by these filters. Since this method uses a recursive approach, the discrete time state space model of kinematic bicycle model is introduced in Equations (4.8), (4.9), (4.10) and (4.11).

$$x_{t+1} = x_t + (v_t \cos(\theta_t))\Delta t \tag{4.8}$$

$$y_{t+1} = y_t + (v_t \sin(\theta_t))\Delta t \tag{4.9}$$

$$\theta_{t+1} = \theta_t + \left(\frac{v_t \tan(\varphi_t)}{l}\right)\Delta t \tag{4.10}$$

$$\varphi_{t+1} = \varphi_t + w_t \Delta t \tag{4.11}$$

Moreover, the procedure shown in Figure 4.2 and Equations from (4.1) to (4.7) are modified for recursive discrete time estimation process for linear systems. This new procedure can be explained through Figure 4.3.

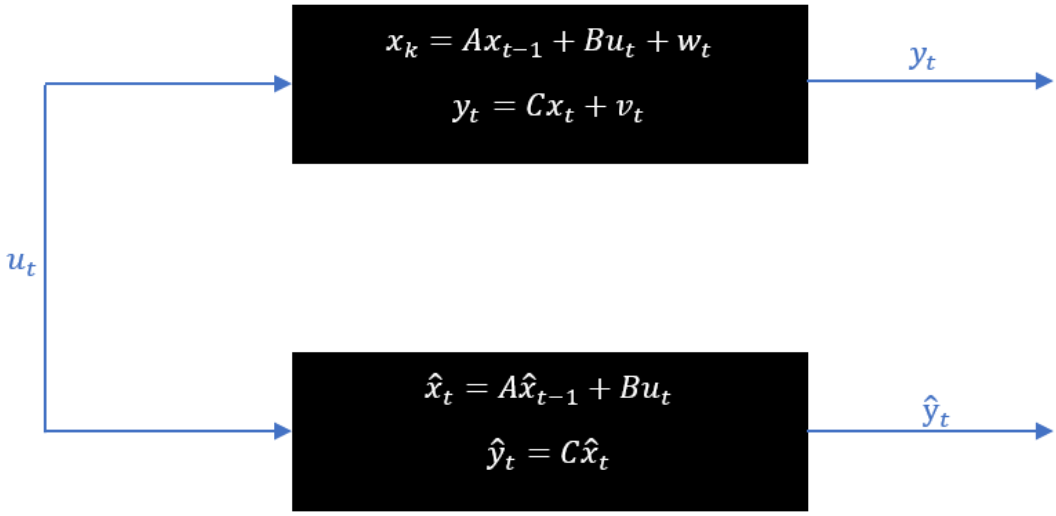

Figure 4.3 Discrete time estimation scheme for Kalman Filtering

In this figure discrete time estimation is represented. Notice that process and measurement noises are included to the model but not for the estimation calculation. These noises are considered as Gaussian functions for Kalman Filter operations. Therefore, they can be defined as random variables with means and covariances. Equations (4.12) and (4.13) represent zero mean Gaussian noises for process and measurement respectively.

$$w \sim \mathcal{N}(0, R) \tag{4.12}$$

$$v \sim \mathcal{N}(0, Q) \tag{4.13}$$

Where, Q and R are covariances of these distributions. Since these are independent random variables, Q and R matrices represent variances of each data point. Therefore, Equations (4.12) and (4.13) can be rewritten and expressed as in Equations (4.14) and (4.15).

$$w \sim \mathcal{N}(0, {\sigma_w}^2) \tag{4.14}$$

$$v \sim \mathcal{N}(0, {\sigma_v}^2) \tag{4.15}$$

An important property of Gaussian functions is that they stay in Gaussian form after linear transformations. Kalman Filter stresses out this fact by utilizing the scheme shown in Figure 4.3. Multiplication of two values shown in Figure 4.3 are sensor measurements and estimation of states. By multiplying these measurements, another Gaussian distribution can be found. Figure 4.4 shows this fact.

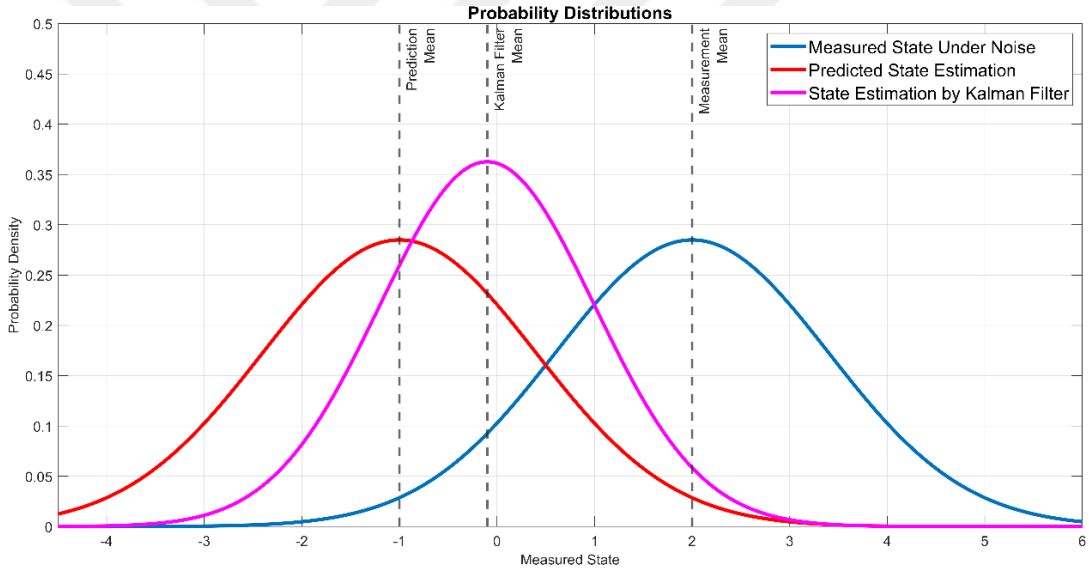

Figure 4.4 Probability distributions

Kalman Filter takes measurement data and estimations separately. Then by combining these informations, it creates another Gaussian distribution for the state. This operation is done by Kalman Filter equations. These equations contain two steps. First step is the prediction step and the second is the update step. Prediction step is expressed in equations (4.16) and (4.17).

$$\hat{x}_{t+1,t} = A\hat{x}_{t,t} + Bu_t \tag{4.16}$$

$$P_{t+1,t} = AP_{t,t}A^T + Q \tag{4.17}$$

Where subindices show when the variable take that value and it is calculated by using which time step data. P is called error covariance matrix and it contains variance of prior estimate. In other words, it can be called as estimate uncertainty. Moreover, update step contains measurements and corrects the prediction step. These calculations can be seen in equations (4.18), (4.19) and (4.20).

$$K_t = P_{t,t-1}C^T\left(CP_{t,t-1}C^T + R_t\right)^{-1} \tag{4.18}$$

$$\hat{x}_{t,t} = \hat{x}_{t,t-1} + K_t\left(y_t - C\hat{x}_{t,t-1}\right) \tag{4.19}$$

$$P_t = (I - K_tC)P_{t,t-1}(I - K_tC)^T + K_tR_t{K_t}^T \tag{4.20}$$

Where K is the Kalman Gain and it decides wether the measurements or the estimation is more reliable. By calculating these equations for each time step, Kalman Filter recursively filters the noise and allows a more reliable feedback for control systems.

Kalman Filter is also a sensor fusion algorithm. By providing multiple sensor data to Kalman Filter, it is possible to increase the accuracy of feedback signals to the system. Figure 4.5 shows an additional measurement for the calculation of probability density function obtained by Kalman Filter.

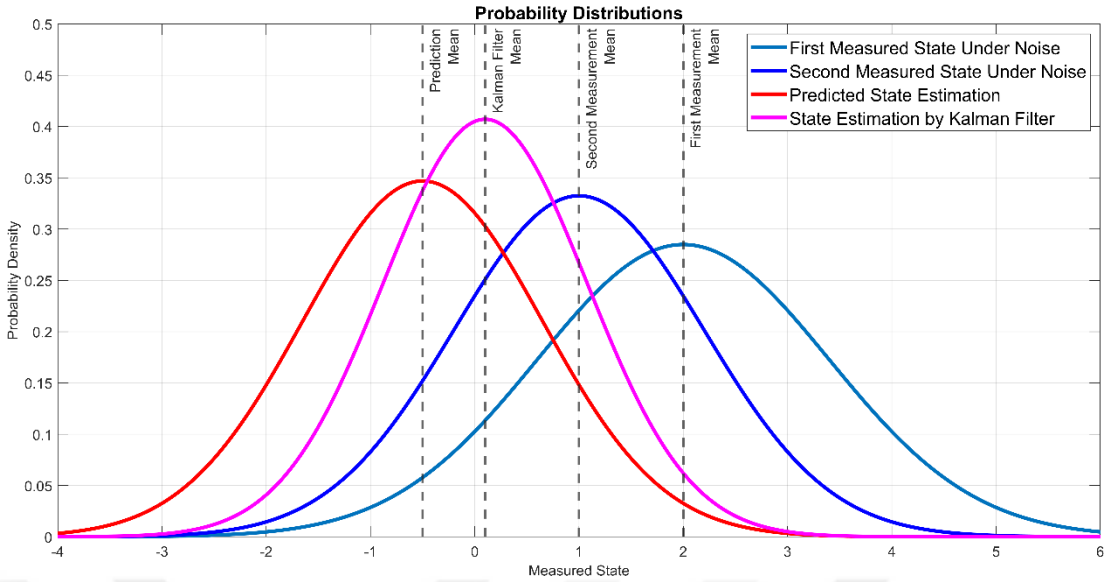

Figure 4.5 Probability distributions with additional measurement

As can be seen in Figure 4.5, with an additional measurement, the new probability density function would have a new form of Gaussian distribution.

Since kinematic bicycle model is a nonlinear system, a nonlinear approach is necessary for application. The main reason behind this consideration is that Gaussian distributions are not guaranteed to be Gaussian distributions after nonlinear contributions made by system. Therefore, system form is rewritten in equations (4.21) and (4.22).

$$x_t = f(x_{t-1}, u_t) + w_t \tag{4.21}$$

$$y_t = g(x_t) + v_t \tag{4.22}$$

For this purpose Extended Kalman Filter is introduced. Extended Kalman Filter linearizes a nonlinear sysytem at the mean values of current states for every time step. This linearizations are done by Jacobian matrices locally. The Jaboians are utilized in

previously introduced Kalman Filter Equations. Equations (4.23) and (4.24) show the calculation of Jacobians.

$$F = \frac{\partial f}{\partial x}\Big|_{\hat{x}_{t-1}, u_t} \tag{4.23}$$

$$G = \frac{\partial g}{\partial x}\Big|_{\hat{x}_t} \tag{4.24}$$

By this way locally linearized system can be expressed again as shown in Equations (4.25) and (4.26).

$$\Delta x_t = F \Delta x_{t-1} + w_t \tag{4.25}$$

$$\Delta y_t = G \Delta x_t + v_t \tag{4.26}$$

As mentioned previously, linearization by Jacobian method is not a completely suitable option for kinematic bicycle model because of its angular states. Because of this situation, angular state wrapping points at 2π can be considered risky. However, at any other point Extended Kalman Filter is expected to work in a satisfactory way.

To overcome this situation, Unscented Kalman Filters or Particle Filters can be employed. They use reference points of a distribution and propagate them through nonlinear system models. Particle Filters are even able to approximate non-Gaussian distributions. However, both techniques are computationally expensive. Therefore, Extended Kalman Filter is employed for this work. Next section presents simulation results and performance comparison of theoretically explained features.

CHAPTER FIVE
SIMULATIONS AND RESULTS

The performance of previously shown methods are simulated. Simulations are performed in MATLAB/Simulink environment. Simulation settings for all methods include time step of 0.001 s and ODE4 is the solver selection. Each simulation starts with definition of a reference trajectory and reference inputs. Then previously explained methods are performed. Noise and filtering contributions are added and details are given in next sections. Single parameter of kinematic bicycle model is the wheelbase. This value is taken 1.5 m. Additionally, steering angle is saturated in $\left[-\frac{\pi}{2}+0.5, \frac{\pi}{2}-0.5\right]$ rad for a realistic approach.

5.1 LQR Trajectory Tracking

First simulated method is the Linear Quadratic Regulator for a reference trajectory of a circle with radius of 5 m. This circular trajectory is aimed to be completed in 10 s and therefore linear velocity is decided to be π m/s. This calculation is made by using Equation (3.6). Start of the trajectory is the bottom of the circle and trajectory is created counterclockwise. Therefore, reference coordinates can be expressed as Equations (5.1) and (5.2).

$$x_{reference}(t) = 5\sin\left(\frac{\pi t}{5}\right) \tag{5.1}$$

$$y_{reference}(t) = -5\cos\left(\frac{\pi t}{5}\right) \tag{5.2}$$

By time derivation, as shown in previous chapters, it is possible to obtain reference inputs and reference states for this trajectory. This selection results a linear time invariant system for kinematic bicycle model represented in equation (3.23). Because, linear speed input is taken constant and a circle with constant radius imply a constant steering angle. The resulting error system can be seen in equation (5.3).

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \\ \dot{e}_4 \end{bmatrix} = \begin{bmatrix} 0 & \frac{\pi}{5} & 0 & 0 \\ -\frac{\pi}{5} & 0 & \pi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{5.3}$$

Moreover, weights matrices Q and R selected as in Equations (5.4) and (5.5).

$$Q = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 1000 \end{bmatrix} \tag{5.4}$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.5}$$

This selection is done based on the importance of errors. To be more clear, angular states are decided to be more important because of their impact on other states. Small deviations on angular states could create severe errors on position coordinates. Inputs are not penalized due to their coordinate transformation. By this selection, Equations (3.32) and (3.33) are solved and the gain matrix K is found. It can be seen in Equation (5.6).

$$K = \begin{bmatrix} 3.5604 & -2.1689 & -0.2213 & 0 \\ -0.2213 & 1.6032 & 31.7809 & 0 \\ 0 & 0 & 0 & 31.6228 \end{bmatrix} \quad (5.6)$$

Eigenvalues of the free error system and LQR controlled error system are calculated and compared. Both have four eigenvalues. Free system has its two eigenvalues at the origin and two complex conjugate eigenvalues at 0±0.6283i. However, eigenvalues defined by feedback action in Equation (3.31) are stable. They are located at -31.6228, -331.6212, -2.9531 and -0.767. Therefore, this approach can be considered stable. On the other hand, controllability and observability of this approach are guaranteed by structure. Figure A.1 shows the block scheme for LQR control of kinematic bicycle model.

Initial conditions for this simulation are selected as in equation (5.7).

$$\begin{bmatrix} x_{initial} \\ y_{initial} \\ \theta_{initial} \\ \varphi_{initial} \end{bmatrix} = \begin{bmatrix} 0 \\ -5 \\ 0 \\ 0 \end{bmatrix} \quad (5.7)$$

Figure 5.1 shows the tracking performance of LQR control technique.

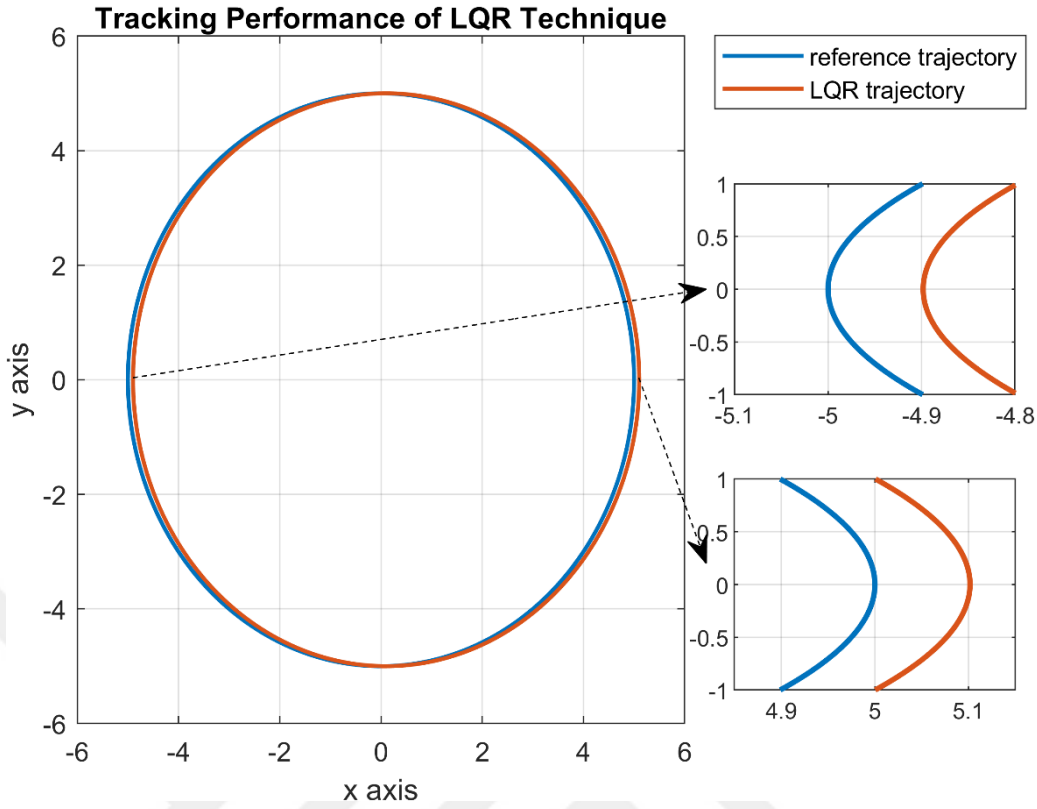

Figure 5.1 Tracking performance of LQR technique

As seen in Figure 5.1, LQR strategy with defined settings track the reference trajectory. However, tracking error from reference trajectory are evident for right and left parts of trajectory. At the same time, this control strategy is able to finish given reference trajectory in time. The performance can be evaluated by cumulative tracking error, mean tracking error and variances of tracking error from reference trajectory. Tracking error from reference trajectory is calculated at each time step by equation (5.8).

$$d_{trajectory} = \sqrt{\left(x_{reference} - x_{measured}\right)^2 + \left(y_{reference} - y_{measured}\right)^2} \qquad (5.8)$$

Table 5.1 gives these performance evaluations of LQR technique.

Table 5.1 Performance Evaluation of LQR Technique

Performance Metric	**Value**
Cumulative Tracking Error [m]	9.1285
Mean Tracking Error on x Axis [m]	-0.0572
Mean Tracking Error on y Axis [m]	0.0388
Variance of Error on x Axis [m^2]	0.0018
Variance of Error on y Axis [m^2]	0.0018

As Table 5.1 shows, relatively small tracking error from the trajectory results a large cumulative tracking error. Figure 5.2 shows these tracking errors with respect to time.

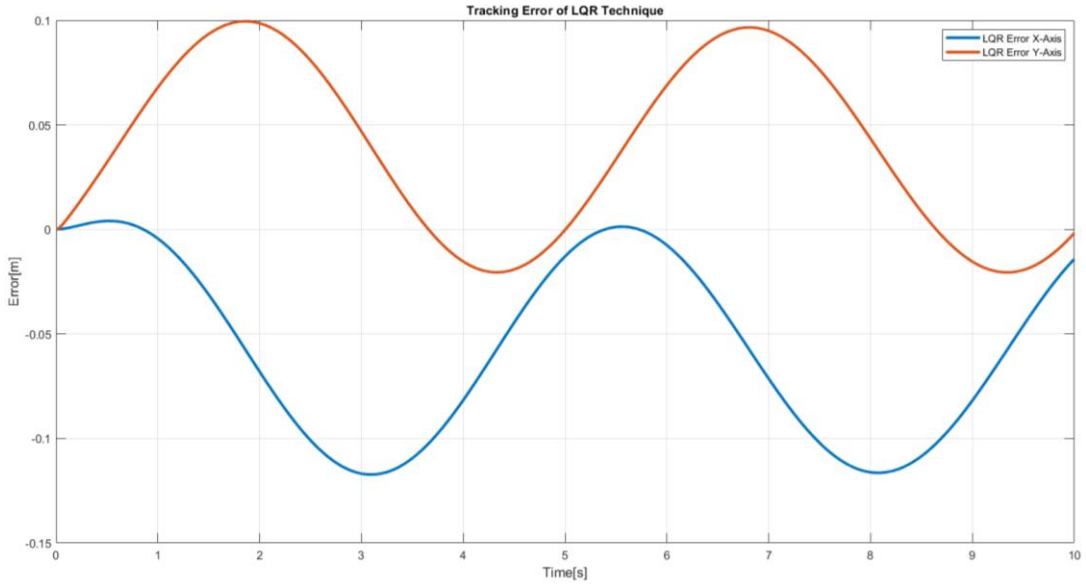

Figure 5.2 Tracking error of LQR technique on both cartesian coordinates

For these calculations noise is not included to the system. Performance under noisy measurement and filtering is shown in later subsections.

