

T.C.

ISTANBUL OKAN UNIVERSITY

INSTITUTE OF GRADUATE SCIENCES

Master Degree Thesis

Mechanical Engineering

Automotive Mechatronics and Intelligent Vehicles

SIEF ADDEEN YACoub A. SHABANEH

(203005002)

**PURE PURSUIT MODEL WHEN THE VEHICLE
FOLLOWING THE X AND Y AXIS AND STANLEY MODEL**

M.Sc. Thesis by

Thesis Advisor

Prof. Dr. Metin Ergeneman

İSTANBUL, August 2022

T.C.

ISTANBUL OKAN UNIVERSITY

INSTITUTE OF GRADUATE SCIENCES

Master Degree Thesis

Mechanical Engineering

Automotive Mechatronics and Intelligent Vehicles

SIEF ADDEEN YACoub A. SHABANEH

(203005002)

**PURE PURSUIT MODEL WHEN THE VEHICLE
FOLLOWING THE X AND Y AXIS AND STANLEY MODEL**

M.Sc. Thesis by

Presentation Date of Thesis: 31/08/2022

Submission Date of Thesis: 18/05/2022

Advisor: Prof. Dr. Metin Ergeneman _____

Jury Members : Prof. Dr. Ramazan Nejat Tuncay _____

Prof. Dr. Orhan Behiç Alankuş _____

İSTANBUL, August 2022

DEDICATION

This project is dedicated to:

The sake of Allah, my creator and my inspiration

My great lecturers Prof. Dr. Metin Ahmet Ergeneman and all my professors during the Master's study period may Allah bless and grant them all, who have taught me the purpose of life.

To my parents for always loving and supporting me. There wasn't a parenting book in the world that could have prepared you for my eccentricity. Who never stopped supporting me financially and morally, for giving all my needs during the time we developed our system and for teaching me. Even the largest task can be accomplished if it's done one step at a time.

I dedicate this project to all people who have worked to help me complete it.

Dedicated this project to the golden spike in my country and the orange groves... To the vineyards and the olive branch... The blood of the martyrs and the tears of the children.... To (Palestine) My lovely country.

To all my colleagues that we have been running towards success and creativity

To those who carry on their shoulders to build the future generation (our honorable professors)

ACKNOWLEDGEMENTS

I would like to express my gratitude for everyone who helps me during the graduation project, starting with endless thanks for my supervisor Prof. Dr. Metin Ergeneman who keeps effort in encouraging me to do a great job, providing me with valuable information and advise me to do my best..special thanks for the continuous support and kind communication which great effect regarding to feel interesting about what am working on. Finally, thanks to the “Mechanical Engineering Department” for the beneficial lectures provided by the Professors in Okan University.



ABSTRACT

The path following is an important topic for autonomous vehicle and its wide used in robotic fields .The main objective of the project is to study the vehicle's tracking of a point on the x and y axes theoretically and using the MATLAB simulation program, This study is concerned with determining the look-ahead distance using the driving scenario, as well as the curvature of the path by drawing the path and applying pure pursuit controller, and it is determined automatically. And by using the Pure Pursuit Model and in addition to the Stanley Model, where the study was conducted under certain conditions and the vehicle's parameters were changed to see the results and certain points were set for the vehicle to follow where It was found that when the value of the point being tracked decreases, a state of instability occurs under all conditions, in addition to studying the change in the center of gravity and its effect on the stability of the vehicle.

ÖZET

Otonom araç için izlenen yol önemli bir konudur ve robotik alanlarda yaygın olarak kullanılmaktadır. Projenin temel amacı, aracın x ve y eksenleri üzerindeki bir noktayı izlemesinin teorik olarak ve MATLAB simülasyon programı kullanılarak incelenmesidir. sürüş senaryosunu kullanarak ileriye bakma mesafesinin yanı sıra yolu çizerek ve saf takip kontrolörü uygulayarak yolun eğriliğini belirlemekle ilgilidir ve otomatik olarak belirlenir. Ve Pure Pursuit Modeli kullanılarak ve Stanley Modeline ek olarak, çalışmanın belirli koşullar altında yapıldığı ve sonuçları görmek için aracın parametrelerinin değiştirildiği ve aracın takip etmesi için belirli noktalar ayarlandığı tespit edildi. izlenen noktanın değeri azalır, ağırlık merkezindeki değişikliği ve bunun aracın dengesi üzerindeki etkisini incelemenin yanı sıra her koşulda bir kararsızlık durumu oluşur.

TABLE OF CONTENTS

Dedication	i
Acknowledgements	ii
Abstract.....	iii
ÖZET	iv
Table of Contents	v
List of Figurs	viii
List of Tables	xvi
List of Symbols and Abbreviations	xvii
Chapter 1. General Introduction.....	1
1.1. Introduction.....	1
1.2. Motivation.....	1
1.3. Theoretical derivation	1
1.3.1. Lateral Dynamics	3
1.3.2. Earth Axis System.....	3
1.3.3. Vehicle Coordinate System.....	4
1.3.4. Bicycle Model.....	4
1.4. Path representation.....	8
1.5. Pure pursuit algorithm.....	9
Chapter 2. Background and Literature Review	10
2.1. Background.....	10
2.2. Literature Review.....	15
Chapter 3. pure pursuit when the vehicle following the x and y axis	20
3.1. Introduction.....	20
3.2 pure pursuit when the vehicle folloeing the x axis	20