5.2 Lyapunov-Based Trajectory Tracking

Previously introduced Lyapunov-based stable control technique is considered for this section. Stability proof of the system given in equations (3.36), (3.37), (3.38), (3.39), (3.42), (3.43) and (3.44) are done in equations (3.45) and (3.46). The same trajectory as in LQR technique is taken as reference for this control technique. Same initial conditions are applied and tracking performance is evaluated again in statistical way. Figure A.2 shows the block scheme for this control approach. Controller gains are picked as $k_1=k_2=40$ and $k_3=50$.

Performance of tracking can be seen in Figure 5.3.

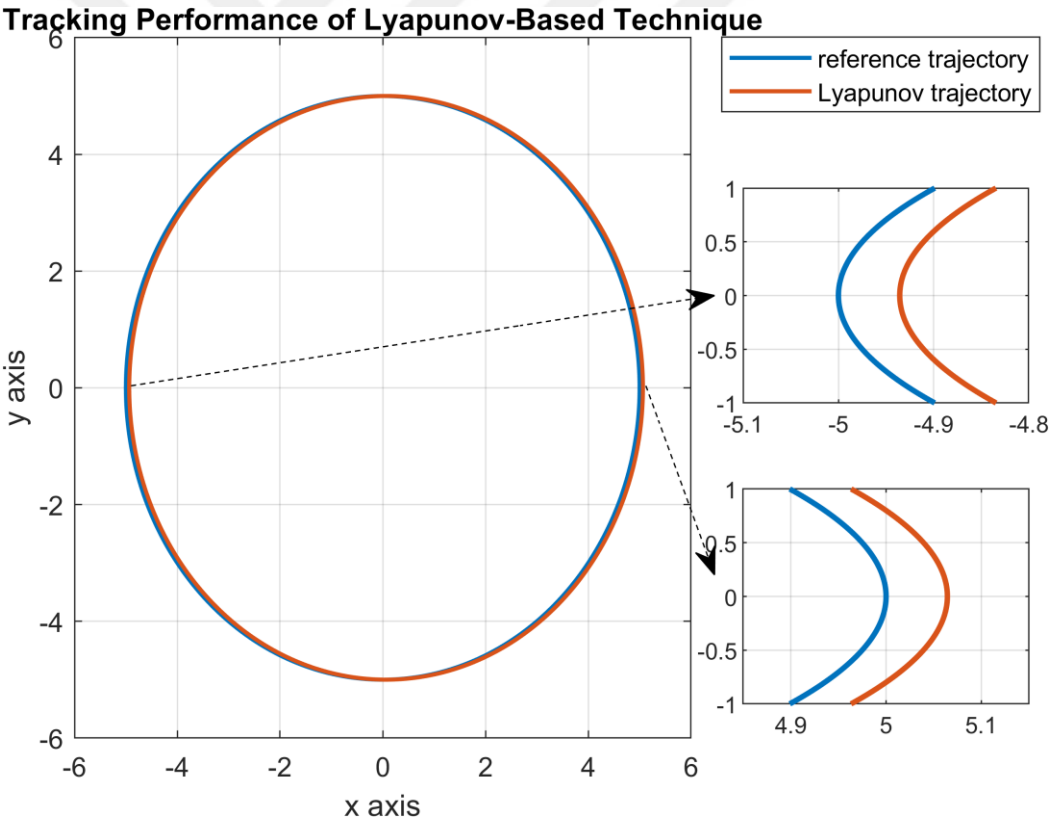

Figure 5.3 Tracking performance of Lyapunov-based technique

Lyapunov-based control approach is also able to track the given trajectory. However, like in LQR technique, tracking error from reference trajectory on left and

right sides are evident. On the other hand, this technique shows less tracking error in overall trajectory. Same performance metrics as shown in Table 5.1 are applied and results are collected in Table 5.2.

Table 5.2 Performance Evaluation of Lyapunov-Based Technique

Performance Metric	**Value**
Cumulative Tracking Error [m]	4.5506
Mean Tracking Error on x Axis [m]	-0.0322
Mean Tracking Error on y Axis [m]	3.0346×10^{-4}
Variance of Error on x Axis [m^2]	5.1758×10^{-4}
Variance of Error on y Axis [m^2]	5.1747×10^{-4}

Performance of Lyapunov-based control technique shows satisfactory results. Figure 5.4 shows the tracking error performance of this technique in time.

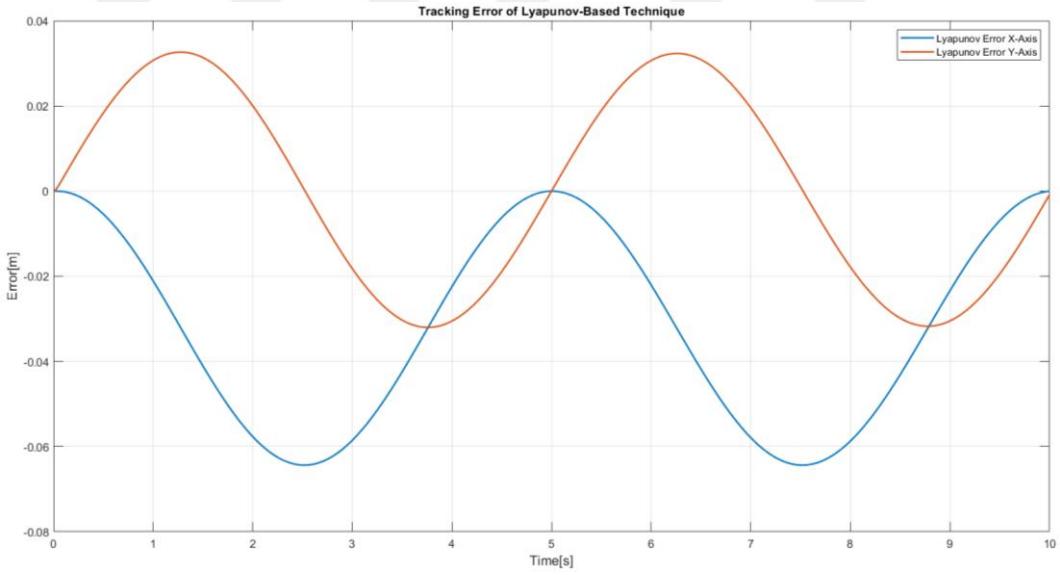

Figure 5.4 Tracking error of Lyapunov-based technique on both cartesian coordinates

Performance of this technique under noisy measurements are shown and evaluated in next subsections.

5.3 Dynamic Feedback Linearizing Control

This control technique is simulated for two different reference trajectories. First one is the same trajectory as previous cases and initial conditions given in previous simulations. Second one is another reference trajectory which is detailed later in this subsection. These simulations are done based on the extended system shown in Equations (3.91), (3.92), (3.93), (3.94), (3.95) and (3.96). The inputs to the system are shown in Equation (3.131) and the utilized coordinate transformation can be seen in Equation (3.132). Stability of the control technique is proven in previous sections.

Additional controller for linear dynamics is taken as another LQR with weights matrices given in equations (5.9) and (5.10).

$$Q = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 500 & 0 \\ 0 & 0 & 300 \end{bmatrix} \tag{5.9}$$

$$R = 1 \tag{5.10}$$

Since outputs are taken as cartesian coordinates x and y, by the structure shown in Equations (3.139) and (3.140), it is possible to say that first state is the cartesian coordinate itself for both chains and the next ones are the derivatives of these coordinates. Derivatives correspond to velocity and acceleration on x and y directions. Therefore, penalization of states are done by reducing importance. Inputs for both chains are not penalized again. The resulting control gain K can be seen in Equation (5.11).

$$K = [31.6228 \quad 41.6937 \quad 19.5803] \tag{5.11}$$

The original system has three eigenvalues at origin and LQR design placed all of them to left half plane. They are located at -17.2724 and -1.1539±0.7066i. This feedback is therefore considered stable. Figure A.3 shows the block scheme for this control technique.

As can be seen in Figure A.3, the number of system states are increased and two additional coordinate transformations are done. This also increase the necessary number of calculations to accomplish this technique. For this technique, two different initializations are necessary. First one is for extended kinematic bicycle model. The second one is for the LQR technique and it can be defined by the coordinate transformation. Equations (5.12) and (5.13) show initializations for extended kinematic bicycle model and for linearized coordinates. Note that the same objective holds as previous two control techniques. The trajectory is desired to be finished in 10 s with constant linear speed of π m/s and initial conditions of first four states are same as Equation (5.7).

$$\begin{bmatrix} x_{initial} \\ y_{initial} \\ \theta_{initial} \\ \varphi_{initial} \\ v_{initial} \\ a_{initial} \end{bmatrix} = \begin{bmatrix} 0 \\ -5 \\ 0 \\ 0 \\ \pi \\ 0 \end{bmatrix} \tag{5.12}$$

$$\begin{bmatrix} z_{1initial} \\ z_{2initial} \\ z_{3initial} \\ z_{4initial} \\ z_{5initial} \\ z_{6initial} \end{bmatrix} = \begin{bmatrix} 0 \\ \pi \\ 0 \\ -5 \\ 0 \\ 0 \end{bmatrix} \tag{5.13}$$

Figure 5.5 shows the tracking performance for dynamic feedback linearizing control technique for the previously defined reference trajectory. Note that both coordinate systems are plotted and tracking errors are evaluated as shown in previous tables.

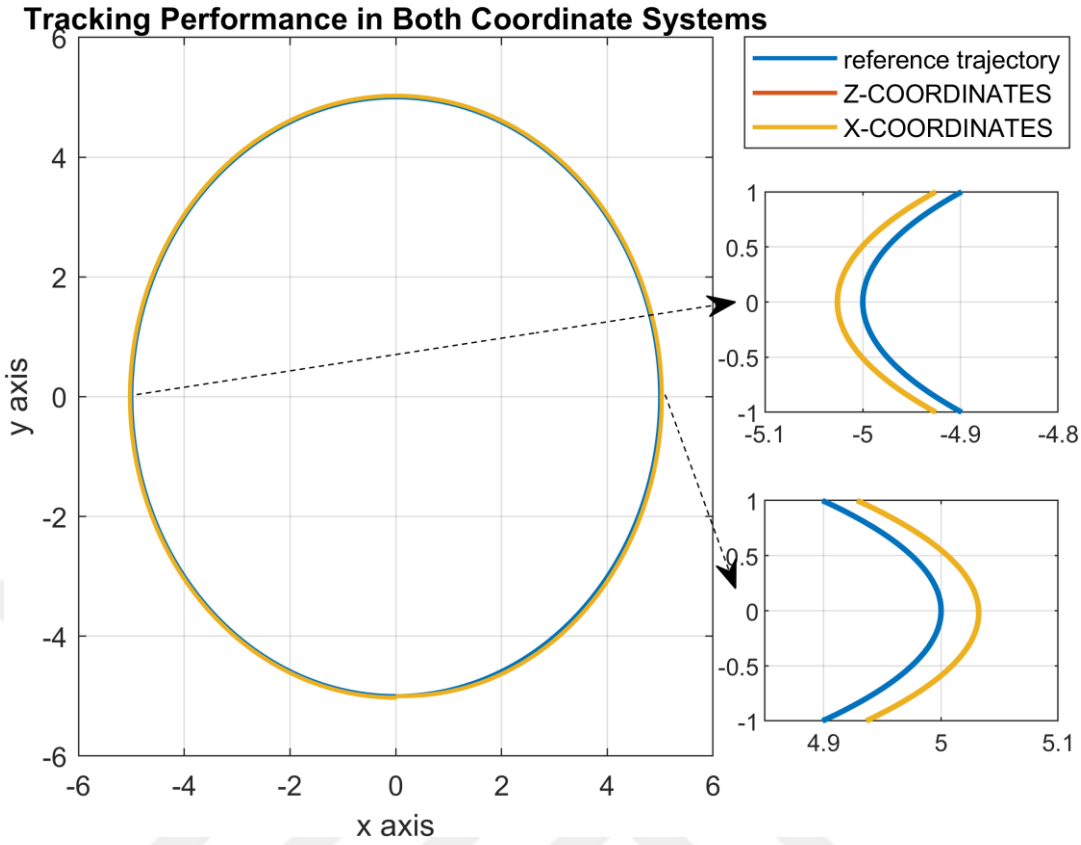

Figure 5.5 Tracking performance of dynamic feedback linearization in both coordinate systems

As can be seen from Figure 5.5, the dynamic feedback linearizing control technique performs well in all regions of the reference trajectory. By this outcome, the performance is evaluated in Table 5.3.

Table 5.3 Performance Evaluation of Dynamic Feedback Linearizing Control

Performance Metric	**Original Coordinates**	**Transformed Coordinates**
Cumulative Tracking Error [m]	3.2775	3.2775
Mean Tracking Error on x Axis [m]	0.0041	0.0041
Mean Tracking Error on y Axis [m]	0.0038	0.0038
Variance of Error on x(z_1) Axis [m^2]	4.2284×10^{-4}	4.2284×10^{-4}
Variance of Error on y(z_4) Axis [m^2]	6.2004×10^{-4}	6.2004×10^{-4}

Table 5.3 shows that transformed and original coordinates show the same characteristics in terms of tracking performance. Therefore, for next works only original coordinates will be shown. Cumulative tracking error is relatively the smallest compared to the previous techniques. The variance of tracking error with respect to both axes are smallest for this technique. Figure 5.6 shows the tracking error with respect to time. Since pairs of signals are identical for both coordinate systems, curves overlap for tracking error.

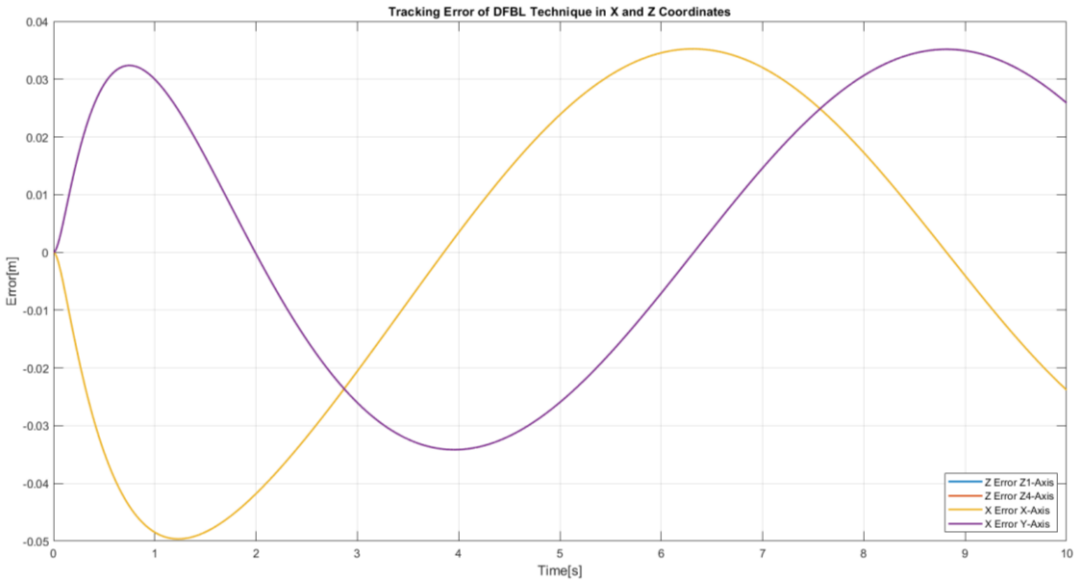

Figure 5.6 Tracking error of dynamic feedback linearization on both coordinates

To improve the performance of tracking, R value of LQR in linearized dynamics changed to 0.1 and simulations performed again. The results can be seen and compared with R equals to 1 case in Table 5.4. Since tracking is almost exact, tracking figure is not plotted.

Table 5.4 Effect of R Value Selection

Performance Metric	**R=1**	**R=0.1**
Cumulative Tracking Error [m]	3.2775	1.039
Mean Tracking Error on x Axis [m]	0.0041	0.0013
Mean Tracking Error on y Axis [m]	0.0038	0.0012
Variance of Error on x Axis [m^2]	4.2284×10^{-4}	4.2587×10^{-5}
Variance of Error on y Axis [m^2]	6.2004×10^{-4}	6.2238×10^{-5}

Effect of smaller R selection shows a better performance with respect to previous selection. However, this selection causes eigenvalues of the linearized system closer to the imaginary axis in complex plane. Therefore, this system would be more sensitive to uncertainties.

An alternative trajectory is selected for this technique and performance is evaluated. This new trajectory is a sinusoid starting from origin and evolving in a linear manner along x axis. Reference coordinates can be seen in Equations (5.14) and (5.15).

$$x_{reference}(t) = -5 + t \tag{5.14}$$

$$y_{reference}(t) = 2\sin\left(\frac{\pi t}{5}\right) \tag{5.15}$$

However, this simulation is initialized away from trajectory. The convergence is evaluated together with tracking performance. Initial conditions can be seen in Equations (5.16) and (5.17).

$$\begin{bmatrix} x_{initial} \\ y_{initial} \\ \theta_{initial} \\ \varphi_{initial} \\ v_{initial} \\ a_{initial} \end{bmatrix} = \begin{bmatrix} -5.5 \\ 4 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{5.16}$$

$$\begin{bmatrix} z_{1initial} \\ z_{2initial} \\ z_{3initial} \\ z_{4initial} \\ z_{5initial} \\ z_{6initial} \end{bmatrix} = \begin{bmatrix} -5.5 \\ 1 \\ 0 \\ 4 \\ 0 \\ 0 \end{bmatrix} \tag{5.17}$$

Simulation is done for 20 s and results can be seen in Figure 5.7.

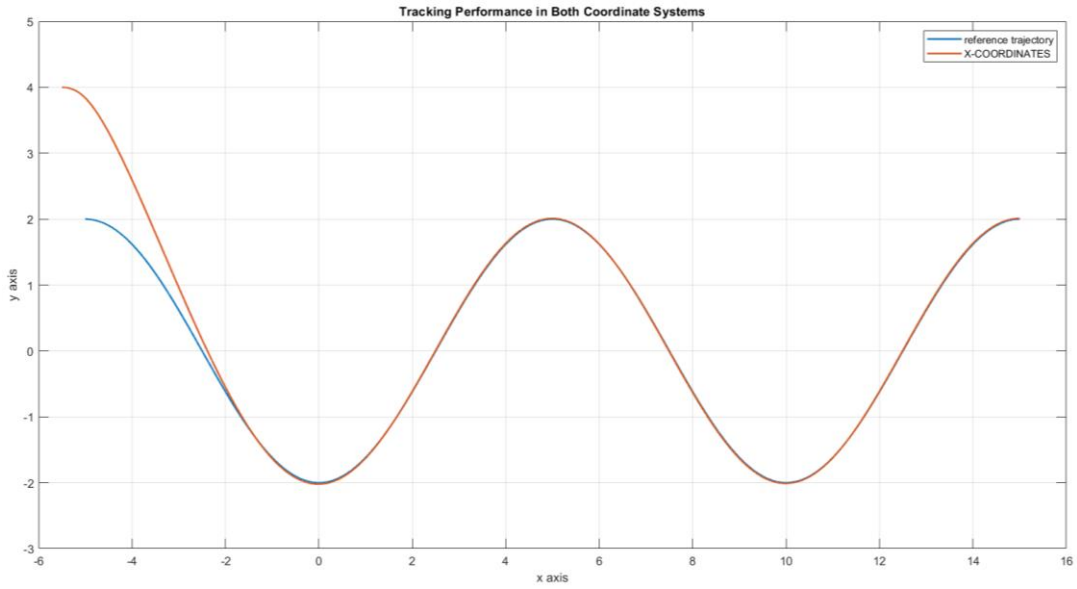

Figure 5.7 Convergence and tracking performance

By these results, it is possible to say that the best tracking controller performance is obtained by dynamic feedback linearization method.

Noisy measurement cases are shown in next subsections and filtering of noise through Extended Kalman Filter is performed.

5.4 Pure Pursuit Tracking

Pure Pursuit Algorithm is subjected to the kinematic bicycle model for tracking of the same circular trajectory given in the beginning of this section. However, defining a trajectory for this approach is not possible and therefore, this task is considered as a path tracking rather than trajectory tracking. The main reason behind this is the definition of the waypoints. Instead of defining a function of time, the waypoints are defined by lookahead distance parameter of Pure Pursuit Algorithm.

For reference input of feedback scheme shown in Figure A.4, all waypoints of the circular trajectory are previously calculated and stored in a 2-by-10001 matrix. Where 2 rows represent x and y values in space and 10001 columns represent the calculated value of x and y coordinates for each time instant. Since simulation time is set to 10 s, a time step of 0.001 s results 10001 data points.