3.3 pure pursuit when the vehicle following the Y axis.....	22
3.4. apply the pure pursuit Model when vehicle following the Y-axis.....	25
3.4.1. For ld is equal 22 m when the vehicle following the Y-Axis	25
3.4.2. For ld is equal 12 m when the vehicle following the Y-Axis	26
3.4.3. For ld is equal 4 m when the vehicle following the Y-Axis	27
3.4.4. Discuss the Results When the ld Has Been Changed (22, 12 and 4) When Vehicle Following the Y- Axis	29
3.5. apply the pure pursuit Model when vehicle following the X-axis.....	30
3.5.1. For ld is equal 22 m when the vehicle following the X-Axis	30
3.5.2. For ld is equal 12 m when the vehicle following the X-Axis	31
3.5.3. For ld is equal 4 m when the vehicle following the X-Axis	32
3.6. Changing the Longitudinal vehicle velocity:.....	34
3.6.1. Changing Longitudinal velocity for pure pursuit model when the vehicle following Y axis	34
3.6.2. Changing Longitudinal velocity for pure pursuit model when the vehicle following X axis	39
3.7. Stanley Method	43
Chapter 4. Change the parameters that was used to see its effect	60
4.1. Study the effect of changing the distance between the front and rear wheels (Wheelbase):..	60
4.1.1. Changing the Wheelbase When the Vehicle Following the Y-Axis (High L)	60
4.1.2. Changing the Wheelbase When the Vehicle Following the Y-Axis (Low L)	65
4.1.3. Changing the Wheelbase When the Vehicle Following the X-Axis (High L)	70
4.1.4. Changing the Wheelbase When the Vehicle Following the X-Axis (Low L)	74
4.2. Study the effect of changing the mass:	78
4.2.1 Bicycle Features:.....	78

4.2.2. When the bicycle following the Y-axis	79
4.2.3. When the bicycle following the X-axis	83
4.2.4. Truck Features:	87
4.2.5. When the truck following the Y-axis.....	87
4.2.6. When the truck following the X-axis.....	92
4.3. Changing the position of center of gravity for pure pursuit.....	96
4.3.1. Making the C.G. Goes forward (near the front wheel) when it's following Y-axis:	96
4.3.2. Making the C.G. Goes backwards (near the rear wheel) For following Y-axis:	97
4.3.3 Making the CG. Goes forward (near the front wheel) For following X-axis	100
4.3.4. Making the CG. Goes backwards (near the rear wheel) For following X-axis:	101
4.4. Changing the center of gravity to the front wheel for Stanley model.....	102
Chapter 5. Results and Conclusion	105
References	110

LIST OF FIGURS

Figure 1.1: Pure Pursuit Geometry	2
Figure 1.2: Simple Kinematic Model.....	2
Figure 1.3: Earth Fixed Axis System.....	3
Figure 1.4: Vehicle Coordinate System	4
Figure 1.5: Kinematics of Lateral Vehicle Motion.....	4
Figure 1.6: Bicycle Model (Dynamic Model).....	6
Figure 1.7: Lateral Slip Angle Calculations	7
Figure 1.8: Lateral Tire Model.....	7
Figure 2.1: Geometry of the Algorithm	10
Figure 2.2: The Step Response of a Second-Order Dynamic System	11
Figure 2.3: The Pure-Pursuit Strategy	12
Figure 2.4: Block Diagram of the Pure-Pursuit Path Tracker.....	13
Figure 2.5: Geometric Explanation of Pure-Pursuit Metho	14
Figure 2.6: Geometric Relationship between Steering Angle and the Vehicle Motion.....	14
Figure 2.7: Path tracking algorithm framework.....	15
Figure 2.8: Pure Tracking Model, Where (x_i, y_i) Are Target Point Coordinates, x, y Are Rear Axle Center Coordinates, R Is Steering Radius.....	16
Figure 2.9 : Path Planning Structural Schematic Diagram	17
Figure 2.10 : Block diagram of the proposed steering control system	18
Figure 3.1: Pure Pursuit Stimulant. When the Vehicle Following the X Axis	20
Figure 3.2: Simple Test When the Vehicle Following a Straight Line in X Axis	21
Figure 3.3: Finding the Coordinates of Rear Axle Center When the Vehicle Following the X-Axis.	21
Figure 3.4: Pure Pursuit Stimulant When the Vehicle Following the Y Axis	22
Figure 3.5: Simple Test When the Vehicle Following a Straight Line in Y Axis	22

Figure 3.6: Finding the Coordinates of Rear Axle Center When the Vehicle Following the Y-Axis.	23
Figure 3.7: The Initial Conditions When the Vehicle Starts From Initial Coordinates (1, 0) and an Initial Angle of 30 Degree with X, Y Axis.....	24
Figure 3.8: The Vehicle Path When (Ld=22 m) When Vehicle Following the Y- Axis	25
Figure 3.9: Stimulant When (Ld=22 m) When Vehicle Following the Y- Axis	25
Figure 3.10: The Vehicle Path When (Ld=12 m) When Vehicle Following the Y- Axis	26
Figure 3.11: Stimulant When (Ld=12 m) When Vehicle Following the Y- Axis	26
Figure 3.12: The vehicle path when (Ld=4 m) when vehicle following the Y- axis.....	27
Figure 3.13: Stimulant When (Ld=4 m) When Vehicle Following the Y- Axis	27
Figure 3.14: The Vehicle Path with 3 Different Looking Ahead Distance (22, 12 And 4 m) When Vehicle Following the Y- Axis.....	28
Figure 3.15: Stimulant When (Ld=22 m) When Vehicle Following the X- Axis	30
Figure 3.16: The Vehicle Path When (Ld=22 m) When Vehicle Following the X- Axis	30
Figure 3.17: Stimulant When (Ld=12 m) When Vehicle Following the X- Axis	31
Figure 3.18: The Vehicle Path When (Ld=12 m) When Vehicle Following the X- Axis	31
Figure 3.19: Simulink When (Ld=4 m) When Vehicle Following the X- Axis	32
Figure 3.20: The Vehicle Path When (Ld=4 m) When Vehicle Following the X- Axis	32
Figure 3.21: The Vehicle Path with 3 Different Looking Ahead Distance (22, 12 and 4 m) When Vehicle Following the X- Axis.....	33
Figure 3.22: Stimulant for the Vehicle That Follows the Y-Axis with 3 Different Values for Ld.....	34
Figure 3.23: The Vehicle Path for 3 Different Look Ahead Distance with (U=3m/s)	35
Figure 3.24: The Vehicle Path for 3 Different Look Ahead Distance with (U=8m/s)	36
Figure 3.25: The Vehicle Path for 3 Different Look Ahead Distance with (U=13m/s)	37
Figure 3.26: The Vehicle Path for 3 Different Look Ahead Distance with (U=25m/s) When the System Crashed.....	38