For this simulation lookahead distance is taken as 2 m. The algorithm searches for a point at the same lookahead distance. If there does not exist any waypoint at that lookahead distance, then it takes the points which are 0.002 m away from the lookahead distance to take an average point for calculations. This distance is calculated through Equation (5.18). Where, subindex waypoints correspond data points generated by prior knowledge on the path.

$$d_{waypoints} = \sqrt{\left(x_{waypoints} - x_{measured}\right)^2 + \left(y_{waypoints} - y_{measured}\right)^2} \quad (5.18)$$

Note that this is only done for forward motion case. Next step is to perform calculations shown in Equations (3.144), (3.145), (3.146), (3.147), (3.148), (3.149), (3.150) and (3.151). For these calculations, desired linear speed and rate of change on steering are taken as π m/s and 0 rad/s respectively. Additionally, control gain and k_w is taken 50. Figure 5.8 shows the performance of Pure Pursuit Algorithm for convergence and tracking.

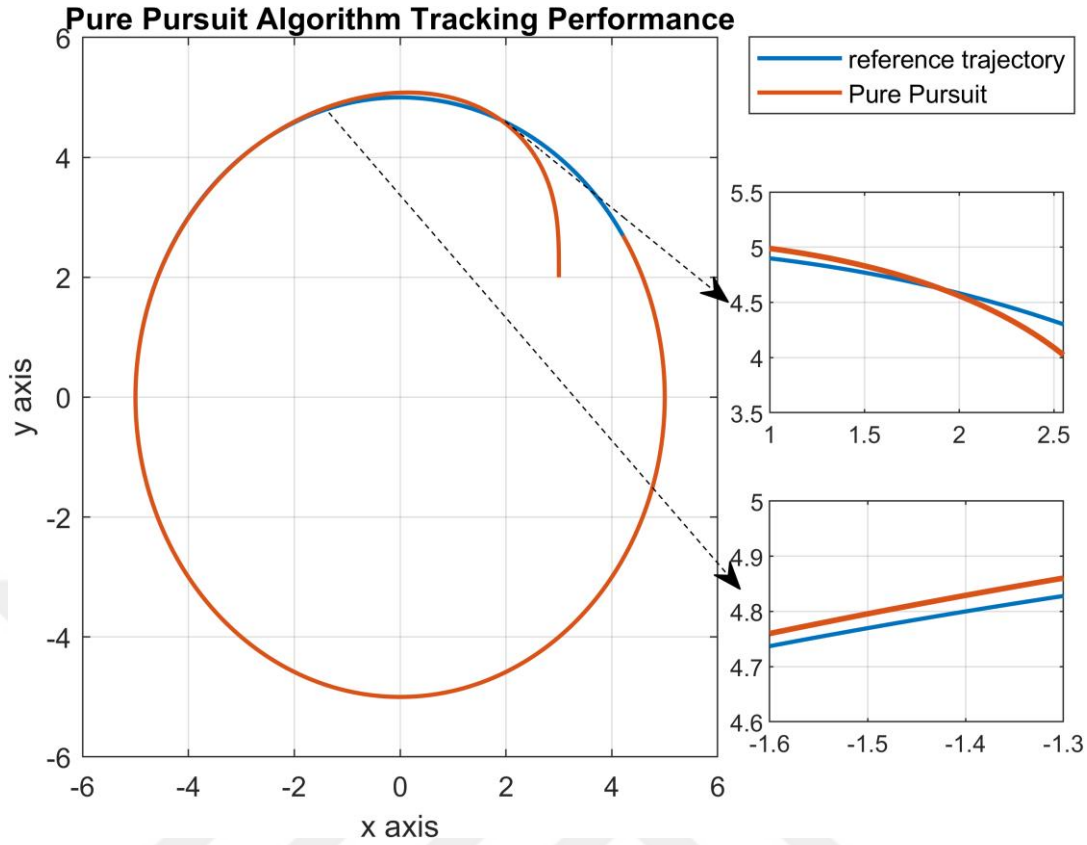

Figure 5.8 Pure Pursuit Algorithm tracking lookahead distance 2 m

For this simulation initial conditions are taken as shown in Equation (5.19).

$$\begin{bmatrix} x_{initial} \\ y_{initial} \\ \theta_{initial} \\ \varphi_{initial} \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ \frac{\pi}{2} \\ 0 \end{bmatrix} \tag{5.19}$$

For this initialization there were a set of points inside lookahead distance. Therefore, this initialization is viable for Pure Pursuit Algorithm. Additionally, it is possible to say that tracking is satisfactory unlike convergence period. Moreover, to evaluate the effect of lookahead distance selection, this value is taken as 1.5 m and 3 m. Figure 5.9 shows all selected lookahead distance simulations together.

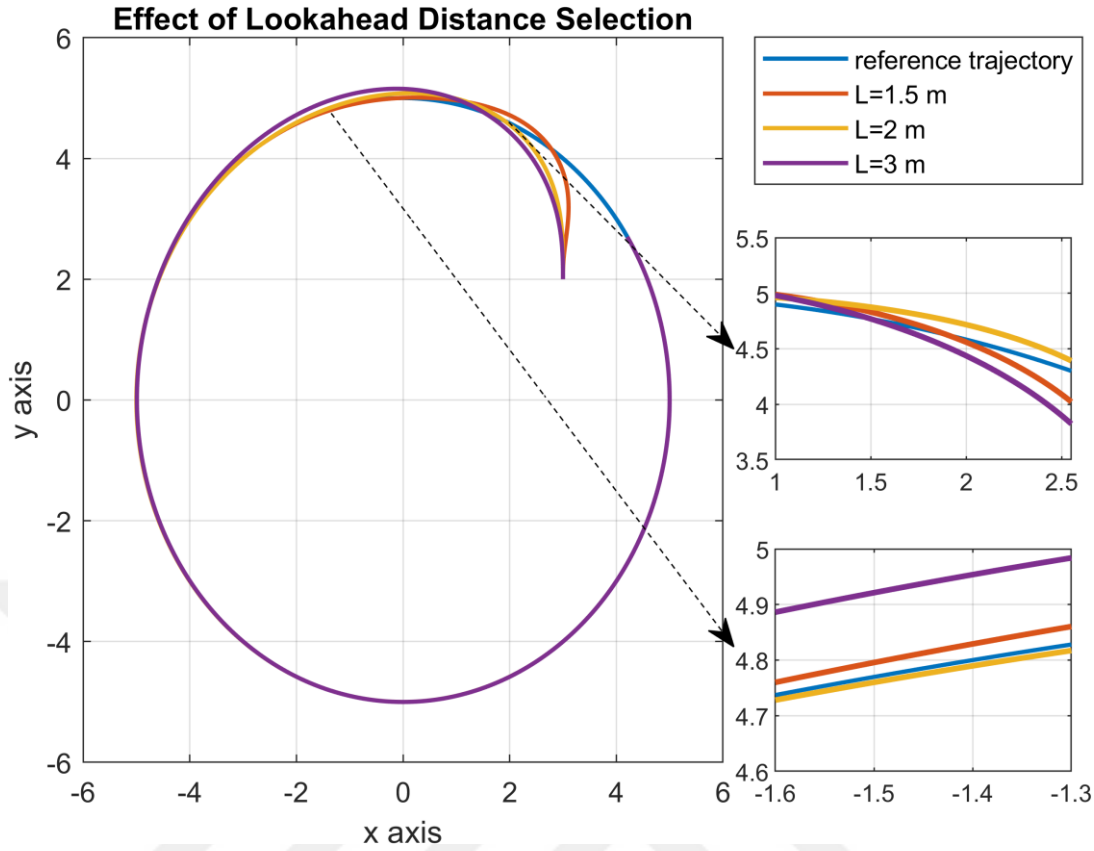

Figure 5.9 Effect of lookahead distance selection

As lookahead distance gets smaller, the movements get more aggressive. Figure 5.9 shows that for lookahead distance selection of 1.5 m results a faster convergence to the path than the other selections. However, a large overshoot is apparent for more aggressive movements. This can be taken as a preliminary result and can be used for designing hybrid control techniques together with other tracking controllers.

5.5 Pure Pursuit Obstacle Avoidance

This section takes Pure Pursuit Algorithm and uses it as an obstacle avoidance algorithm because of the satisfactory results shown in Figure 5.9. Figure 4.1 shows the measurement system for the kinematic bicycle model type robots or vehicles. By referencing to that figure, it is possible to define a new Pure Pursuit Algorithm for obstacle avoidance task.

Obstacle avoidance is one of the most important topics in autonomous navigation. The subject must avoid any space taken by the obstacles or environmental boundaries. To define this concept in detail, configuration spaces are studied in the literature. These spaces are constructed by states of the kinematic bicycle model. Any occupied space in configuration spaces are created by modifying the original map of the environment via applying all possible configurations of the model in that space. Therefore, a four-dimensional configuration space includes four different submaps for each state and their combination creates occupied and free spaces. A model is considered to move only in free space.

This newly created space consists bigger obstacles with respect to their original sizes. Therefore, a greater occupied area is present in cartesian plane. When all possible maneuvering moves of kinematic bicycle model are considered, an additional safety region is created and a switching mechanism is added to the system. This switching system allows to switch between baseline tracking controller and obstacle avoidance strategy. Therefore, this section can be considered a hybrid control approach for the kinematic bicycle model.

Main idea of this approach is explained by next arguments. Measurements of the relative position and bearing to any point in space with respect to a reference point can be derived by sensor measurements. Therefore, by measuring the states of the kinematic bicycle model and then measuring external features such as relative distances and bearings define a center for a circle of inaccessible or dangerous regions for the kinematic bicycle model. The calculation of relative distance and bearing can be expressed as Equations (5.20) and (5.21).

$$d_{obstacle} = \sqrt{(x_{obstacle} - x_{measured})^2 + (y_{obstacle} - y_{measured})^2} \tag{5.20}$$

$$\xi_{obstacle} = \tan^{-1}\left(\frac{y_{obstacle} - y_{measured}}{x_{obstacle} - x_{measured}}\right) \tag{5.21}$$

By this way it is possible to create an alternative path for the trajectory intersecting with a dangerous zone defined by an obstacle. Figure 5.10 shows the dangerous zone created by a stationary point obstacle and alternative waypoint generation on the edge of the same dangerous zone. A stationary point obstacle is taken for the simplicity of representation.

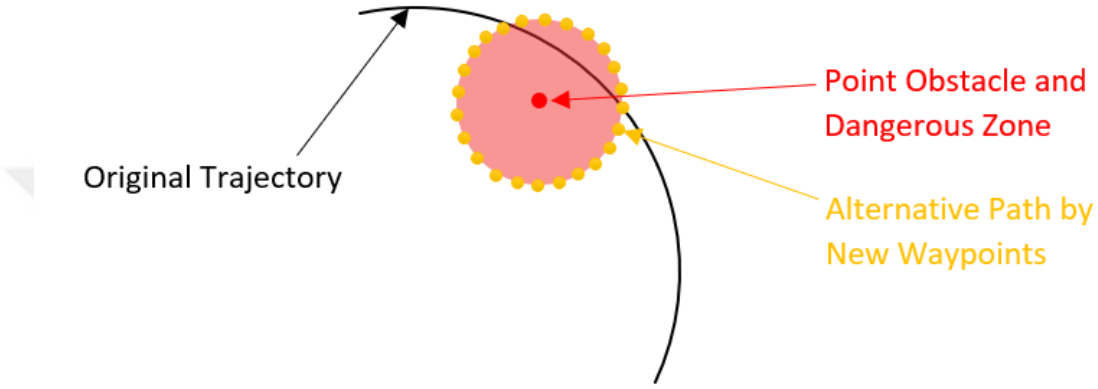

Figure 5.10 Dangerous zone defined by an obstacle

Dangerous zone is defined by parameters of kinematic bicycle model and additional considerations. The wheelbase of the model is 1.5 m. Additionally, a 0.5 m distance is added to include the dimensions of wheels to be attached. A total of 2 m radius circle is defined as dangerous zone for the kinematic bicycle model to enter. Therefore, after detecting an obstacle at a certain relative range and bearing, Pure Pursuit Algorithm for obstacle avoidance activates and creates additional waypoints of an alternative path. These waypoints are tracked as in previous section and this circular path is tracked as long as the obstacle is ahead of the kinematic bicycle model. Unlike the previous section, this strategy needs a smaller lookahead distance. Otherwise, the model might not find a feasible set of points to track. Results including obstacle avoidance is presented in hybrid control study.

5.6 Tracking Controller Performances Under Noisy Measurement and Filtering

For tracking task, kinematic bicycle model needs measurements on its four states. These measurements are assumed to be done by GPS for global position, gyroscope/IMU for orientation angle and potentiometer for steering angle. Since all states are measured by different measurement systems, the measurements can be considered uncorrelated. Moreover, each measurement would be done independently for any time instant. Therefore, resulting covariance matrix of the noise can be shown as in Equation (5.22).

$$S_1 = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 0.15 & 0 \\ 0 & 0 & 0 & 0.15 \end{bmatrix} \tag{5.22}$$

Values on this covariance matrix are decided by the reference papers given in last section. To be more clear, GPS data is reliable within approximately 10 m around common city areas without unusual high buildings. On the other hand, the angular measurements done by gyroscope and potentiometer are reliable within approximately 0.15 rad. This matrix is necessary for defining the measurement error. Assumed measurements do not carry any drift or bias. Figure 5.11 show the additive noise of the measured states for LQR, Lyapunov and dynamic feedback linearizing control techniques. Note that in the block of measurement noise, the additive noise components are organized as a column vector.

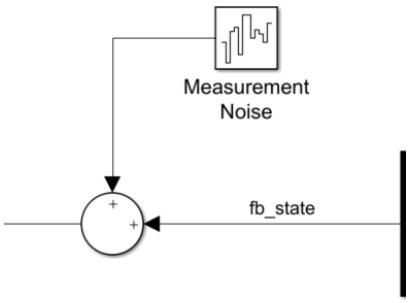

Figure 5.11 Additive measurement noise on feedback arm

These additive noise effects are added to feedback vector. LQR and Lyapunov-based techniques are not capable to follow a reference trajectory under this noise.

A similar approach is done for dynamic feedback linearization technique. However, additional measurements define a new covariance matrix and therefore a new additive noise vector on the feedback arm. These additional measurements are done by IMU's accelerometers. It is assumed that integrating the acceleration data is always done correctly. Therefore, both states are measured with the same noise. On the other hand, incremental encoders can be utilized in order to have another on-board sensor. Equation (5.23) shows the noise covariance matrix for 6-dimensional technique.

$$S_2 = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.15 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.05 \end{bmatrix} \tag{5.23}$$

These additive noise effects are added to feedback vector of dynamic feedback linearizing control block scheme. This technique is not able to cope with the high level of noisy feedback on positional information.

Under the assumption of Gaussian and white noise with uncorrelated measurements, an Extended Kalman Filter is appropriate to apply for both cases. Both

noisy cases need an Extended Kalman Filter block on their feedback arm. Figures A5 and A6 show block schemes of noisy measurement included LQR and Lyapunov-based techniques with Extended Kalman Filter.

An important feature to underline here is the zero-order hold blocks. Since Extended Kalman Filter needs discrete time signals, discretization of continuous simulation data is done by this way. Necessary initializations for Extended Kalman Filter can be considered as initial states of the kinematic bicycle model, initialization covariance and measurement noise covariance. System is linearized through its Jacobian at each time step. Initial state is the same as given in Equation (5.7). Noise covariance matrices are given in Equations (5.22) and (5.23). The initialization covariance is taken 10^{-6}. Reference trajectory taken the same as in Equations (5.1) and (5.2). Figure 5.12 shows tracking of LQR and Lyapunov-based controllers under noisy measurement with Extended Kalman Filter.

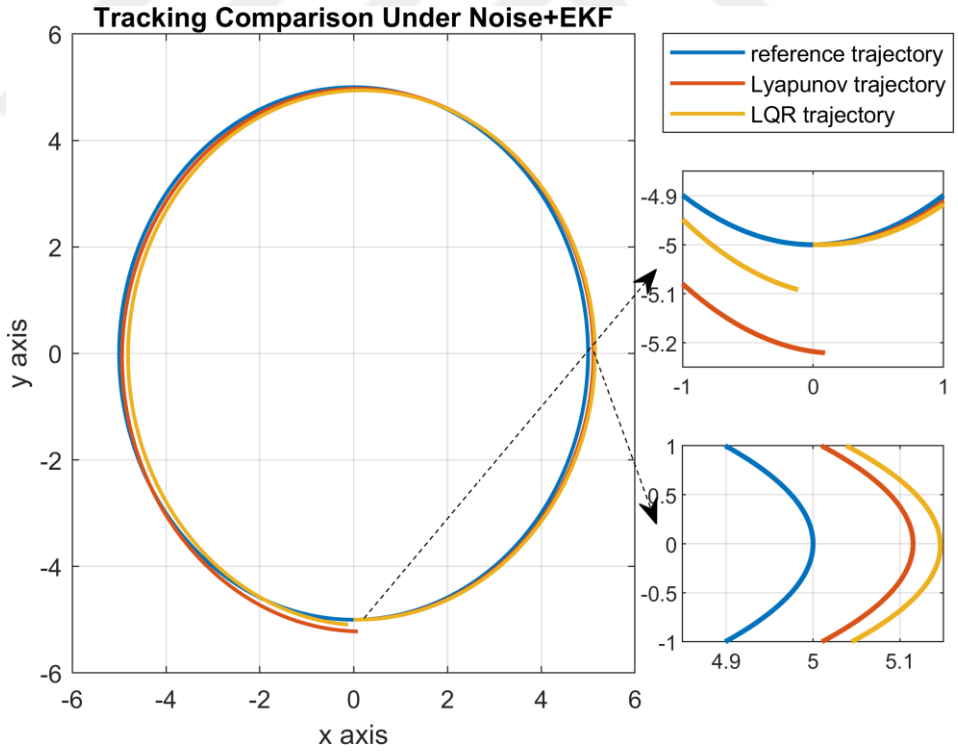

Figure 5.12 Tracking performance of LQR and Lyapunov-based controllers under noisy measurement and Extended Kalman Filter application

This figure shows that both techniques suffer severely from noisy measurements. Until the top of circular shaped trajectory, both control strategies perform similar. However, in the second half of the circle LQR controller is not able to follow the trajectory as Lyapunov-based controller. On the other hand, LQR controller has less final error at the end of the trajectory. Therefore, the same table is created and inspected as in previous cases.

Table 5.5 shows that LQR strategy outperforms Lyapunov-based controller in most of the performance metrics. This is an expected result due to forms of feedback terms in both techniques. Lyapunov-based controller is more sensitive to noisy measurements because of the definition of input terms.

Table 5.5 Comparison of Performance Under Noise and Extended Kalman Filter

Performance Metric	**Lyapunov**	**LQR**
Cumulative Tracking Error [m]	23.1095	20.9193
Mean Tracking Error on x Axis [m]	0.0514	-0.0106
Mean Tracking Error on y Axis [m]	-0.0431	0.051
Variance of Error on x Axis [m^2]	0.0215	0.0176
Variance of Error on y Axis [m^2]	0.0275	0.0235

Figures 5.21 and 5.22 show each state measurement and estimation after Extended Kalman Filter for Lyapunov-based and LQR control techniques respectively.

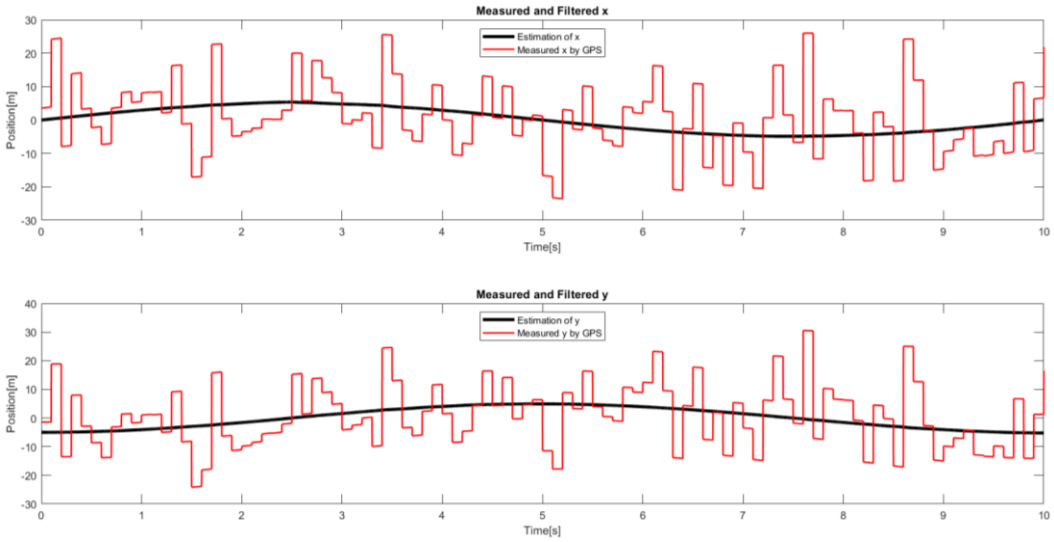

Figure 5.13 Measured and estimated states of Lyapunov-based Controller

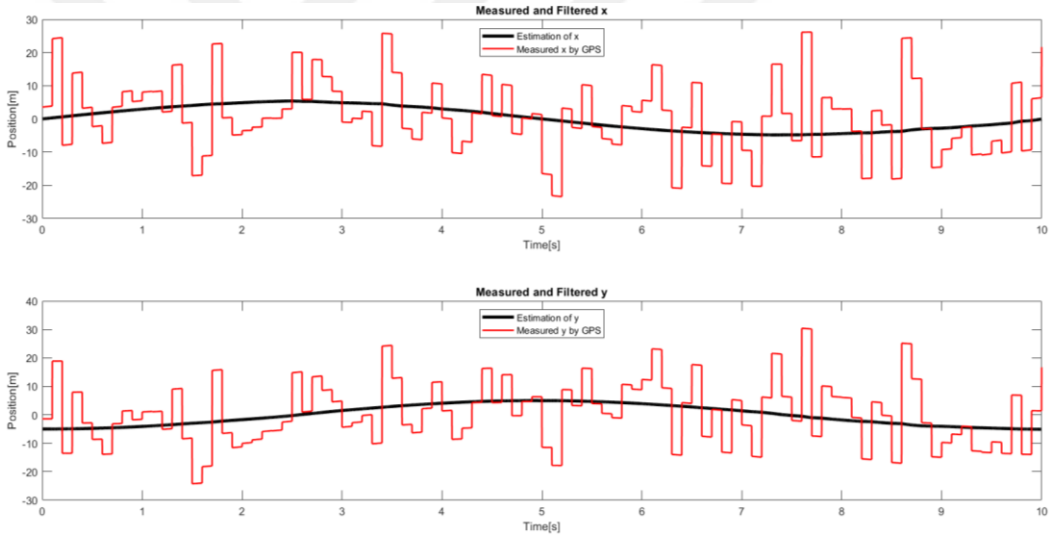

Figure 5.14 Measured and estimated states of LQR controller

In Figures 5.21 and 5.22 red lines correspond to measured noisy states and black lines correspond to estimated states after Extended Kalman Filter. Note that position information is in meters and angular information is in radians. To enhance the performance of this filtering technique, an additional measurement based on landmarks is included. Landmark positions are known and stationary. 4 different landmarks are added to the workspace of 4-dimensional case. These landmarks are placed on 3-meter perimeter circle with 90° difference by starting at 270°. The landmark-based localization measures range and bearing between each landmark and

rear wheel of kinematic bicycle model at each time instant. By taking the closest landmark as the measurement point, it creates an additional localization information. However, this measurement cannot give information on orientation angle and steering angle of the model. Measurement calculations can be given as in Equations (5.24), (5.25) and (5.26).

$$d = min\sqrt{\left(x - L_{x_i}\right)^2 + \left(y - L_{y_i}\right)^2}, i = 1,2,3,4 \tag{5.24}$$

$$\beta = \tan^{-1}\left(\frac{y - L_y}{x - L_x}\right) \tag{5.25}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} L_x + d\cos(\beta) \\ L_y + d\sin(\beta) \end{bmatrix} \tag{5.26}$$

Note that Equations (5.24) and (5.25) represent a measurement based on a LIDAR mounted on the kinematic bicycle model. Therefore, it is not affected by GPS noise. On the other hand, Equation (5.26) results a global position information. For this measurement, noise covariance matrix is selected as in Equation (5.27).

$$S_3 = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} \tag{5.27}$$

This noisy measurement is included to Extended Kalman Filter equations and simulation is done again for the same settings. Tracking performance comparison can be seen in Figure 5.15.

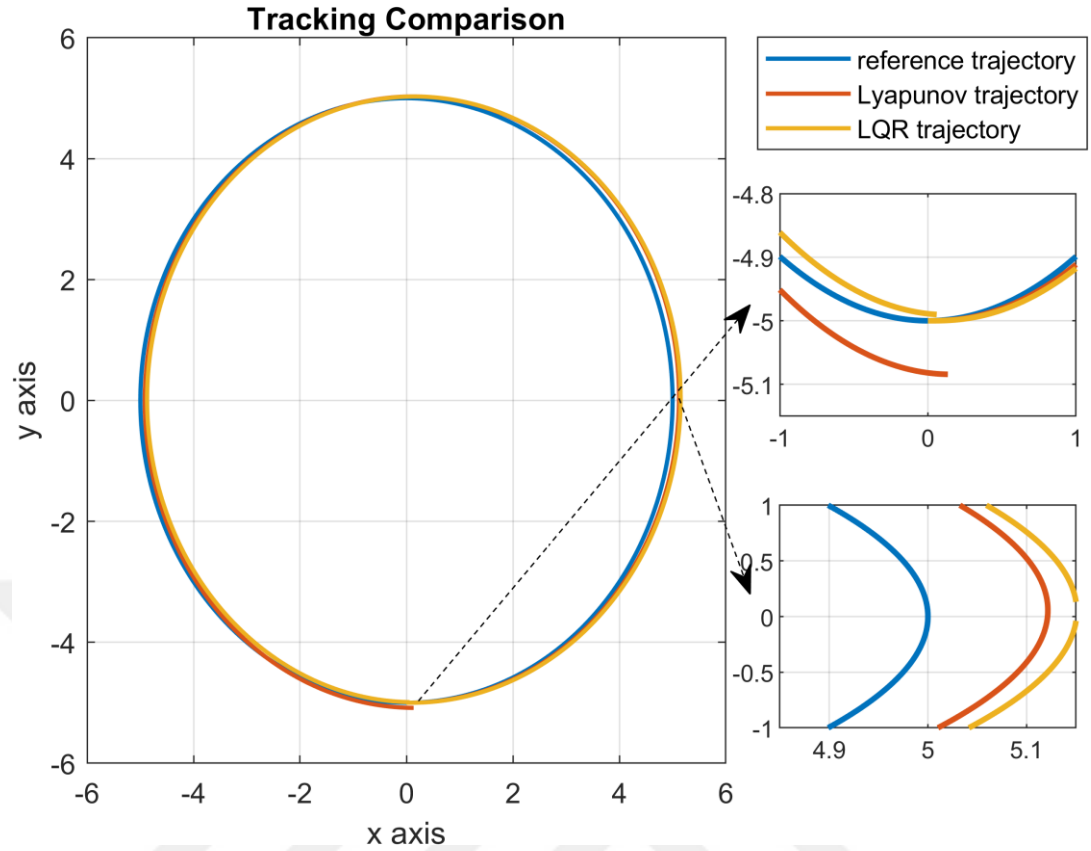

Figure 5.15 Tracking performance of LQR and Lyapunov-based controllers under noisy measurement and Extended Kalman Filter application with landmark-based localization

Evidently, both control approaches performed better than previous case. However, the statistical evaluation is necessary and same performance evaluation table is created. Results can be seen in Table 5.6.

Table 5.6 Comparison of performance under noise and Extended Kalman Filter with landmark-based localization

Performance Metric	**Lyapunov**	**LQR**
Cumulative Tracking Error [m]	12.4392	12.9427
Mean Tracking Error on x Axis [m]	-0.0472	-0.089
Mean Tracking Error on y Axis [m]	0.0038	0.0441
Variance of Error on x Axis [m^2]	0.0042	0.0038
Variance of Error on y Axis [m^2]	0.0091	0.0031

Application of the sensor fusion with landmarks is performing better than previous case. Both control approaches perform similarly and one is not outperforming the other in any performance metric. However, this type of measurements needs additional installments of landmarks and this work assumes that all landmarks are visible for the LIDAR of the model at any time instant. Measurements and estimations for x and y states for Lyapunov and LQR techniques are presented in Figures 5.24 and 5.25.

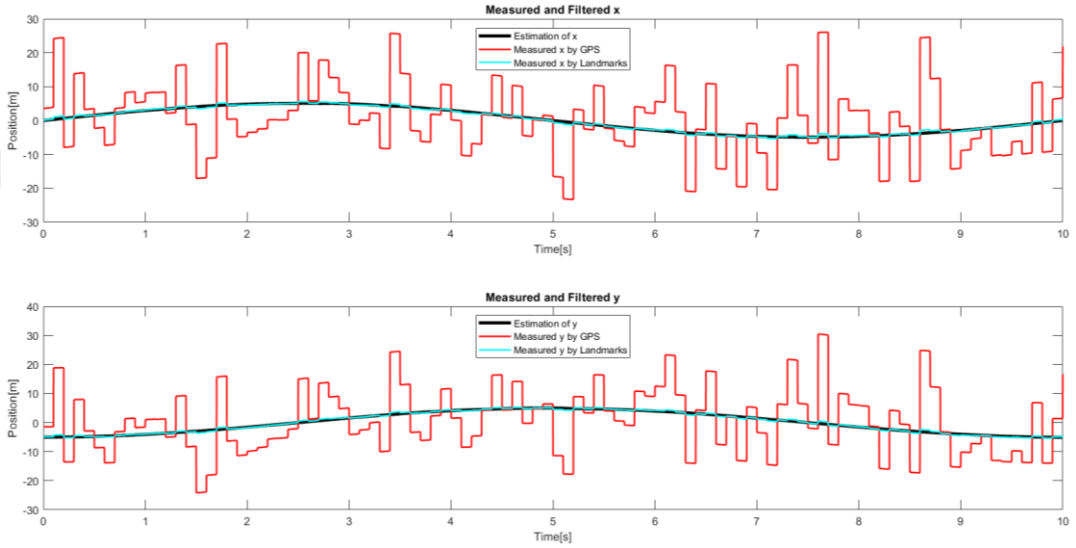

Figure 5.16 Measured and estimated cartesian coordinates of Lyapunov-based controller

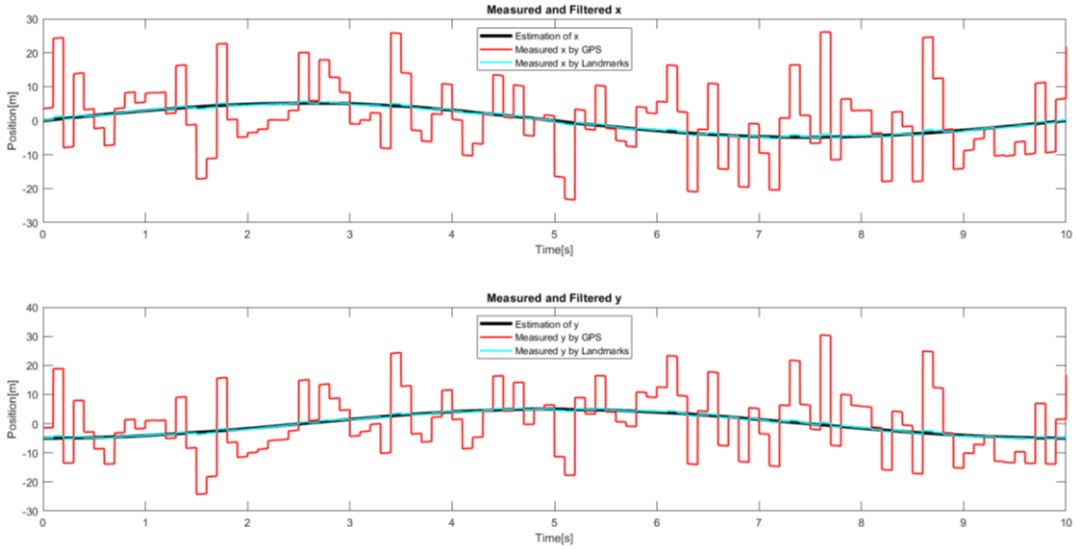

Figure 5.17 Measured and estimated cartesian coordinates of LQR controller

As said previously, higher accuracy LIDAR measurement increased the performance of estimation of the Extended Kalman Filter.

Similar considerations are done for dynamic feedback linearizing control. 6-dimensional case utilizes the Equation (5.23) for error covariance matrix. Moreover, Figure A.7 shows the block scheme with noise addition and Extended Kalman Filter in feedback arm.

Simulations are done for the same trajectory as given in Equations (5.1) and (5.2). For this trajectory, initialization given by Equations (5.12) and (5.13) are applied. LQR weights matrices are taken as in Equation (5.9) for Q and 0.1 for R. Figure 5.18 shows the first results of tracking performance under mentioned settings.

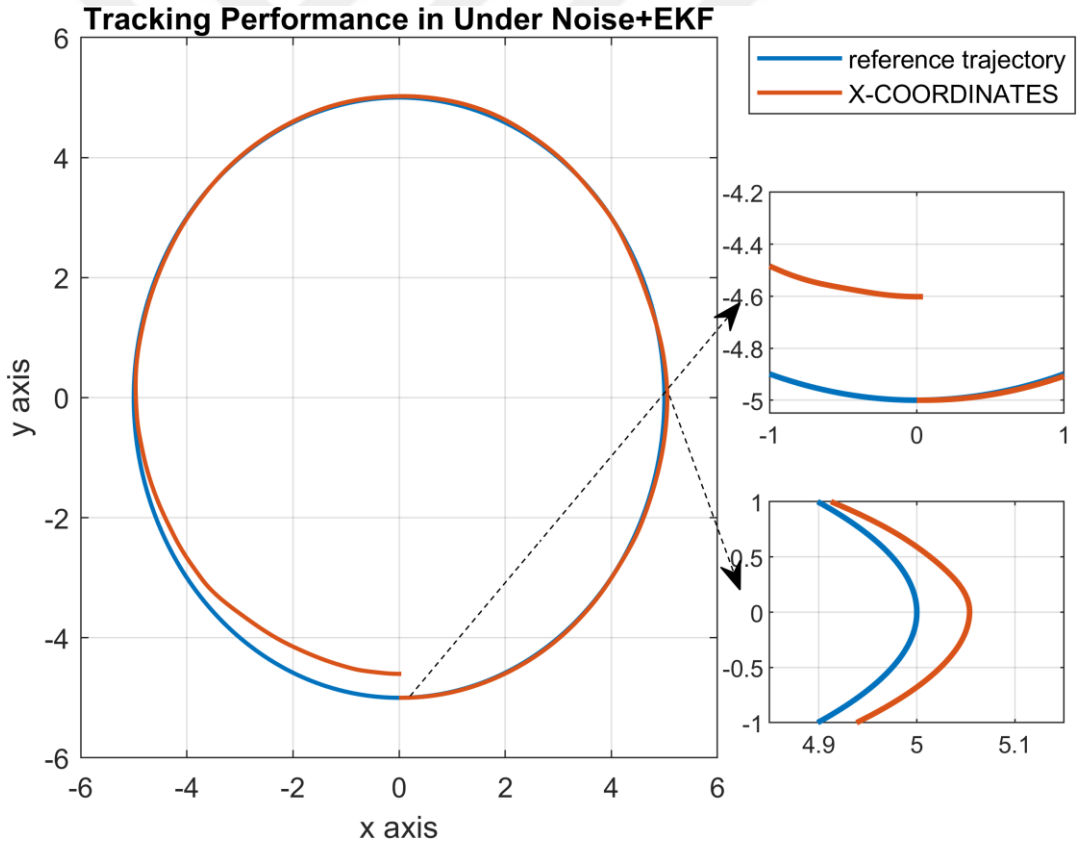

Figure 5.18 Tracking performance under noisy measurement with Extended Kalman Filter

Dynamic feedback linearizing control technique utilize many complex multiplication and division operations. Therefore, it is severely suffering from noisy

measurements. Moreover, it can be said that dynamic feedback linearization technique is even more sensitive to noise if compared to previously inspected techniques. However, it is able to follow the reference trajectory in three quarters. Main deviation occurs in last quarter of the circular trajectory. Therefore, it is necessary to evaluate performance of this controller by the statistical performance metrics and filter performance. Figure 5.19 shows measured and estimated values of the states.

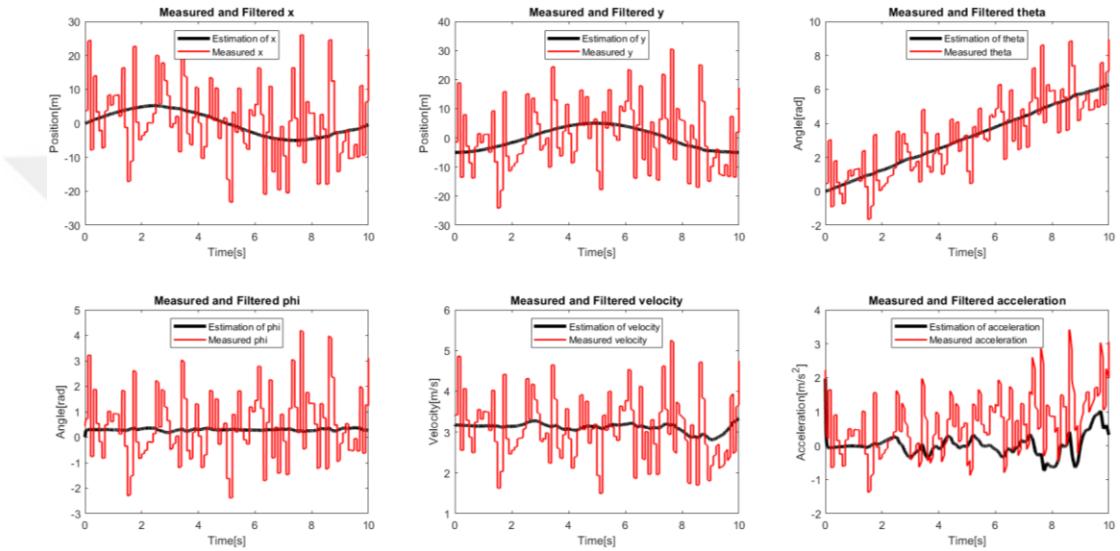

Figure 5.19 Measured and estimated states of dynamic feedback linearizing controller

Figure 5.19 shows that acceleration and speed states are not tracked perfectly by Extended Kalman Filter. Therefore, this failure of tracking by the filter results the deviation.

Table 5.7 shows performance metrics and compares the results by including the same landmark-based localization.

Table 5.7 Effect of performance for landmark-based localization

Performance Metric	**No Landmark Measurement**	**With Landmark Measurement**
Cumulative Tracking Error [m]	28.4058	2.4785
Mean Tracking Error on x Axis [m]	-0.0578	-0.0038
Mean Tracking Error on y Axis [m]	-0.1875	0.0017
Variance of Error on x Axis [m^2]	0.0026	1.1371×10^{-4}
Variance of Error on y Axis [m^2]	0.0396	4.8339×10^{-4}

Table 5.7 shows that landmark-based localization is increased performance visibly. Sensor fusion with smaller error covariance measurement increase the precision of estimation. Figure 5.20 shows the measured and estimated states by landmark-based technique.

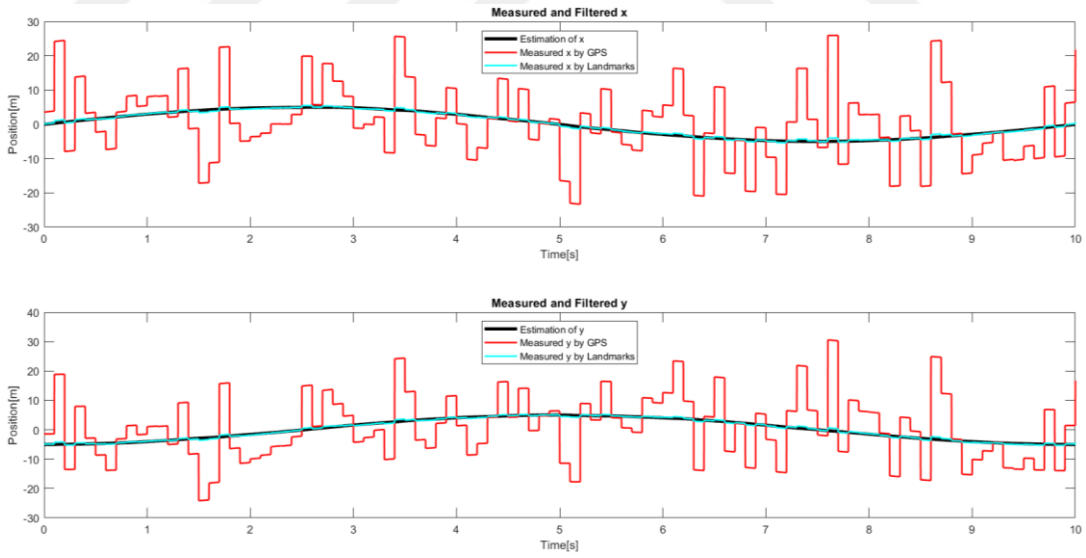

Figure 5.20 Measured and estimated cartesian coordinates of dynamic feedback linearizing controller with landmark-based localization

In Figure 5.20 additional measurements of x and y coordinates are included. Therefore, a better tracking of filter is guaranteed. On the other hand, tracking performance of the kinematic bicycle model can be seen in Figure 5.21.