Figure 3.27: Stimulant for the Vehicle That Follows the X-Axis with 3 Different Values for L_d	39
Figure 3.28: The Vehicle Path for 3 Different Look Ahead Distance with ($U=3\text{m/s}$)	40
Figure 3.29: The Vehicle Path for 3 Different Look Ahead Distance with ($U=8\text{m/s}$)	41
Figure 3.30: The Vehicle Path for 3 Different Look Ahead Distance with ($U=13\text{m/s}$)	42
Figure 3.31: Stanley Geometric Relationship	43
Figure 3.32: Large Heading Error (Course Error)	44
Figure 3.33: Large Heading Error Correction Process	44
Figure 3.34: Large Course Offset, Turn Onto Track First.....	45
Figure 3.35: As The Cross-Track Error Decreases, The Direction Aligns The Route.	45
Figure 3.36: The Large Cross-Track Error Correction Process	45
Figure 3.37: Stanley Controller Stimulant	46
Figure 3.38: Finding the Coordinates of Front Axle Center When the Vehicle Following the X-Axis.....	46
Figure 3.39: Stanley Method When $U = 3 \text{ m/s}$	48
Figure 3.40: Stanley Method When $U = 8 \text{ m/s}$	49
Figure 3.41: Stanley Method When $U = 13 \text{ m/s}$	50
Figure 3.42: Pure Pursuit Controller for an Autonomous Vehicle with Two Different Driving Scenario.....	51
Figure 3.43: Pure Pursuit Controller and the "Wheel Steering Angle Formulation"	51
Figure 3.44: Saif scenario 1 for path tracking with radius $=114.3\text{m}$	54
Figure 3.45: 3D Animation for tracking a path for scenario 1.....	54
Figure 3.46: Vehicle Position By 2D-Visualiza For Saif Scenario 1with Small Look Ahead Distance And Vehicle Steering Angle (δ)	55
Figure 3.47: Vehicle Velocity At The (X, Y) Direction (X,Y) For Saif Scenario 1 For Small Look Ahead Distance	55

Figure 3.48: Pure Pursuit Controller The Results For Small Look Ahead Distance	55
Figure 3.49: Pure Pursuit Controller The Results For Large Look Ahead Distance	56
Figure 3.50: Vehicle Velocity At The (X, Y) Direction (X, Y) For Saif Scenario 1 for large look ahead distance	56
Figure 3.51: Vehicle Position By 2D-Visualiza For Saif Scenario 1with large Look Ahead Distance And Vehicle Steering Angle (δ).....	56
Figure 3.52:Saif Scenario 2 For Path Tracking With Radius =135.6 m.....	57
Figure 3.53: 3D Animation For Tracking A Path For Scenario 2	57
Figure 3.54: Vehicle Position By 2D-Visualiza For Saif Scenario 2 with Small Look Ahead Distance And Vehicle Steering Angle (δ).....	58
Figure 3.55: Vehicle Velocity At The (X, Y) Direction (X, Y) For Saif Scenario 2 For Small Look Ahead Distance	58
Figure 3.56: Pure Pursuit Controller the Results for Small Look Ahead Distance	58
Figure 3.57: Vehicle Velocity At The (X, Y) Direction (X, Y) For Saif Scenario 2 for large look ahead distance	59
Figure 3.58: Vehicle Position By 2D-Visualiza For Saif Scenario 2 with large Look Ahead Distance And Vehicle Steering Angle (δ).....	59
Figure 3.59: Pure Pursuit Controller the Results for Large Look Ahead Distance	59
Figure 4.1: The Vehicle Follows the Y Axis with A Change In the Value of $L=3.4$ m and the ($U=10\text{m/s}$), $L_d= (4, 12, 22 \text{ m})$	61
Figure 4.2: The Vehicle Follows the Y Axis with A Change In the Value of $L=3.4\text{m}$ and the ($U=15\text{m/s}$), $L_d= (4, 12, 22 \text{ m})$	62
Figure 4.3: The Vehicle Follows the Y Axis with A Change In the Value of $L=3.4\text{m}$ and the ($U=19\text{m/s}$), $L_d= (4, 12, 22 \text{ m})$	63

Figure 4.4: The Vehicle Follows the Y Axis with A Change In the Value of $L=3.4\text{m}$ and the ($U=20\text{m/s}$), $L_d = (4, 12, 22\text{m})$	64
Figure 4.5: The Vehicle Follows the Y Axis with A Change In the Value of $L=0.8\text{m}$ and the ($U=10\text{m/s}$), $L_d = (4, 12, 22\text{m})$	65
Figure 4.6: The Vehicle Follows the Y Axis with A Change In the Value of $L=0.8\text{m}$ and the ($U=15\text{m/s}$), $L_d = (4, 12, 22\text{m})$	66
Figure 4.7: The Vehicle Follows the Y Axis with A Change In the Value of $L=0.8\text{m}$ and the ($U=19\text{m/s}$), $L_d = (4, 12, 22\text{ m})$	67
Figure 4.8: The Vehicle Follows the Y Axis with A Change In the Value of $L=0.8\text{m}$ and the ($U=20\text{m/s}$), $L_d = (4, 12, 22\text{m})$	68
Figure 4.9: The Vehicle Follows the X Axis with A Change In the Value of $L=3.4\text{m}$ and the ($U=10\text{m/s}$), $L_d = (4, 12, 22\text{m})$	70
Figure 4.10: The Vehicle Follows the X Axis with A Change In the Value of $L=3.4\text{m}$ and the ($U=15\text{m/s}$), $L_d = (4, 12, 22\text{m})$	71
Figure 4.11: The Vehicle Follows the X Axis with A Change In the Value of $L=3.4\text{m}$ and the ($U=19\text{m/s}$), $L_d = (4, 12, 22\text{ m})$	72
Figure 4.12: The Vehicle Follows the X Axis with A Change In the Value of $L=3.4\text{m}$ and the ($U=20\text{m/s}$), $L_d = (4, 12, 22\text{m})$	73
Figure 4.13: The Vehicle Follows the X Axis with A Change In the Value of $L=0.8\text{m}$ and the ($U=10\text{m/s}$), $L_d = (4, 12, 22\text{m})$	74
Figure 4.14: The Vehicle Follows the X Axis with A Change In the Value of $L=0.8\text{m}$ and the ($U=15\text{m/s}$), $L_d = (4, 12, 22\text{m})$	75
Figure 4.15: The Vehicle Follows the X Axis with A Change In the Value of $L=0.8\text{m}$ and the ($U=19\text{m/s}$), $L_d=(4,12,22\text{m})$	76

Figure 4.16: The Vehicle Follows the X Axis with A Change In the Value of $L=0.8\text{m}$ and the ($U=20\text{m/s}$), $L_d = (4, 12, 22\text{m})$	77
Figure 4.17: Dimensions of the Bicycle That Was Used.....	78
Figure 4.18: The Bicycle ($U=2.78\text{m/s}$) and $L_d = (4, 12, 22\text{m})$ When It's Following the Y-Axis.....	79
Figure 4.19: The Bicycle ($U=5.556\text{m/s}$), $L_d = (4, 12, 22\text{m})$ When It's Following the Y-Axis.....	80
Figure 4.20: The Bicycle ($U=8.334\text{m/s}$), $L_d = (4, 12, 22\text{m})$ When It's Following the Y-Axis.....	81
Figure 4.21: The Bicycle ($U=24\text{m/s}$), $L_d = (4, 12, 22\text{m})$ When It's Following the Y-Axis.....	82
Figure 4.22: The Bicycle ($U=2.78\text{m/s}$) and $L_d = (4, 12, 22\text{m})$ When It's Following the X-Axis.....	83
Figure 4.23: The Bicycle ($U=5.556\text{m/s}$), $L_d = (4, 12, 22\text{m})$ When It's Following the X-Axis.....	84
Figure 4.24: The Bicycle ($U=8.334\text{m/s}$), $L_d = (4, 12, 22\text{m})$ When It's Following the X-Axis.....	85
Figure 4.25: The Bicycle ($U=23\text{m/s}$), $L_d = (4, 12, 22\text{m})$ When It's Following the X-Axis.....	86
Figure 4.26: Dimensions of the truck that was used.....	87
Figure 4.27: The Truck Speed Is 10 Km/h ($U=2.78\text{m/s}$), $L_d = (4, 12, 22\text{m})$ and Following the Y- Axis	88
Figure 4.28: The Truck Speed Is 20 Km/h ($U=5.556\text{m/s}$), $L_d = (4, 12, 22\text{m})$ and Following the Y- Axis	89
Figure 4.29: The Truck Speed is 30 Km/h ($U=8.334\text{m/s}$), $L_d = (4, 12, 22\text{m})$ And Following the Y- Axis	90
Figure 4.30: The Truck Speed Is ($U=11\text{m/s}$) (39.6Km/h), $L_d = (4, 12, 22\text{m})$ and Following the Y- Axis	91
Figure 4.31: The Truck Speed Is 10 Km/h ($U=2.78\text{m/s}$), $L_d = (4, 12, 22\text{m})$ and Following the X- Axis	92
Figure 4.32: The Truck Speed is 20 Km/h ($U=5.556\text{m/s}$), $L_d = (4, 12, 22\text{m})$ And Following the X- Axis	93

Figure 4.33: The Truck Speed is 30 Km/h ($U=8.334\text{m/s}$), $L_d = (4, 12, 22\text{m})$ and Following the X-Axis	94
Figure 4.34: The Truck Speed Is ($U=13\text{m/s}$), (46.8Km/h), $L_d = (4, 12, 22\text{m})$ and Following the X-Axis	95
Figure 4.35: The Vehicle Path for 3 Different Look Ahead Distance with ($U=15\text{m/s}$) and the C.G Goes Forward.....	96
Figure 4.36: The Vehicle Path for 3 Different Look Ahead Distance with ($U=15\text{m/s}$) and the C.G Goes Backward	97
Figure 4.37: The Vehicle Path for 3 Different Look Ahead Distance with ($U=10\text{m/s}$) and the C.G Goes Backward	98
Figure 4.38: The Vehicle Path for 3 Different Look Ahead Distance with ($U=15\text{m/s}$) and the C.G at the Rear Wheel.....	100
Figure 4.39: The Vehicle Path for 3 Different Look Ahead Distance with ($U=15\text{m/s}$) and the C.G Goes Backward	101
Figure 4.40: Stanley Model with Changing the Center of Gravity When ($U=3\text{m/s}$).....	102
Figure 4.41: Stanley Model with Changing the Center of Gravity When ($U=8\text{m/s}$).....	103
Figure 4.42: Stanley Model with Changing the Center of Gravity When ($U=13\text{m/s}$).....	104
Figure 5.1: Small and Large Look Ahead Distance.....	105
Figure 5.2: All Results for Bicycle When It's Following the Y-Axis.....	105
Figure 5.3: All Results for Bicycle When It's Following the X-Axis.....	106
Figure 5.4: All Results for Truck When It's Following the Y-Axis	107
Figure 5.5: All Results for Truck When It's Following the X-Axis	108
Figure 5.6: All Results for Stanly Model with Different Speeds.....	108
Figure 5.7: All the Results for Stanley Method with Changing the Center of Gravity Near to the Front Wheel	109