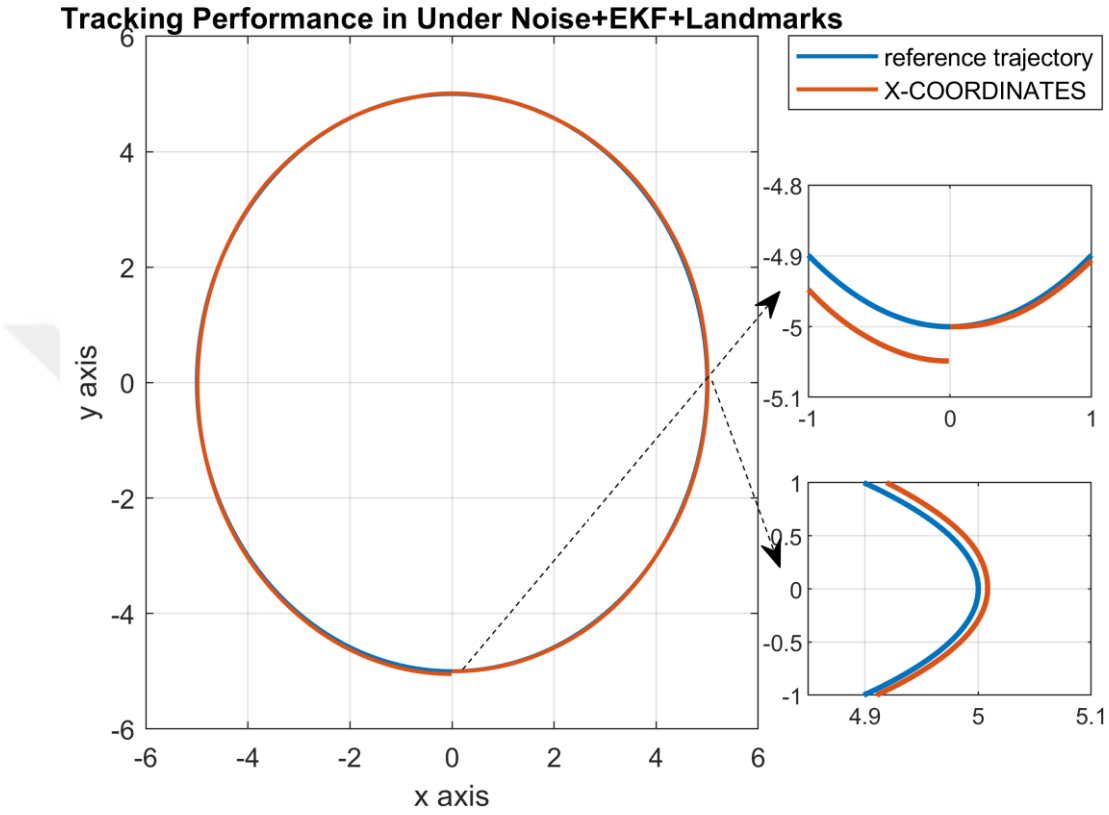

Figure 5.21 Tracking performance of dynamic feedback linearizing controller under noisy measurement and landmark-based localization with Extended Kalman Filter

End point of the trajectory is close to the unstable point of Extended Kalman Filter because of Jacobian-based linearization used for Kalman Filter operations. Therefore, it is expected to be closer to unstable. This situation becomes visible in Figure 5.21.

Next subsection shows simulation results of hybrid control approaches by considering multi-directional cases.

5.7 Hybrid Motion Control of Kinematic Bicycle Model

As shown in previous sections, the tracking performance of controllers differ from each other. While some of them are more insensitive to noisy measurements, some others are almost perfectly tracking when a clean signal is obtained. Another fact is that obstacle avoidance is studied by Pure Pursuit Algorithm and the same algorithm is utilized for approaching the path for tracking purposes. The approaching is done by Pure Pursuit Algorithm because of the assumptions made for LQR and Lyapunov-based controllers. Next subsections show the results of hybrid control strategies by cases.

5.7.1 Case 1: Model Starting Away from Trajectory and Avoiding a Static Point Obstacle

For first case the noise is assumed to be absent. All controller performances are evaluated together with Pure Pursuit Algorithm performing obstacle avoidance. Additionally, another Pure Pursuit Algorithm for approaching the path is utilized for LQR and Lyapunov-based controllers. This consideration is done because of the possibility of the model positioning itself far away from the trajectory. However, dynamic feedback linearizing control is performed without an approaching consideration. Main reason behind this can be pointed out by two points of view. First is to reduce the computational effort of the control action. Second is that convergence of dynamic feedback linearizing control is proven in Figure 5.7 and therefore, this consideration is not necessary. Figure A.8 and A9 show block schemes with two switching mechanisms for Lyapunov-based control and LQR strategies respectively.

As can be seen in Figures A8 and A9, two switching mechanisms utilize relative distances to the path and the obstacle. The switch using the relative distance to the path decides the switching action between approaching the path with Pure Pursuit Algorithm and trajectory tracking controller. If the kinematic bicycle model rear wheel contact point is close to any point of the path it switches to trajectory tracking control.

The threshold of closeness is 0.01 m. Switching is activated by a second condition. It is the orientation error threshold. If the rear wheel reference point is in the closeness threshold and orientation error is inside 0.01 rad region switch activates trajectory tracking controller. This additional condition is guaranteeing a smoother convergence around the path. Approaching Pure Pursuit Algorithm is utilized for three different lookahead distances 1, 2 and 3 m. The second switch is for obstacle avoidance by Pure Pursuit Algorithm. After detecting an obstacle at 2 m relative distance, this block creates a set of waypoints around the obstacle as shown in Figure 5.10. Then activates a Pure Pursuit Algorithm with lookahead distance of 1.5 m. When the obstacle is behind the rear wheel, the algorithm is deactivated. Therefore, it is not possible for the model to get stuck on a circular loop. If this avoidance strategy leads the model away from the path, the first switch would activate the first Pure Pursuit Algorithm to approach. On the feedback arm an additional block is shown. This block calculates the minimum distance point to the path. Moreover, it also provides the time stamp of that point on the trajectory. This is a necessary information for the reference generation block because reference must be expressed as it is starting from that time instant. Initial conditions are expressed in Equation (5.28) and obstacle position is given in Equation (5.29).

$$\begin{bmatrix} x_{initial} \\ y_{initial} \\ \theta_{initial} \\ \varphi_{initial} \end{bmatrix} = \begin{bmatrix} 5.5 \\ 1 \\ \frac{\pi}{2} \\ 0 \end{bmatrix} \tag{5.28}$$

$$\begin{bmatrix} x_{obstacle} \\ y_{obstacle} \end{bmatrix} = \begin{bmatrix} 0 \\ 4.5 \end{bmatrix} \tag{5.29}$$

Same switching mechanism for obstacle avoidance is added to dynamic feedback linearizing control block scheme. Figure A.10 shows the dynamic feedback linearizing control block scheme integrated with obstacle avoidance switching mechanism.

Initialization of 6-dimensional case is given for x and z coordinates in equations (5.30) and (5.31) respectively.

$$\begin{bmatrix} x_{initial} \\ y_{initial} \\ \theta_{initial} \\ \varphi_{initial} \\ v_{initial} \\ a_{initial} \end{bmatrix} = \begin{bmatrix} 5.5 \\ 1 \\ \frac{\pi}{2} \\ 0 \\ \pi \\ 0 \end{bmatrix} \tag{5.30}$$

$$\begin{bmatrix} z_{1initial} \\ z_{2initial} \\ z_{3initial} \\ z_{4initial} \\ z_{5initial} \\ z_{6initial} \end{bmatrix} = \begin{bmatrix} 5.5 \\ 0 \\ 0 \\ 1 \\ \pi \\ 0 \end{bmatrix} \tag{5.31}$$

This block scheme includes a double derivation step after linear and angular speed generation of obstacle avoidance block. This is done because of necessary input transformation for 6-dimensional case. Around the dangerous zone, desired speed of kinematic bicycle model is 1 m/s. Pure Pursuit Algorithm utilized for approaching the path takes π m/s as desired linear velocity. Figure 5.22 shows the performance of all hybrid control strategies.

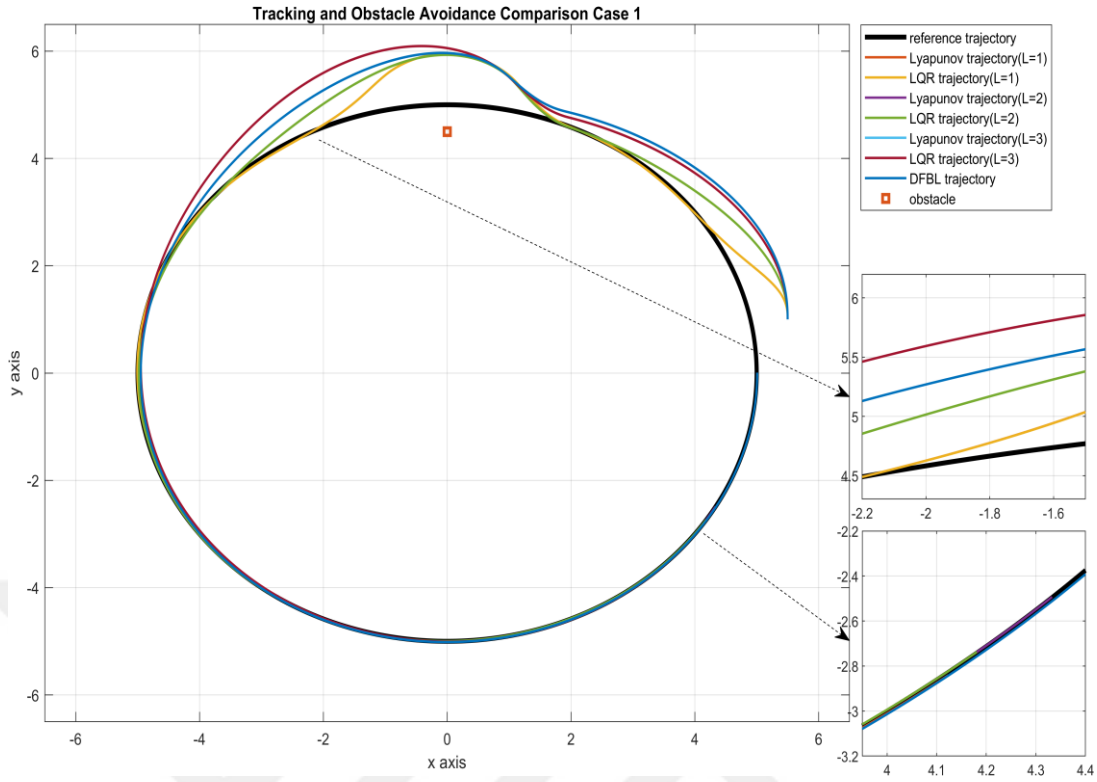

Figure 5.22 Tracking and obstacle avoidance comparison of hybrid controllers

Figure 5.22 shows small difference of performance for all hybrid approaches. Statistical performance metrics are evaluated and presented in Table 5.8.

Table 5.8 Comparison of performance for hybrid controllers

Performance Metric	Cumulative Tracking Error [m]	Mean Tracking Error on x Axis [m]	Mean Tracking Error on y Axis [m]	Variance of Error on x Axis [m^2]	Variance of Error on y Axis [m^2]
LQR +PPA(L=1)	278.4969	0.3796	-0.9469	3.7584	2.9568
Lyapunov +PPA(L=1)	290.5426	0.4606	-0.9536	4.1293	3.1906
LQR +PPA(L=2)	243.1985	0.3213	-0.8235	2.8385	2.2947
Lyapunov +PPA(L=2)	249.0289	0.3654	-0.8134	2.9994	2.4069
LQR +PPA(L=3)	282.6216	0.4359	-1.0246	3.5344	3.2132
Lyapunov +PPA(L=3)	284.1188	0.448	-1.0199	3.5833	3.2479
DFBL	52.0618	0.1453	-0.305	0.0274	0.1295

Table 5.8 shows that the dynamic feedback linearizing control with Pure Pursuit Algorithm-based obstacle avoidance strategy has better performance in every statistical field of this simulation. On the other hand, this strategy is capable to reach the end point of the trajectory at desired time. As a preliminary result, dynamic feedback linearizing control with Pure Pursuit Algorithm-based obstacle avoidance strategy can be said is the best option under clean feedback signals. Another important point to be pointed out is the selection of the best lookahead distance for this case. As previously shown simulations, LQR and Lyapunov-based controller approaches have almost overlapping plots for same selected lookahead distances of approaching Pure Pursuit Algorithm. While smaller lookahead distance selection is increasing convergence speed to the path, it results a more aggressive convergence angle to the

path. Therefore, it is harder for smaller lookahead distance Pure Pursuit Algorithm to satisfy the angular switching condition given above. Larger value selection as 3 m is resulting a longer convergence period and this results a worse tracking outcome. As a result, 2 m selection performed the best tracking and convergence for this scenario. However, lagging results are present for this case because of the selection of the linear speed for avoiding period and approaching period. Dynamic feedback linearizing control approach does not suffer from the lagging result. This is thanks to the possibility of variable speed applicability of the controller itself. Thus, can be pointed out as an advantageous side of this controller. On the other hand, Pure Pursuit Algorithm with designed switching mechanism results a continuation to the trajectory at the converged point.

5.7.2 Case 2: Model Starting Away from Trajectory and Avoiding Moving Point Obstacles

Same initial conditions and operations are performed for all three hybrid controllers. However, this time number of obstacles are increased and they are moving in the workspace. Equations of motions for different obstacles are shown in equations (5.32), (5.33), (5.34), (5.35), (5.36) and (5.37).

$$x_{obstacle_1}(t) = 4 - 0.2\cos\left(\frac{2\pi t}{10}\right) \tag{5.32}$$

$$y_{obstacle_1}(t) = 2 - 0.2\sin\left(\frac{2\pi t}{10}\right) \tag{5.33}$$

$$x_{obstacle_2}(t) = -3 + 0.3\cos\left(\frac{2\pi t}{5}\right) \tag{5.34}$$

$$y_{obstacle_2}(t) = 5 + 0.3\sin\left(\frac{2\pi t}{5}\right) \tag{5.35}$$

$$x_{obstacle_3}(t) = 2.5 + 0.4\sin\left(\frac{2\pi t}{2.5}\right) \tag{5.36}$$

$$y_{obstacle_3}(t) = -2.5 - 0.4\sin\left(\frac{2\pi t}{2.5}\right) \tag{5.37}$$

Distances to the obstacles are measured for all the time instants and Pure Pursuit Algorithm is activated again if model is near the dangerous zone for obstacle avoidance. Figure 5.23 shows the subsystem of multiple obstacle case.

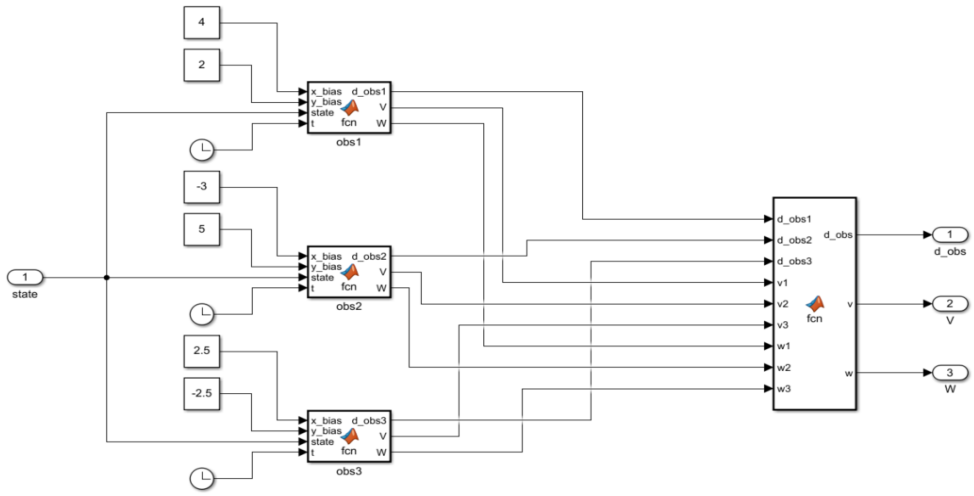

Figure 5.23 Multiple moving obstacle environment decision making subsystem

For this case obstacle avoidance linear speed is set to 1 m/s and approaching linear speed is set to 4 m/s to avoid lagging results. Figure 5.24 shows the tracking and obstacle avoidance performance for the settings shown in case 2. Simulations are again performed for 3 different lookahead distance selection. Simulations are performed in a noise free environment.

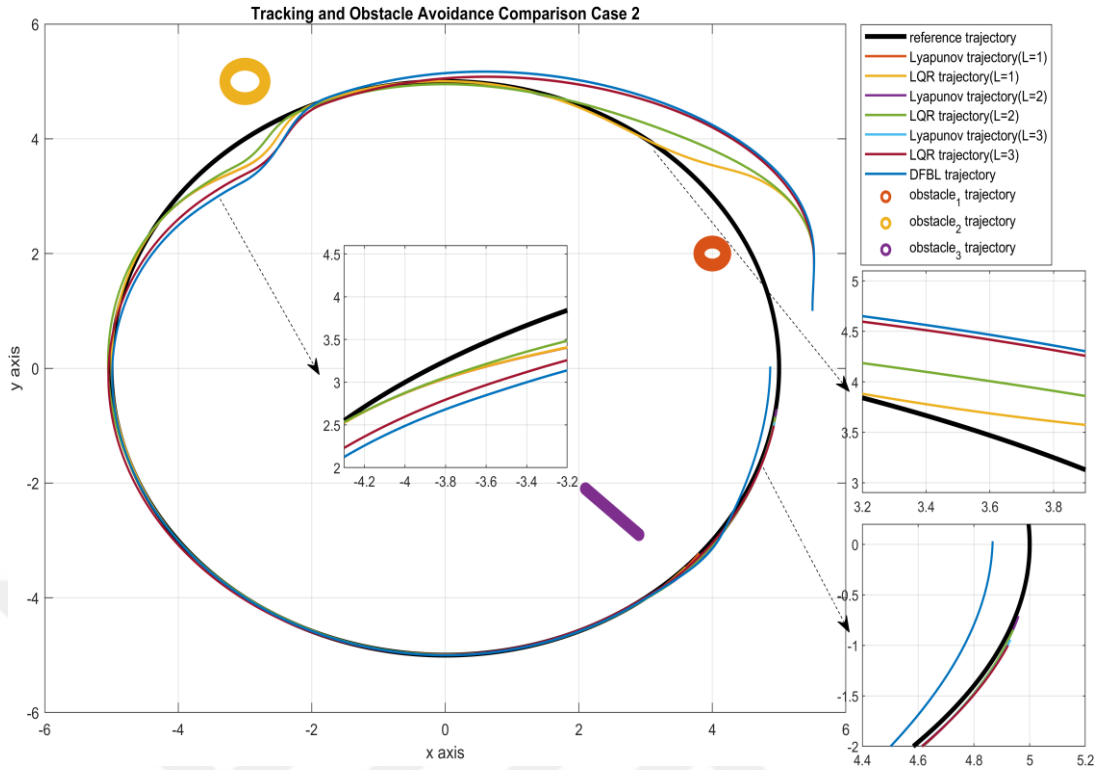

Figure 5.24 Tracking and obstacle avoidance comparison of hybrid controllers in dynamic environment

As can be seen in Figure 5.24, dynamic feedback linearizing controller is having longer distances to converge to the trajectory after obstacle avoidance. However, this approach is again able to finish the trajectory at the end of simulation time. Another fact can be seen around third obstacle. Since dynamic feedback linearizing control technique is able to reach the end in time, it is encountering the third obstacle in a dangerous relative distance. Therefore, it switches to Pure Pursuit Algorithm to avoid this meeting. This behavior results a final error from the trajectory. On the other hand, other two hybrid control techniques are not encountering the same obstacle in a dangerous position. Similar to the previous case, LQR and Lyapunov-based control techniques are having a final error due to lagging and this phenomenon is less in dynamic feedback linearizing control technique. Moreover, convergence profiles of different approaches are evaluated. Like in previous case lookahead distance of 2 m is the best performing selection of parameter. Unlike selection of 3 m and dynamic feedback linearizing control approach, this selection does a faster convergence together with Pure Pursuit Algorithm with 1 m lookahead distance selection. Moreover, 2 m lookahead distance selection with Lyapunov-based technique

approached the most to the end point of the trajectory. Table 5.9 gives the statistical evaluation of this case.