LIST OF TABLES

Table 1-1: Summary of kinematic model equations	6
--	---



LIST OF SYMBOLS AND ABBREVIATIONS

C.G: Center of Gravity

K: Curvature

a : Distance between front axle and center of gravity (C.G)

b : Distance between rear axle and center of gravity (C.G)

Y_r : Rear tire lateral force

Y_f : Front tire lateral force

δ : Front tire steering angle

V : Lateral velocity at C.G (in vehicle fixed frame)

U : Longitudinal velocity at C.G (in vehicle fixed frame)

r : Vehicle yaw rate (in vehicle fixed frame)

l_d : Looking head distance

L: Wheelbase of the vehicle

l_f : Distance between front axle and center of gravity (C.G)

l_r : Distance between rear axle and center of gravity (C.G)

R: Radius of arc

Ψ : Yaw angle; orientation angle of vehicle with respect to global X axis

B: Vehicle slip angle

X: Global X axis coordinate

Y: Global Y axis coordinate

CHAPTER 1. GENERAL INTRODUCTION

1.1. INTRODUCTION

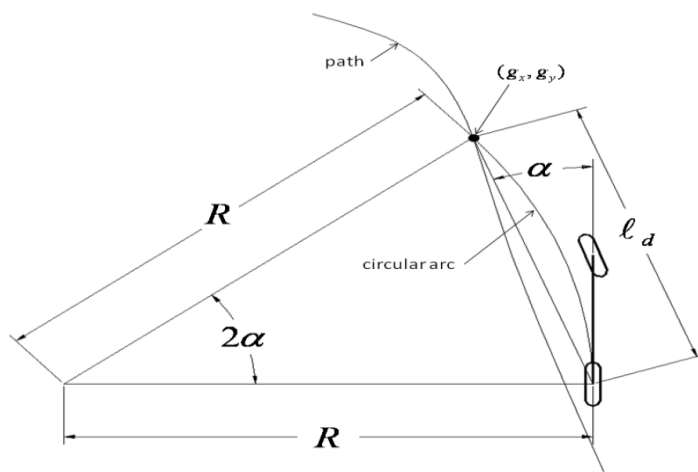
What is pure pursuit? Pure tracking is a tracking algorithm that calculates the curvature of the vehicle moving from its current position to a specific target position.[1] The algorithm's approach is to select a destination that is a certain distance ahead of the vehicle on the route. The name pure pursuit comes from the analogy we used to describe the method. We tend to think that the vehicle is chasing a point some distance ahead of the road (it is chasing that point in motion). This analogy is often used to compare this method to human driving. We tend to look into the distance in front of the car and then walk toward it. As we drive to reflect the road's visual occlusion and distortion, this forward distance changes.

1.2. MOTIVATION

Path tracking is an important fact of autonomous navigation. Nowadays, in many sciences that use robots, the path it follows is of great importance. The importance of making the vehicle follow a certain point is to obtain the best stability and stability of the vehicle during its movement from one place to another, in other words obtaining high efficiency in tracking points at the speed of Different and this topic discussed in the thesis varies with the speed and the point to be tracked, as well as with the difference in the distance between the wheels and the change in the center of gravity.

1.3. THEORETICAL DERIVATION

The pure monitoring approach is a way of geometrically figuring out the curvature, which will drive the vehicle to the selected path point, referred to as the target point. The target point is the point in the direction this is one line of sight from the current vehicle position. An arc connecting the current point and the target point is constructed. The chord length of the arc is the looking head distance and serves as the third constraint situation for figuring out the most straightforward arc connecting the two points. Consider that the ahead distance is much like the distance in the front of the automobile that a human driver might look to track the road. Therefore this approach is likewise referred to as **the dangling carrot controller**.



$$\frac{l_d}{\sin(2\alpha)} = \frac{R}{\sin\left(\frac{\pi}{2} - \alpha\right)}$$

$$\frac{l_d}{2\sin(\alpha)\cos(\alpha)} = \frac{R}{\cos(\alpha)}$$

$$\frac{l_d}{\sin(\alpha)} = 2R$$

$$k = \frac{2\sin(\alpha)}{l_d}$$

Figure 1.1: Pure Pursuit Geometry

Also shown is the point (g_x, g_y) , which is a forward looking distance from the origin. The point (g_x, g_y) must lie on the path. The goal is to calculate the curvature of the arc connecting the origin to (g_x, g_y) and whose chord length is R .

The simple kinematic models was used to get the equations in pure pursuit:

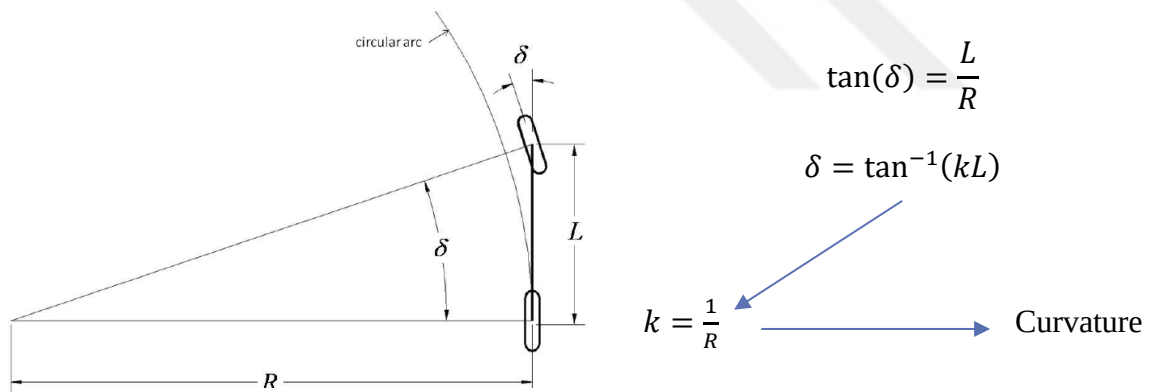


Figure 1.2: Simple Kinematic Model

A common simplification of Ackerman advised motors for geometric direction tracing is the bicycle model. This version simplifies a four-wheeled automobile via way of means of combining the 2 front wheels and the 2 rear wheels collectively to shape a two-wheeled version, just like a bicycle. The second simplification is that the automobile can simplest flow in a plane (no homonymic constraints). These simplifications bring about an easy geometric dating among the guidance perspective of the front wheels (steering angle) and the curvature that the rear axle will comply with as proven withinside the figure above

From equation 1 and 2:

$$\delta(t) = \tan^{-1} \left(\frac{2L \sin(\alpha(t))}{l_d} \right)$$

$$\delta(t) = \tan^{-1} \left(\frac{2L \sin(\alpha)}{kv_x(t)} \right)$$

1.3.1. Lateral Dynamics

Modeling of the lateral dynamics is essential to control the lateral movement of the vehicle. Lateral Motion Control is used for various ADAS functions such as track centering, lane keeping, etc. Lateral dynamics models are also essential for modeling vehicle behavior during lateral maneuvers and can be used for research and design of systems and components.

In order to calculate acceleration and velocity in different directions of interest, it is necessary to define the axis system. In general, the axle system can be fixed to the ground or fixed to the vehicle.

1.3.2. Earth Axis System

Earth's fixed axis system uses X, Y (horizontal), and Z (vertical) as principal directions in global coordinates. The image below shows the vehicle's location and other attributes, such as the heading angle on the ground or the global coordinates.

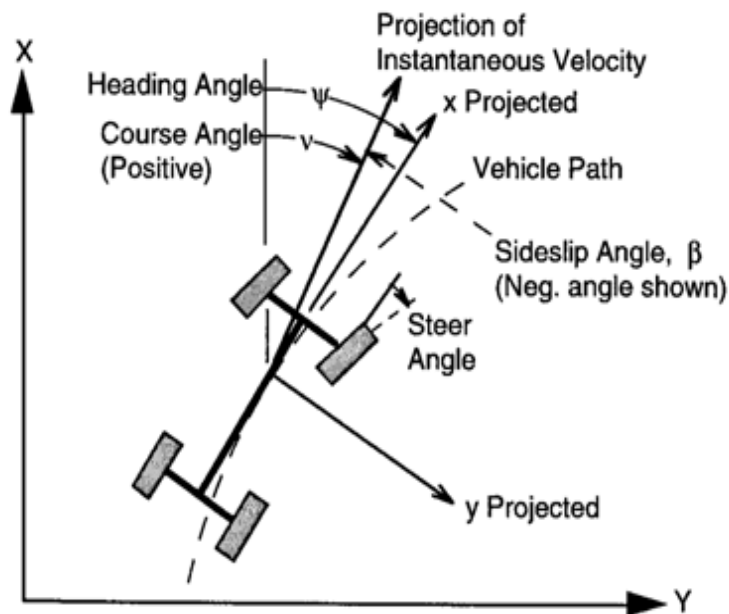


Figure 1.3: Earth Fixed Axis System

1.3.3. Vehicle Coordinate System

This is attached to the vehicle. X is the longitudinal direction, Y is lateral and Z is vertical. The origin is in the center of gravity of the vehicle. The turning movement along the vehicle X is called rolling.

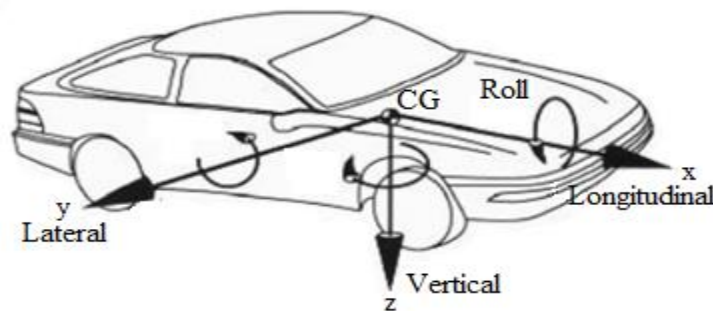


Figure 1.4: Vehicle Coordinate System

1.3.4. Bicycle Model

Many processing vehicle dynamics models of varying complexity and accuracy are available. One of the simplest and most common models is the bicycle model. The term bicycle comes from treating the two front wheels as one unit and the two rear wheels making it a two wheeler model. However, the term “bicycle model” is misleading. It is not used to simulate bike handling dynamics. This is a very poor model for bike dynamics because it doesn't account for roll.

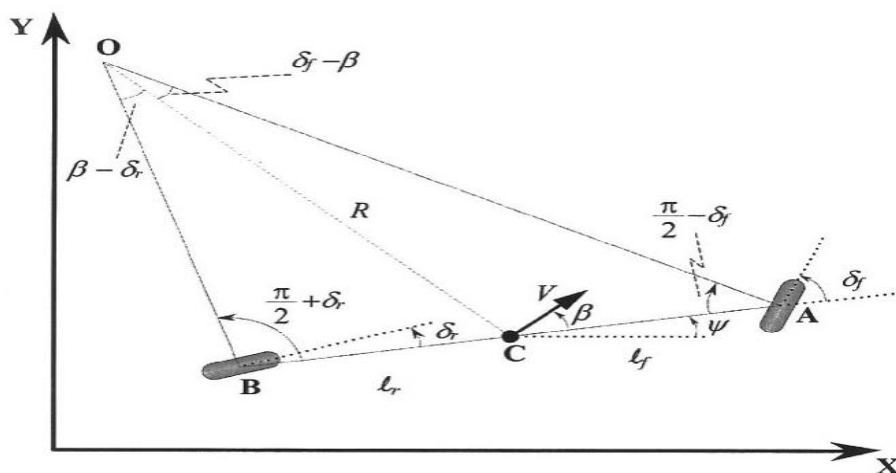


Figure 1.5: Kinematics of Lateral Vehicle Motion

(x, y) is the middle of mass of the car. ψ is the current heading (heading perspective) of the auto, and v is the speed of the vehicle. l_f and l_r are the distances from the middle of mass to the front and rear axis respectively. β is the slip perspective of v relative to the axis of the auto (the sideslip

perspective of the car we take at CG). The guidance perspective of the the front wheels is δ_f , and the acceleration of the auto is a . For the sake of simplicity, we anticipate that it's miles the front-wheel car, and we can now write δ_f as δ .