Table 5.9 Comparison of performance for hybrid controllers in dynamic environment

Performance Metric	Cumulative Tracking Error [m]	Mean Tracking Error on x Axis [m]	Mean Tracking Error on y Axis [m]	Variance of Error on x Axis [m^2]	Variance of Error on y Axis [m^2]
LQR +PPA(L=1)	196.3295	-0.0815	-0.4675	1.8196	1.8097
Lyapunov +PPA(L=1)	208.5151	-0.0133	-0.4691	2.094	2.0336
LQR +PPA(L=2)	90.7704	-0.444	-0.2344	0.2795	0.2923
Lyapunov +PPA(L=2)	87.6957	-0.4087	-0.2095	0.2725	0.2856
LQR +PPA(L=3)	119.8571	-0.4852	-0.4711	0.4461	0.5332
Lyapunov +PPA(L=3)	119.0746	-0.4722	-0.4623	0.4463	0.5349
DFBL	45.8053	-0.1498	-0.1354	0.0531	0.1159

Although time spent outside the trajectory is longer compared to the other techniques, dynamic feedback linearizing control technique is again providing most satisfactory results. Like previous case, all statistical performance metrics are better for dynamic feedback linearizing control technique. If two other control approaches are compared, the results are very similar. The results differ for different lookahead

distance selection and for this case 2 m selection performed best for approaching the path task. For this case the lookahead distance for obstacle avoidance is set to 1.5 m like in the previous task.

5.7.3 Case 3: Model Starting on a Linear Trajectory and Avoiding Moving Point Obstacles Under Noisy Measurement

As a final case, a linear vertical trajectory is taken in a noisy measurement environment. Simulation is done for 3 different lookahead distances for approaching phase with Pure Pursuit Algorithm. Obstacle avoidance is done again with Pure Pursuit Algorithm with 1.5 m lookahead distance. The reference trajectory is presented in Equations (5.38) and (5.39).

$$x(t) = 0 \tag{5.38}$$

$$y(t) = -5 + t \tag{5.39}$$

Time derivation of this reference results of a reference linear speed of 1 m/s. In addition to this, other parameters and desired inputs of this scenario is decided by an analysis of previous cases and the objective. This case is considered like a lane changing maneuver with escaping from all obstacles. Therefore, desired linear speed around obstacles are taken as 1.5 m/s and approaching phase has a desired linear speed of 1 m/s. Because it is necessary to reduce oscillating results while approaching the path after avoiding the obstacles. Initial conditions are given in Equations (5.40) and (5.41) for 4-dimensional and 6-dimensional state spaces respectively.

$$\begin{bmatrix} x_{initial} \\ y_{initial} \\ \theta_{initial} \\ \varphi_{initial} \end{bmatrix} = \begin{bmatrix} 0 \\ -5 \\ \frac{\pi}{2} \\ 0 \end{bmatrix} \tag{5.40}$$

$$\begin{bmatrix} x_{initial} \\ y_{initial} \\ \theta_{initial} \\ \varphi_{initial} \\ v_{initial} \\ a_{initial} \end{bmatrix} = \begin{bmatrix} 0 \\ -5 \\ \frac{\pi}{2} \\ 0 \\ 1 \\ 0 \end{bmatrix} \tag{5.41}$$

The initial conditions after coordinate transformation in z-coordinates are shown in equation (5.42).

$$\begin{bmatrix} z_{1initial} \\ z_{2initial} \\ z_{3initial} \\ z_{4initial} \\ z_{5initial} \\ z_{6initial} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -5 \\ 1 \\ 0 \end{bmatrix} \tag{5.42}$$

Noisy measurements are provided by previously shown covariances matrices and utilized Extended Kalman Filters are the same that are provided previously. Movement of the point obstacles are shown in Equations from (5.43) to (5.48).

$$x_{obstacle_1}(t) = -1.5 \tag{5.43}$$

$$y_{obstacle_1}(t) = -2 + 0.1t \tag{5.44}$$

$$x_{obstacle_2}(t) = 1.5 \tag{5.45}$$

$$y_{obstacle_2}(t) = 1 + 0.1t \tag{5.46}$$

$$x_{obstacle_3}(t) = 1 \tag{5.47}$$

$$y_{obstacle_3}(t) = 2 + 0.5t \tag{5.48}$$

Simulation is done for 20 s. Figure 5.25 show the tracking and obstacle avoidance performances of different hybrid control strategies.

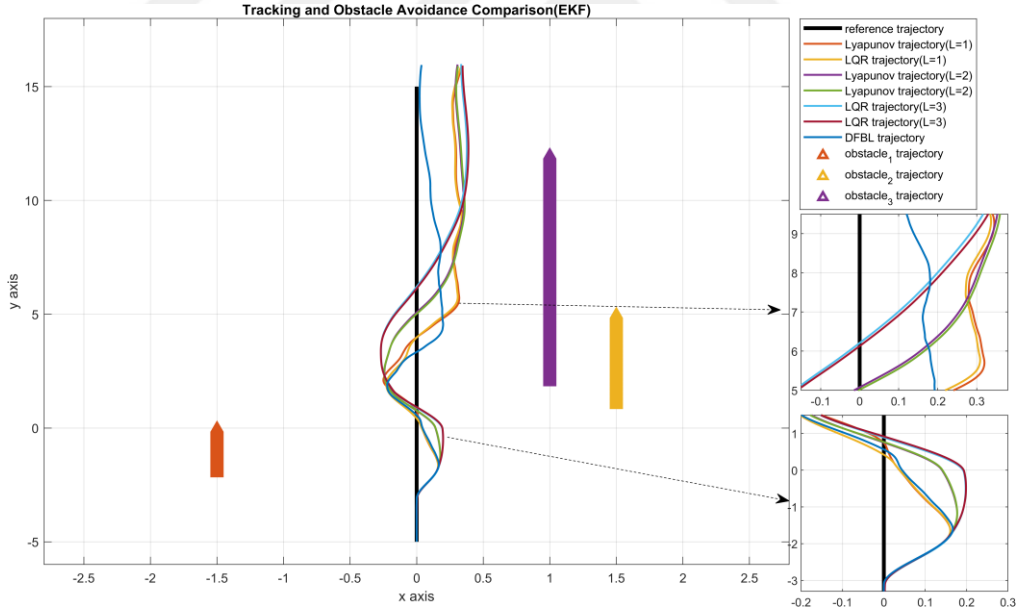

Figure 5.25 Obstacle avoidance and tracking performances comparison under noisy measurement and Extended Kalman Filter

As shown in Figure 5.25, all control approaches start with a good tracking performance in the beginning and then obstacle avoidance is performed. This results a deviation from reference trajectory. Then the convergence and approach to the path starts. However, because of noise effects, the convergence becomes oscillatory and a

larger constant error of position is apparent at the end point of the simulation output for LQR and Lyapunov-based control strategies. Statistical performance metrics and results are shown in Table 5.10.

Table 5.10 Tracking comparison and obstacle avoidance performance of case 3 with EKF

Performance Metric	Cumulative Tracking Error [m]	Mean Tracking Error on x Axis [m]	Mean Tracking Error on y Axis [m]	Variance of Error on x Axis [m^2]	Variance of Error on y Axis [m^2]
LQR +PPA(L=1)	104.1425	-0.1616	-0.6549	0.029	0.0583
Lyapunov +PPA(L=1)	110.491	-0.1579	-0.6909	0.0275	0.0806
LQR +PPA(L=2)	126.8844	-0.1442	-0.805	0.0373	0.0988
Lyapunov +PPA(L=2)	118.559	-0.1471	-0.7402	0.0382	0.095
LQR +PPA(L=3)	130.0776	-0.1213	-0.8238	0.0481	0.1046
Lyapunov +PPA(L=3)	121.5742	-0.1268	-0.757	0.0497	0.1003
DFBL	46.252	-0.0658	0.1088	0.0106	0.0802

This result again shows that dynamic feedback linearizing control approach is more effective on this task under noise and Extended Kalman Filter application. However, the filtering operation is not satisfactory for all techniques and this can be seen in Figures 5.38 and 5.39.

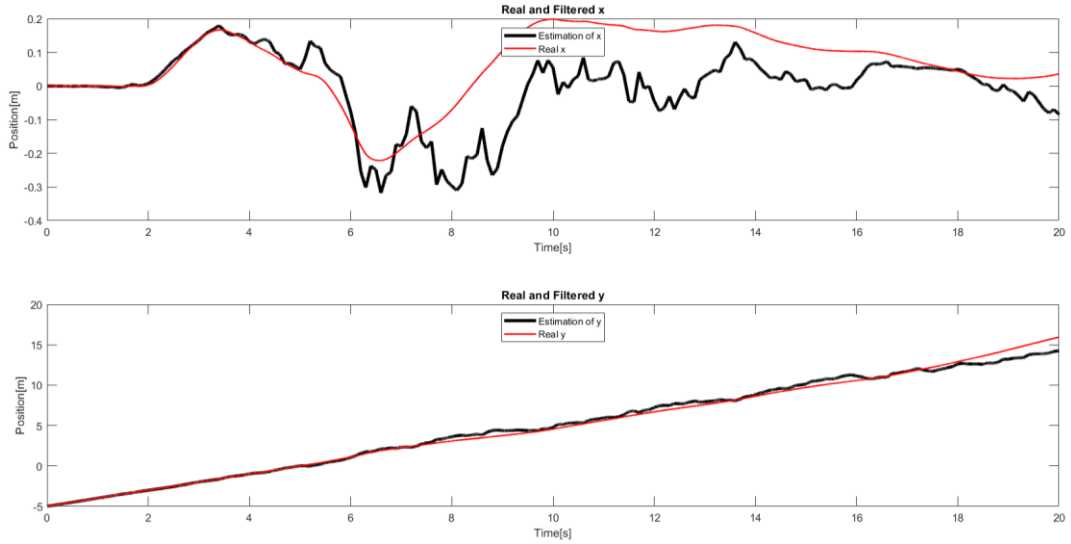

Figure 5.26 Filter performance of cartesian coordinates of dynamic feedback linearizing control approach

In Figure 5.26 cartesian coordinates and their estimations can be seen. This figure is plotted for dynamic feedback linearizing control. Which is the approach that performed the best compared to the others. However, even for this technique, estimation through Extended Kalman Filter has errors. This error over time can be seen in Figure 5.27.

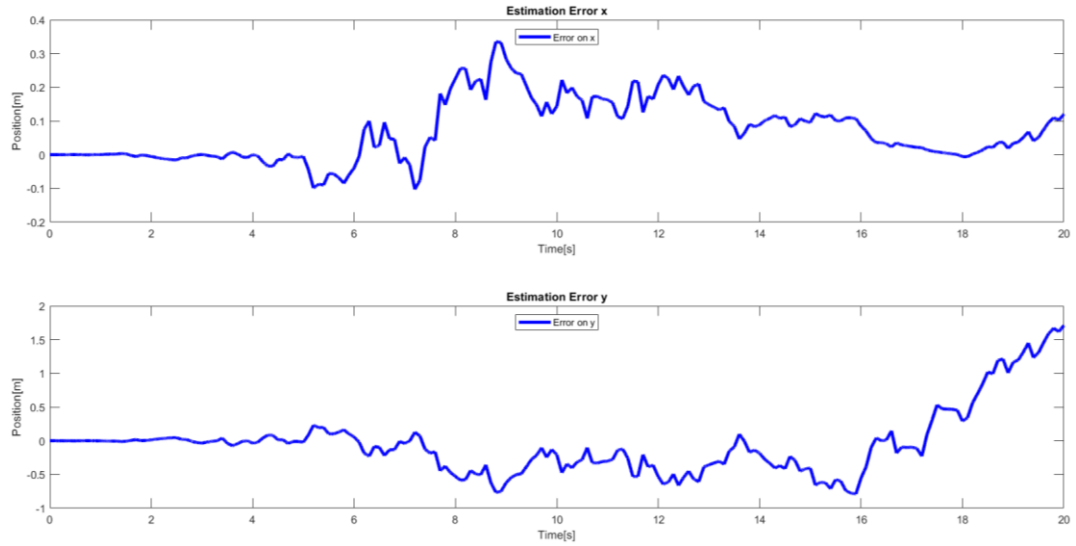

Figure 5.27 Cartesian coordinates estimation error over time for dynamic feedback linearizing control

Estimation error is calculated by difference of real and estimated states. Because of the large random noise effect and avoidance maneuver done by the model, the filter is failing the perfect tracking task and therefore, estimation has larger errors in time.

To overcome this problem, an additional landmark-based measurement is added to the Extended Kalman Filter and sensor fusion is provided for cartesian coordinates. Results can be seen in Figure 5.28.

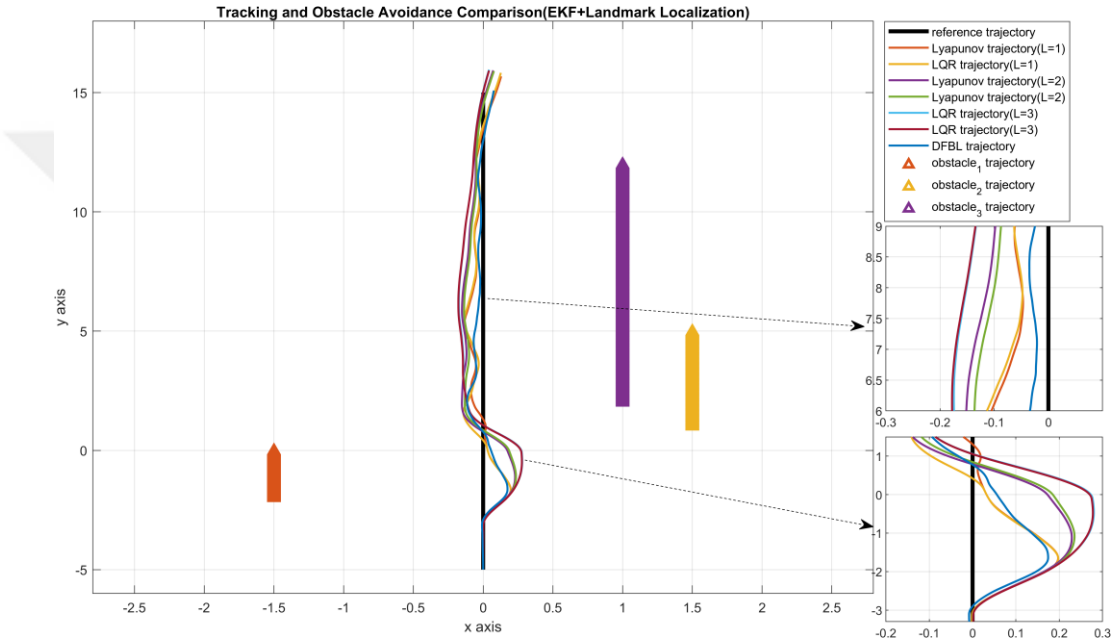

Figure 5.28 Obstacle avoidance and tracking performances comparison under noisy measurement and Extended Kalman Filter with landmark-based localization

Landmark positions are expressed by the Equations (5.49), (5.50), (5.51) and (5.52).

$$\begin{bmatrix} x \\ y \end{bmatrix}_{Landmark_1} = \begin{bmatrix} 5 \\ -3 \end{bmatrix} \tag{5.49}$$

$$\begin{bmatrix} x \\ y \end{bmatrix}_{Landmark_2} = \begin{bmatrix} -6 \\ 0 \end{bmatrix} \tag{5.50}$$

$$\begin{bmatrix} x \\ y \end{bmatrix}_{Landmark_3} = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \tag{5.51}$$

$$\begin{bmatrix} x \\ y \end{bmatrix}_{Landmark_4} = \begin{bmatrix} -3 \\ 10 \end{bmatrix} \tag{5.52}$$

These settings resulted a better tracking and this can be said due to less oscillating signals outside of the trajectory and a smaller final error. More details are given in Table 5.11.

Table 5.11 Tracking comparison and obstacle avoidance performance for case 3 with EKF+landmarks

Performance Metric	Cumulative Tracking Error [m]	Mean Tracking Error on x Axis [m]	Mean Tracking Error on y Axis [m]	Variance of Error on x Axis [m^2]	Variance of Error on y Axis [m^2]
LQR +PPA(L=1)	85.6748	0.0133	-0.563	0.0056	0.0442
Lyapunov +PPA(L=1)	103.8639	0.0187	-0.6732	0.0006	0.0799
LQR +PPA(L=2)	117.031	0.0388	0.7598	0.0115	0.0945
Lyapunov +PPA(L=2)	112.2012	0.0286	-0.7234	0.0107	0.0946
LQR +PPA(L=3)	121.0707	0.0441	-0.783	0.0162	0.1016
Lyapunov +PPA(L=3)	116.1041	0.0452	-0.7446	0.0163	0.1012
DFBL	11.8617	0.0057	0.0029	0.0038	0.0032

Table 5.11 shows that all means and variances of the tracking error signals are reduced drastically. This reduction is more evident on x axis rather than y. For a performance evaluation of the Extended Kalman Filter, similar plots of Figures 5.28 and 5.29 are inspected for this case. Filter tracking performance can be seen in Figure 5.29.

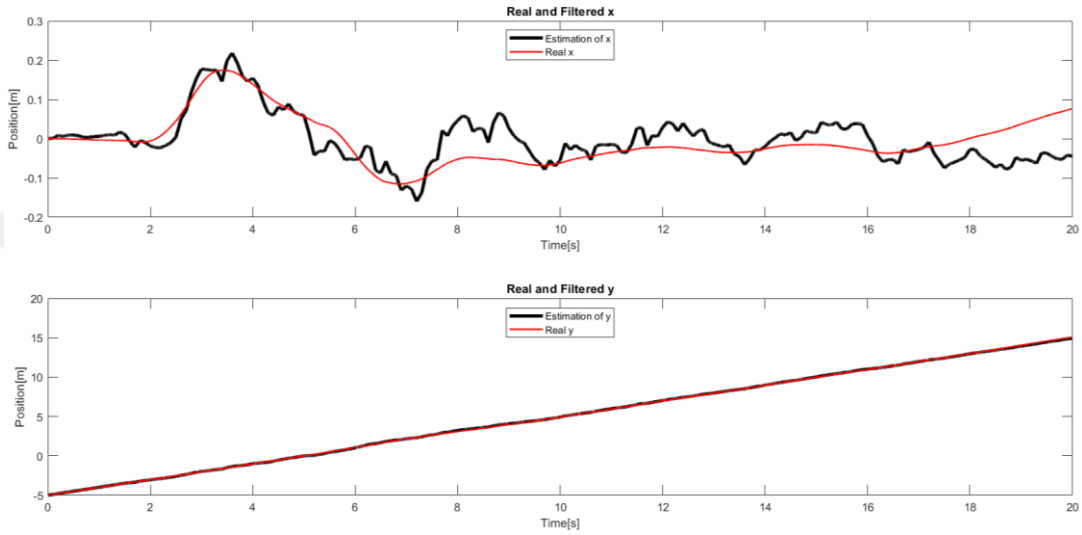

Figure 5.29 Filter performance of cartesian coordinates of dynamic feedback linearizing control approach for EKF+landmarks scenario

This time filter is performing better compared to the only Extended Kalman Filter scenario. A final estimation error is evident but overall performance is better than before. Moreover, estimation error plot is given in Figure 5.30.