The main assumption used in developing the kinematic model is that the velocity vectors at point A and point B are in the direction of the front and rear wheels, respectively. In other words, the speed vector of the front wheels forms an angle δ_f with the longitudinal axis of the vehicle. Likewise, the speed vector of the rear wheels encloses an angle δ_r with the longitudinal axis of the vehicle. This is equivalent to assuming that the "slip angle" of the two wheels is zero. This is a reasonable assumption for low speed vehicle movements (e.g. speeds less than 5 m/s). At low speeds, the lateral force generated by the tires is very small.

In order to drive on any circular road of radius R , the total lateral force from both tires is [2]:

$$\frac{mV^2}{R}$$

It changes quadratically with the speed V and is small at low speeds. When the lateral forces are small it is very reasonable to assume that the velocity vector at each wheel is pointing in the direction of the wheel. Point O is the vehicle's instantaneous roll center. Point O is defined by the intersection of lines AO and BO drawn perpendicular to the alignment of the two rollers. The radius of the vehicle path R is defined by the length of the line OC connecting the center of gravity C to the instantaneous roll center O. The velocity in the CG is perpendicular to the line OC. The direction of the speed at the center of gravity in relation to the longitudinal axis of the vehicle is referred to as the vehicle slip angle β . The angle γ is called the course angle of the vehicle. The course angle of the vehicle is $\gamma = \psi + \beta$. Apply the rule of sine to the triangle OCA.

$$\frac{\sin(\delta_f - \beta)}{l_f} = \frac{\sin\left(\frac{\pi}{2} - \delta_f\right)}{R}$$

$$\frac{\sin(\beta - \delta_r)}{l_r} = \frac{\sin\left(\frac{\pi}{2} + \delta_r\right)}{R}$$

$$\frac{\sin(\delta_f) \cos(\beta) - \sin(\beta) \cos(\delta_f)}{l_f} = \frac{\cos(\delta_f)}{R}$$

$$\frac{\cos(\delta_r) \sin(\beta) - \cos(\beta) \sin(\delta_r)}{l_r} = \frac{\cos(\delta_r)}{R}$$

$$\tan(\delta_f) \cos(\beta) - \sin(\beta) = \frac{l_f}{R}$$

$$\sin(\beta) - \tan(\delta_r) \cos(\beta) = \frac{l_r}{R}$$

$$\{\tan(\delta_f) - \tan(\delta_r)\} \cos(\beta) = \frac{l_f + l_r}{R}$$

$$\dot{\Psi} = \frac{V}{R}$$

$$\dot{\Psi} = \frac{V \cos(\beta)}{l_f + l_r} (\tan(\delta_f) - \tan(\delta_r))$$

$$\dot{X} = V \cos(\Psi + \beta)$$

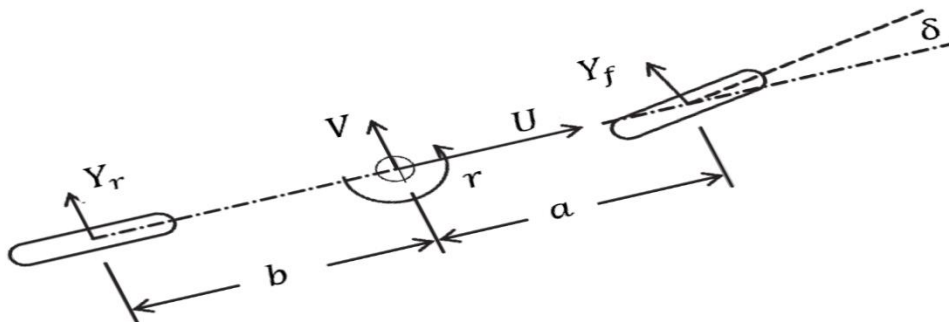
$$\dot{Y} = V \sin(\Psi + \beta)$$

$$\dot{\Psi} = \frac{V \cos(\beta)}{l_f + l_r} (\tan(\delta_f) - \tan(\delta_r))$$

$$\beta = \tan^{-1} \left(\frac{l_f \tan \delta_r + l_r \tan \delta_f}{l_f + l_r} \right)$$

Table 1-1: Summary of kinematic model equations

Symbol	Nomenclature	Equations
X	Global X axis coordinate	$\dot{X} = V \cos(\Psi + \beta)$
Y	Global Y axis coordinate	$\dot{Y} = V \sin(\Psi + \beta)$
ψ	Yaw angle; orientation angle of vehicle with respect to global X axis	$\dot{\psi} = \frac{V \cos(\beta)}{l_f + l_r} (\tan(\delta_f) - \tan(\delta_r))$
β	Vehicle slip angle	$\beta = \tan^{-1} \left(\frac{l_f \tan \delta_r + l_r \tan \delta_f}{l_f + l_r} \right)$



$$m(\dot{V} + rU) = Y_f + Y_r$$

$$I_z \dot{r} = aY_f - bY_r$$

Figure 1.6: Bicycle Model (Dynamic Model)

Now instead of body side slip angles we would assume tire slip angles (angles from tire speed to tire orientation). The analysis would also be done in vehicle coordinates instead of globally. In addition, the longitudinal velocity of the vehicle is u and the lateral velocity is v . The yaw rate (rate of change of angular velocity about the Z-axis) is r and δ_f is the forward steering angle.