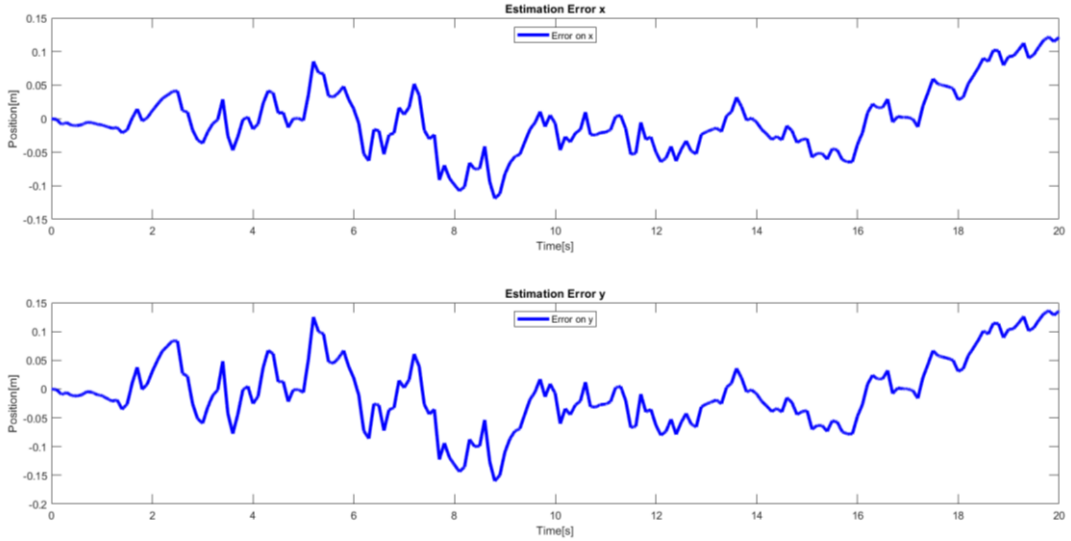

Figure 5.30 Cartesian coordinates estimation error over time for dynamic feedback linearizing control for EKF+landmarks scenario

When Figures 5.29 and 5.32 are compared, the estimation error is visibly reducing for landmark measurement containing scenario. On the other hand, the second best performance is obtained by LQR technique with Pure Pursuit Algorithm approach with 1 m lookahead distance and obstacle avoidance with 1.5 m lookahead distance.

CHAPTER SIX
CONCLUSION

In this study, a hybrid control strategy accomplishing trajectory tracking and obstacle avoidance tasks is investigated. While doing this research, Linear Quadratic Regulator, Lyapunov-based control, Dynamic Feedback Linearization and Pure Pursuit Algorithm are taken into account. As a result, for trajectory tracking purposes Dynamic Feedback Linearization technique performed better than the other techniques. Under the noisy measurements, the Extended Kalman Filter is employed and it is seen that the performance of the control techniques get better by including extra measurements to the filter. Moreover, Pure Pursuit Algorithm performed well for approaching the desired trajectory when the kinematic bicycle model is initialized away from it. Same algorithm worked well on obstacle avoidance task with its geometric approach to the problem.

For future work, can include validation of these approaches by experimenting. Dynamic models can be investigated and different control strategies can be compared to even create a better hybrid approach to the autonomous motion control of the wheeled mobile robots.

APPENDICES

Appendix 1: LQR Control Scheme

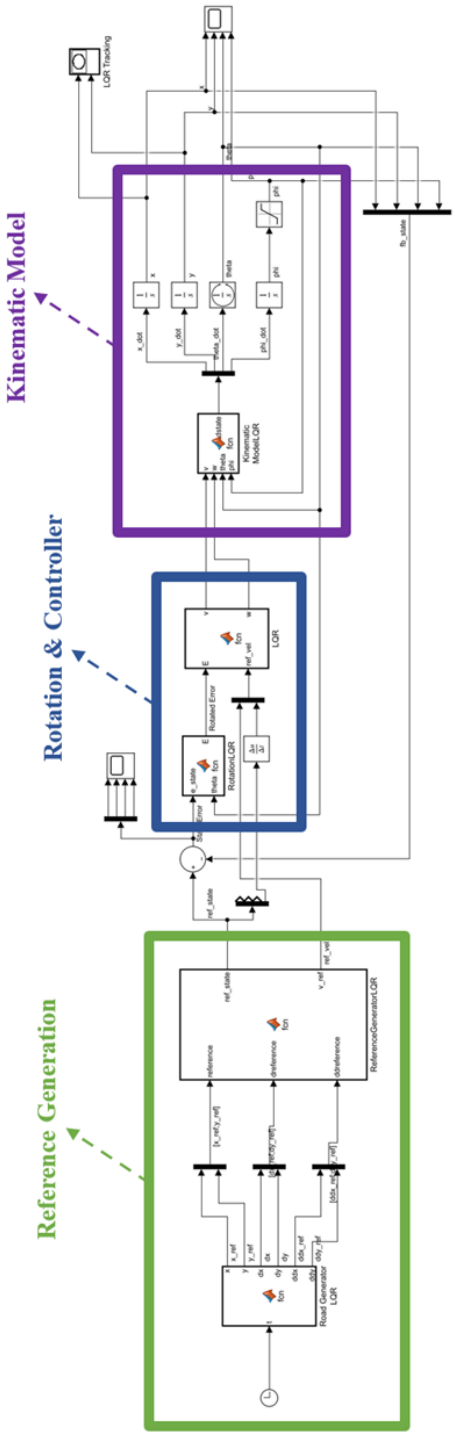

Figure A.1: LQR control scheme Simulink screen

Appendix 2: Lyapunov-Based Control Scheme

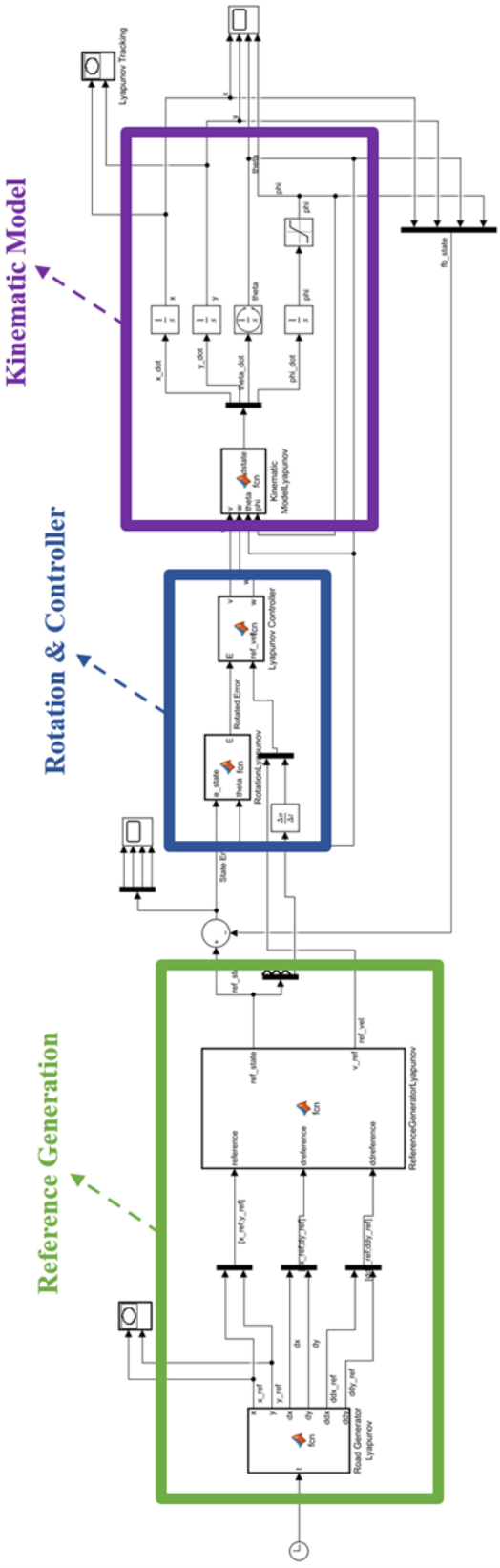

Figure A.2: Lyapunov-based control scheme Simulink screen

Appendix 3: Dynamic Feedback Linearizing Control Scheme

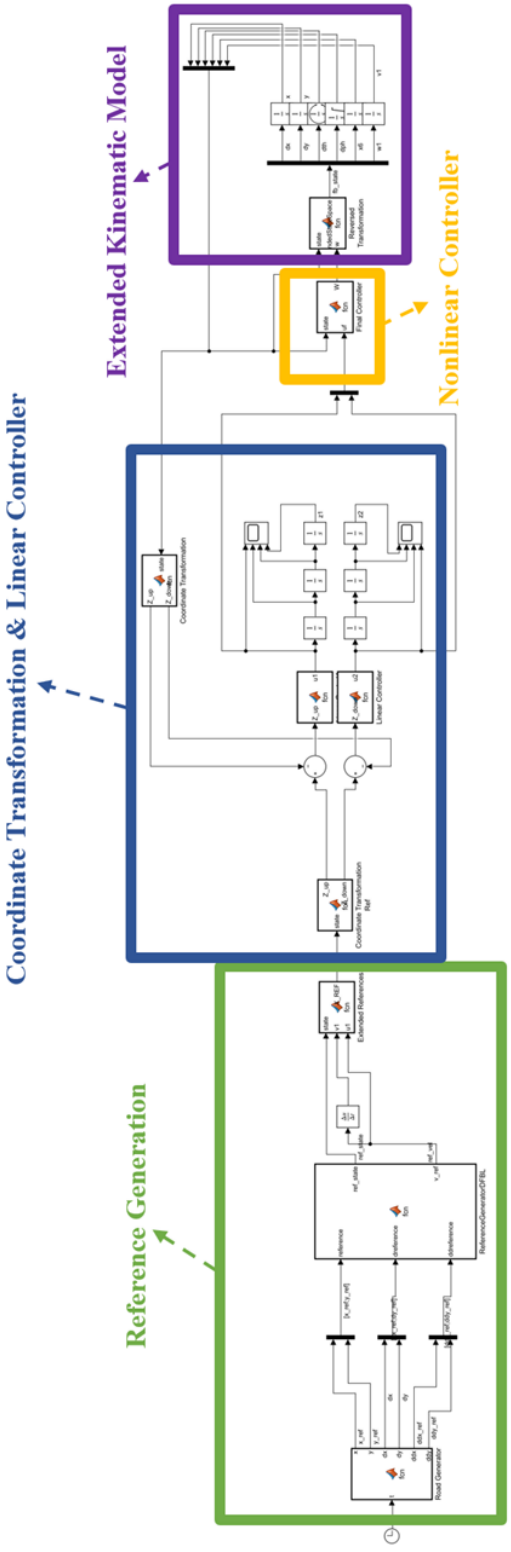

Figure A.3: Dynamic feedback linearizing control scheme Simulink screen

Appendix 4: Pure Pursuit Algorithm Block Scheme

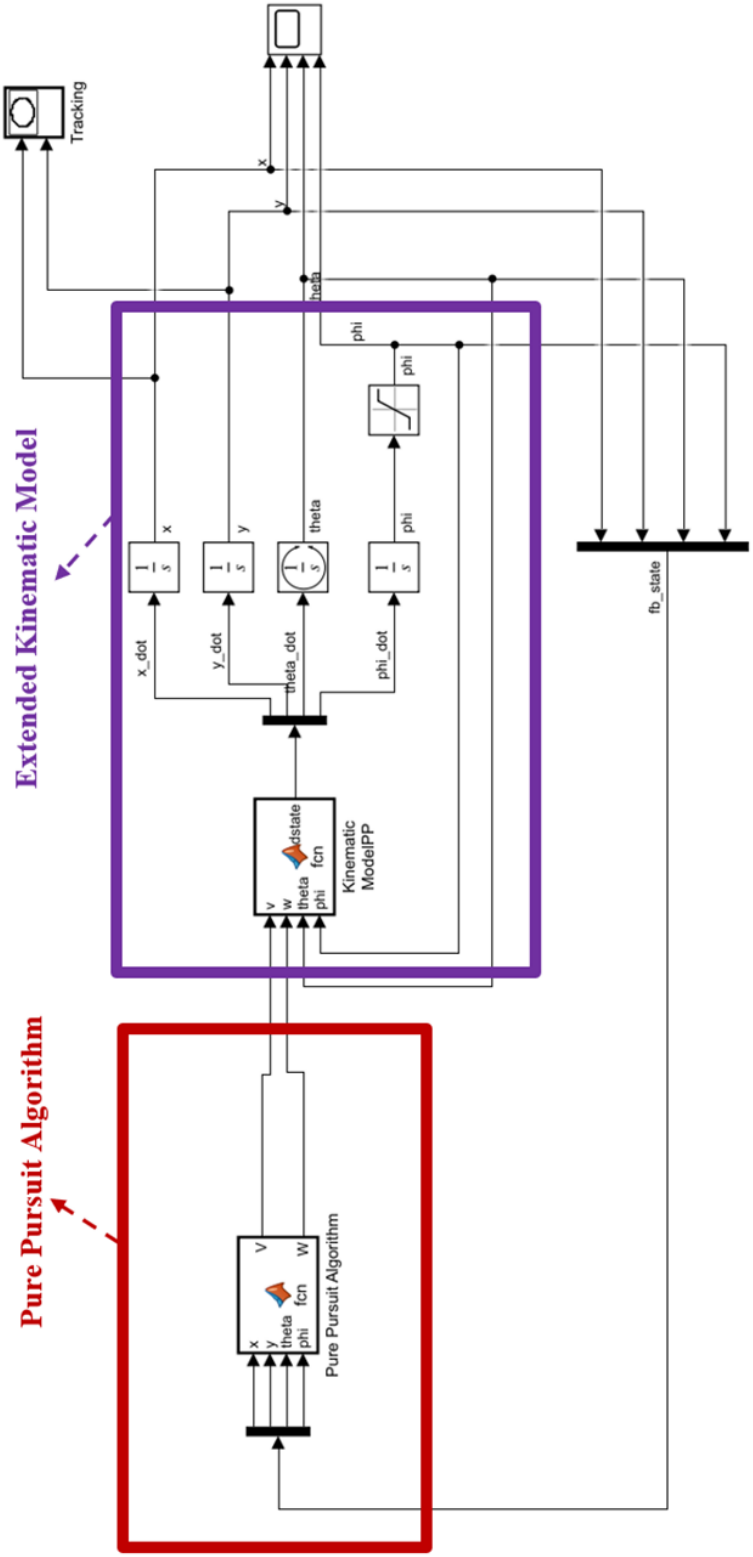

Figure A.4: Pure Pursuit Algorithm Simulink screen

Appendix 5: Lyapunov-Based Technique with Extended Kalman Filter Scheme

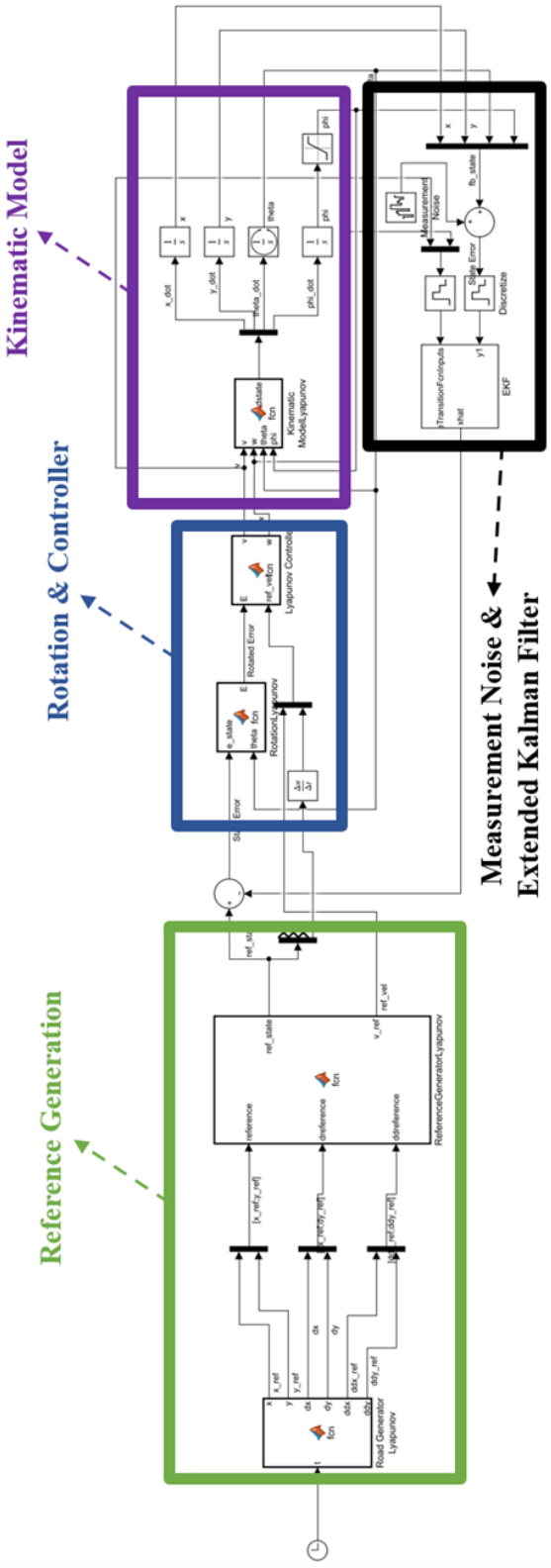

Figure A.5: Lyapunov-Based technique with Extended Kalman Filter Simulink screen

Appendix 6: LQR Technique with Extended Kalman Filter Scheme

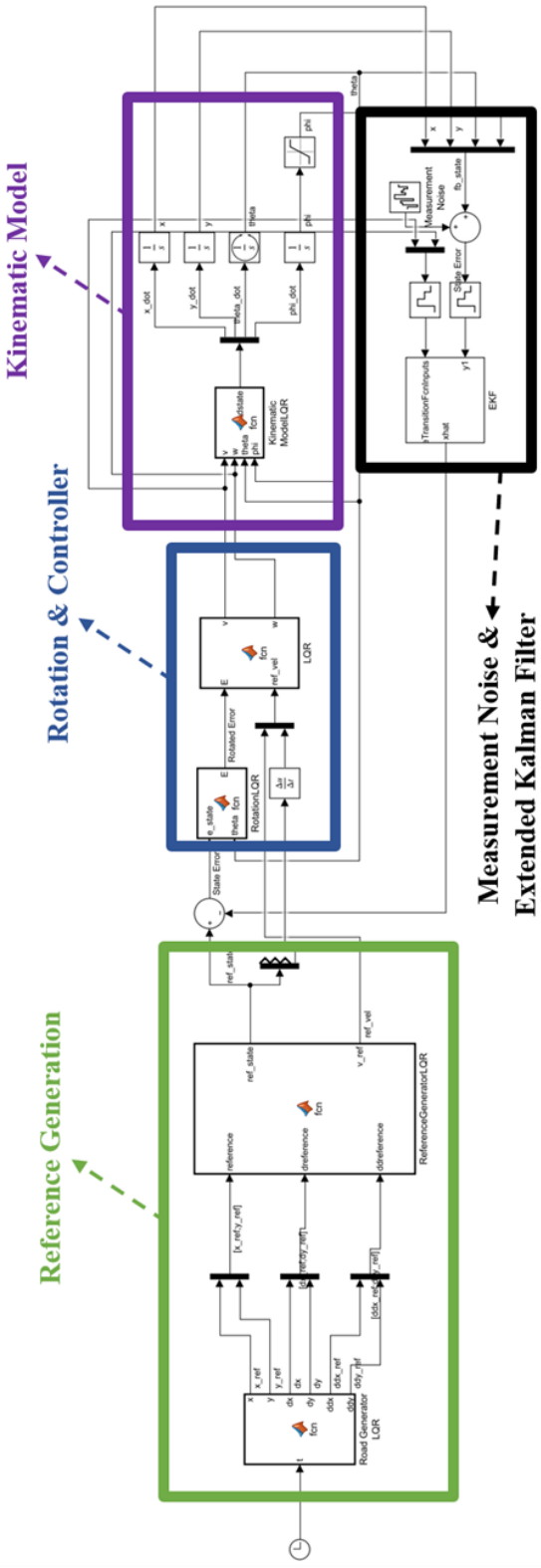

Figure A.6: LQR technique with Extended Kalman Filter Simulink screen

Appendix 7: Dynamic Feedback Linearizing Control with Extended Kalman Filter Scheme

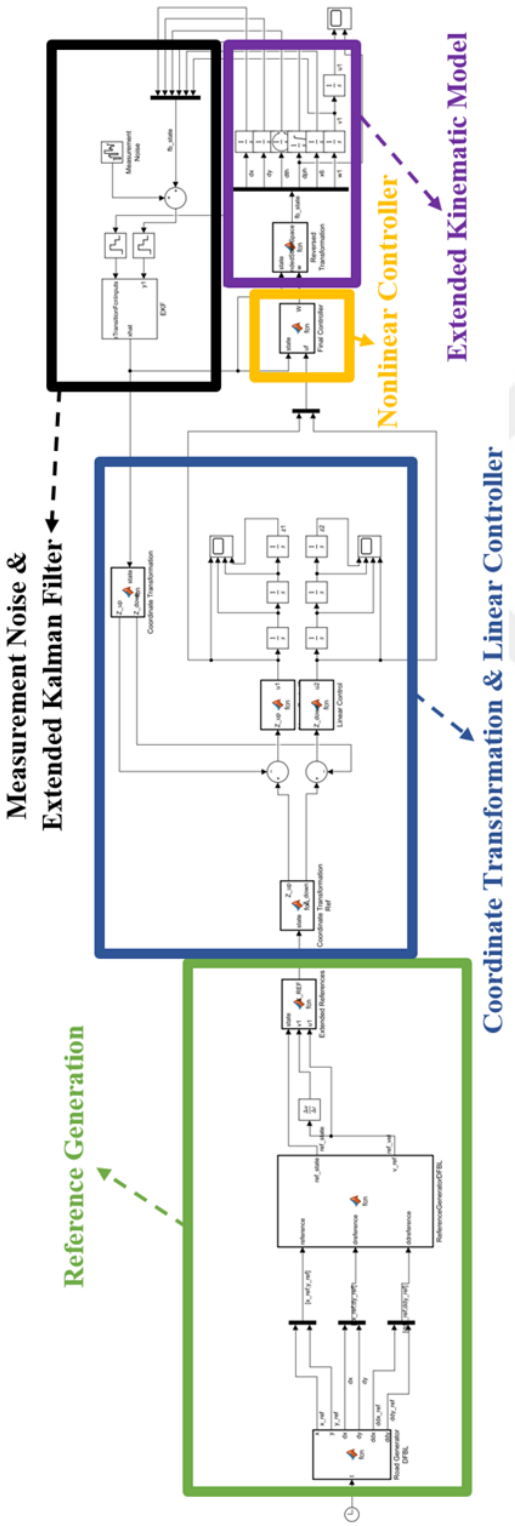

Figure A.7: Dynamic feedback linearizing control with Extended Kalman Filter Simulink screen

Appendix 8: Lyapunov-Based Hybrid Control Scheme

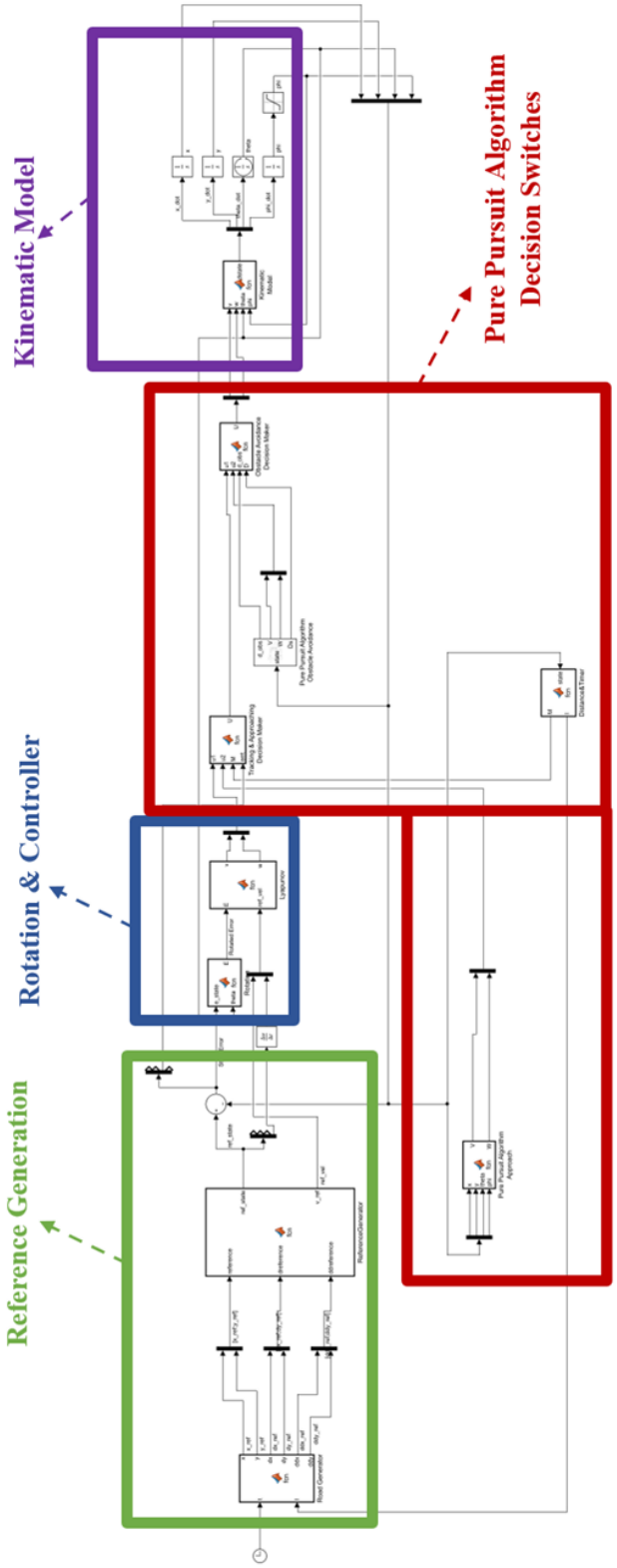

Figure A.8: Lyapunov-based hybrid control with Pure Pursuit Algorithm Simulink screen

Appendix 9: LQR Hybrid Control Scheme

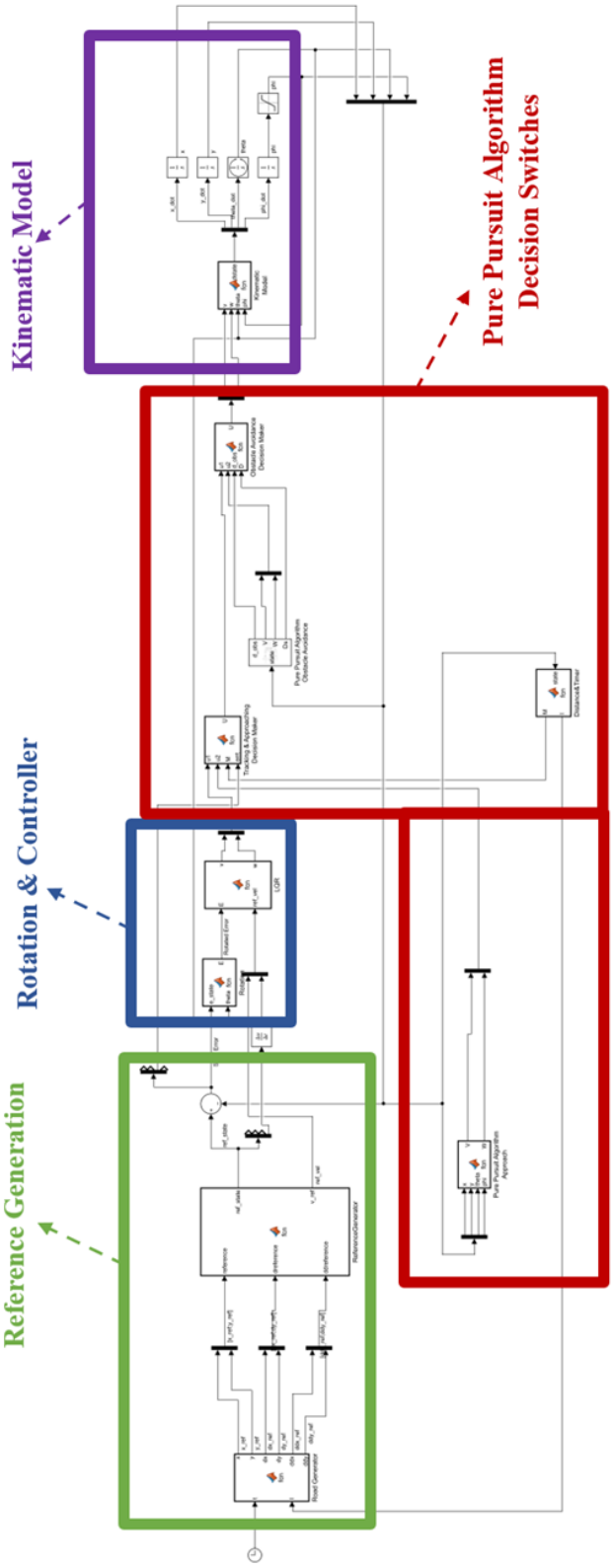

Figure A.9: LQR hybrid control with Pure Pursuit Algorithm Simulink screen

Appendix 10: Dynamic Feedback Linearizing Hybrid Control Scheme

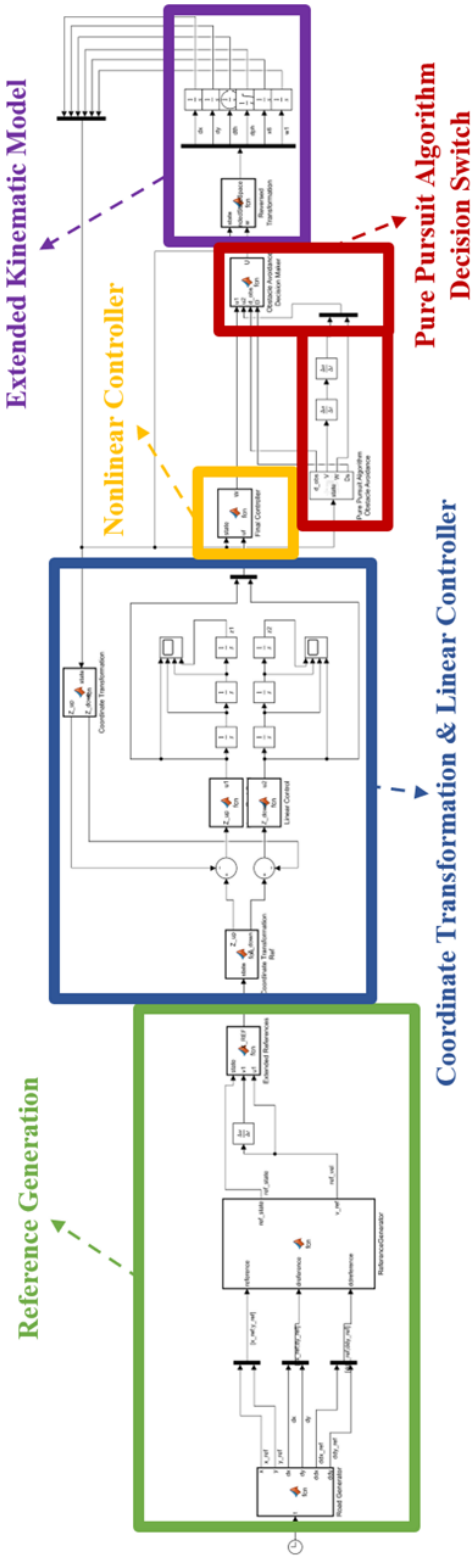

Figure A.10: Dynamic feedback linearizing hybrid control with Pure Pursuit Algorithm Simulink screen

REFERENCES

Alcala, E., Puig, V., Quevedo, J., Escobet, T., & Comasolivas, R. (2018). Autonomous vehicle control using a kinematic Lyapunov-based technique with LQR-LMI tuning. *Control Engineering Practice*, *73*. https://doi.org/10.1016/j.conengprac.2017.12.004

Bronislaw Jakubczyk. (2014). Introduction to Geometric Nonlinear Control; Controllability and Lie Bracket. *Igarss 2014*, *1*.

Bui, T. H., Nguyen, T. T., Chung, T. L., & Kim, S. B. (2003). A simple nonlinear control of a two-wheeled welding mobile robot. *International Journal of Control, Automation and Systems*, *1*(1).

CEYHUN, H. E., & GOREN, A. (2022). Trajectory tracking performance comparison of kinematic bicycle model with LQR and Lyapunov based controllers. *Uluslararası Teknolojik Bilimler Dergisi*. https://doi.org/10.55974/utbd.1130198

Connors, J., & Elkaim, G. H. (2008). Trajectory generation and control methodology for an autonomous ground vehicle. *AIAA Guidance, Navigation and Control Conference and Exhibit*. https://doi.org/10.2514/6.2008-7261

Corke, P. (2017). Robotics, Vision and Control - Fundamental Algorithms In MATLAB®Second, Completely Revised, Extended And Updated Edition. In *Annals of Mathematics and Artificial Intelligence* (Vol. 75, Issues 1–2).

D'andréa-Novel, B., Campion, G., & Bastin, G. (1995). Control of Nonholonomic Wheeled Mobile Robots by State Feedback Linearization. *The International Journal of Robotics Research*, *14*(6). https://doi.org/10.1177/027836499501400602

Divelbiss, A. W., & Wen, J. T. (1997). Trajectory tracking control of a car-trailer system. *IEEE Transactions on Control Systems Technology*, *5*(3). https://doi.org/10.1109/87.572125

Dixon, W. E., Jiang, Z. P., & Dawson, D. M. (2000). Global exponential setpoint control of wheeled mobile robots: A Lyapunov approach. *Automatica*, *36*(11). https://doi.org/10.1016/S0005-1098(00)00099-6

Fareh, R., Saad, M., Khadraoui, S., & Rabie, T. (2018). Lyapunov-Based Tracking Control for Nonholonomic Wheeled Mobile Robot. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, *10*(8), 1042–1047. https://www.researchgate.net/publication/319555601

Forte, M. D. N., Correia, W. B., Nogueira, F. G., & Torrico, B. C. (2018). Reference Tracking of a Nonholonomic Mobile Robot using Sensor Fusion Techniques and Linear Control. *IFAC-PapersOnLine*, *51*(4). https://doi.org/10.1016/j.ifacol.2018.06.092

Freeman, R. A., & Kokotović, P. (1996). Robust Nonlinear Control Design. In *Robust Nonlinear Control Design*. Birkhäuser Boston. https://doi.org/10.1007/978-0-8176-4759-9

Gamkrelidze, R. V. (1978). Principles of Optimal Control Theory. In *Principles of Optimal Control Theory*. https://doi.org/10.1007/978-1-4684-7398-8

Giesbrecht, J., Mackay, D., Collier, J., & Verret, S. (2005). Path tracking for unmanned ground vehicle navigation. In *DRDC Suffield TM* (Vol. 224, Issue December).

Gracia, L., & Tornero, J. (2008). Kinematic control of wheeled mobile robots. *Latin American Applied Research*, *38*(1).

Hoffmann, G. M., Tomlin, C. J., Montemerlo, M., & Thrun, S. (2007). Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. *Proceedings of the American Control Conference*. https://doi.org/10.1109/ACC.2007.4282788

Howie Choset, Kevin Lynch, S. H. (2005). Principles of Robot Motion. In *The Canadian nurse* (Vol. 53).

ISHIKAWA, K., & YOKOYAMA, M. (2018). Control of a wheeled mobile robot via input-output linearization and integral sliding mode control. *The Proceedings of*

the Dynamics & Design Conference, *2018*(0). https://doi.org/10.1299/jsmedmc.2018.728

Jiang, J., & Astolfi, A. (2018). Lateral Control of an Autonomous Vehicle. *IEEE Transactions on Intelligent Vehicles*, *3*(2). https://doi.org/10.1109/TIV.2018.2804173

Jiang, J., Evangelou, S. A., Torquil, M. R., & Astolfi, A. (2017). Robust H∞ Control for Autonomous Scooters. *IFAC-PapersOnLine*, *50*(1). https://doi.org/10.1016/j.ifacol.2017.08.049

Keymasi Khalaji, A., & Jalalnezhad, M. (2021). Robust forward\backward control of wheeled mobile robots. *ISA Transactions*, *115*. https://doi.org/10.1016/j.isatra.2021.01.016

Kirk, D. E. (2004). Optimal control theory: An introduction. *IEEE Transactions on Automatic Control*, *1*.

Kong, J., Pfeiffer, M., Schildbach, G., & Borrelli, F. (2015). Kinematic and dynamic vehicle models for autonomous driving control design. *IEEE Intelligent Vehicles Symposium, Proceedings*, *2015-August*. https://doi.org/10.1109/IVS.2015.7225830

Kozlowski, K., & Pazderski, D. (2004). Modeling and control of a 4-wheel skid-steering mobile robot. *International Journal of Applied Mathematics and Computer Science*, *14*(4).

Laumond, J. P. (1993). Controllability of a Multibody Mobile Robot. *IEEE Transactions on Robotics and Automation*, *9*(6). https://doi.org/10.1109/70.265919

Li, X., Yang, B., Xie, X., Li, D., & Xu, L. (2018). Influence of waveform characteristics on LiDAR ranging accuracy and precision. *Sensors (Switzerland)*, *18*(4). https://doi.org/10.3390/s18041156

Loevsky, I., & Shimshoni, I. (2010). Reliable and efficient landmark-based localization for mobile robots. *Robotics and Autonomous Systems*, *58*(5). https://doi.org/10.1016/j.robot.2010.01.006

Mester, G. (2006). Motion Control of Wheeled Mobile Robots Modeling of the Wheeled Mobile Robots. *4th Serbian-Hungarian Joint Symposium on Intelligent Systems, SISY*.

Michałek, M., & Kozłowski, K. (2011). Unified approach to trajectory tracking and set-point control for a front-axle driven car-like mobile robot. *Proceedings of the American Control Conference*. https://doi.org/10.1109/acc.2011.5990749

Miquel, T. (2020). Introduction to Optimal Control (Doctoral dissertation, ENAC).

Muir, P. F., & Neuman, C. P. (1987). Kinematic modeling of wheeled mobile robots. *Journal of Robotic Systems*, *4*(2). https://doi.org/10.1002/rob.4620040209

Nurmaini, S., Dewi, K., & Tutuko, B. (2016). Differential-drive mobile robot control design based-on linear feedback control law. *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, *3*. https://doi.org/10.1088/1757-899X/190/1/012001

Ohta, H., Akai, N., Takeuchi, E., Kato, S., & Edahiro, M. (2016). Pure pursuit revisited: Field testing of autonomous vehicles in urban areas. *Proceedings - 4th IEEE International Conference on Cyber-Physical Systems, Networks, and Applications, CPSNA 2016*. https://doi.org/10.1109/CPSNA.2016.10

Qu, X., Soheilian, B., & Paparoditis, N. (2018). Landmark based localization in urban environment. *ISPRS Journal of Photogrammetry and Remote Sensing*, *140*. https://doi.org/10.1016/j.isprsjprs.2017.09.010

Renfro, B. A., Stein, M., Reed, E. B., Morales, J., & Villalba, E. J. (2020). An Analysis of Global Positioning System (GPS) Standard Positioning Service Performance for 2019. *Tr-Sgl-20-02*.

Sailan, K., & Kuhnert, Klaus. D. (2013). *Modeling and Design of Cruise Control System with Feedforward for all Terrian Vehicles*. https://doi.org/10.5121/csit.2013.3828

Sert, H., Kökösy, A., & Perruquetti, W. (2011). A single landmark based localization algorithm for non-holonomic mobile robots. *Proceedings - IEEE International Conference on Robotics and Automation*. https://doi.org/10.1109/ICRA.2011.5980002

Siciliano, B., & Khatib, O. (2016). Springer handbook of robotics. In *Springer Handbook of Robotics*. https://doi.org/10.1007/978-3-319-32552-1

Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). Robotics: Modelling, planning and control. In *Advanced Textbooks in Control and Signal Processing* (Issue 9781846286414). https://doi.org/10.5860/choice.46-6226

Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). Introduction to Autonomous Mobile Robots, Second Edition. In *MIT Press* (Vol. 23).

Snider, J. M. (2009). Automatic Steering Methods for Autonomous Automobile Path Tracking. *Work, February*.

Soloperto, R., Wenzelburger, P., Meister, D., Scheuble, D., Breidohr, V. S. M., & Allgöwer, F. (2021). A control framework for autonomous e-scooters. *IFAC-PapersOnLine*, *54*(2). https://doi.org/10.1016/j.ifacol.2021.06.030

Uyar, E., Çetin, L., & Gören, A. (2006). Navigation and GPS based path control of an autonomous vehicle. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3949 LNAI*. https://doi.org/10.1007/11803089_3

Uyar, E., Çetin, L., Gören, A., & Ensoner, S. (2004). Vision based tracking control and obstacle avoidence of a mobile vehicle. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *37*(8).

Vivek, K., Sheta, M. A., & Gumtapure, V. (2019, June). A comparative study of Stanley, LQR and MPC controllers for path tracking application (ADAS/AD). In

2019 IEEE International Conference on Intelligent Systems and Green Technology (ICISGT) (pp. 67-674). IEEE.

Werther Alexandre de Oliveira Serralheiro, Newton Maruyama, & Eduardo Aoun Tannuri. (2015). Motion Control of an Underactuated Wheeled Mobile Robot: a Kinematic Input-Output Linearization Approach. *Proceedings of the 23rd ABCM International Congress of Mechanical Engineering*. https://doi.org/10.20906/cps/cob-2015-1291

Xu, X., Yang, Y., & Pan, S. (2018). Motion Planning for Mobile Robots. In *Advanced Path Planning for Mobile Entities*. https://doi.org/10.5772/intechopen.76895

Zidani, G., Drid, S., Larbi, C. A., Arar, D., & Bussy, P. (2016). Robust nonlinear control of a mobile robot. *Journal of Electrical Engineering and Technology*, *11*(4). https://doi.org/10.5370/JEET.2016.11.4.1014