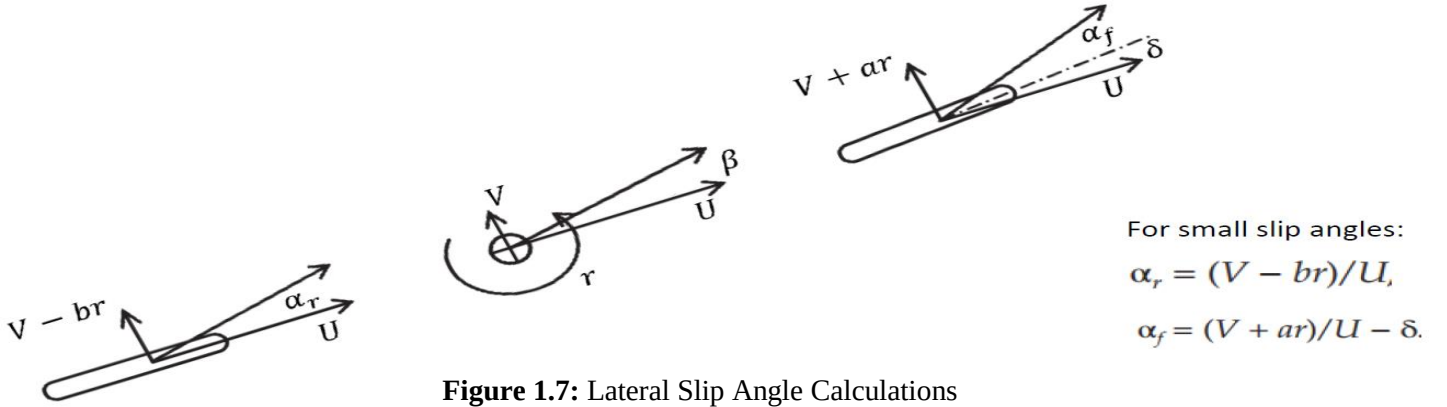


Figure 1.7: Lateral Slip Angle Calculations

Substitute slip angles into tire force equations:

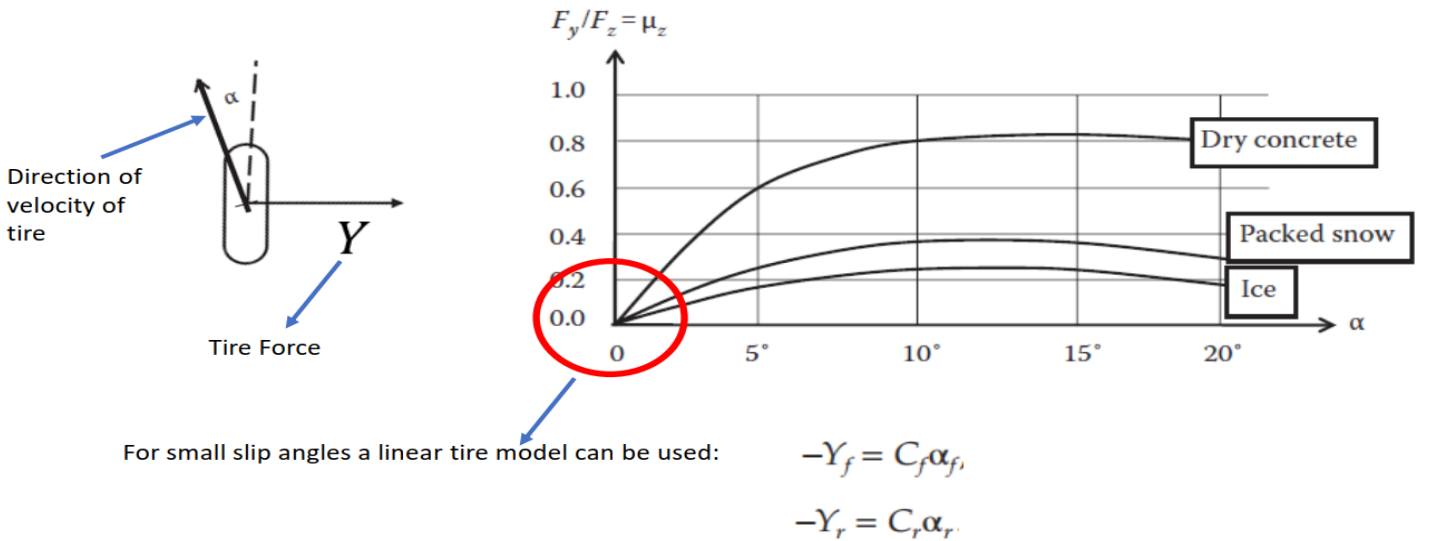


Figure 1.8: Lateral Tire Model

$$-Y_f = C_f \alpha_f \quad \alpha_f = \frac{V + ar}{U - \delta} \quad eq. 1$$

$$-Y_r = C_r \alpha_r \quad \alpha_r = \frac{V - br}{U} \quad eq. 2$$

Substitute tire forces in dynamic equations:

$$m(\dot{V} + rU) = Y_f + Y_r \quad eq. 3$$

$$m(\dot{V} + rU) = -(C_f + C_r)V/U - (aC_f - bC_r)r/U + C_f\delta \quad eq. 4$$

$$I_z \dot{r} = aY_f - bY_r \quad eq. 5$$

$$I_z \dot{r} = -(aC_f - bC_r)V/U - (a^2C_f + b^2C_r)r/U + aC_f\delta \quad eq. 6$$

From equations 4 and 6 and put these equations in state space form:

$$\begin{bmatrix} \dot{V} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -(Cf + Cr)/mU & \left(-\frac{aCf - bCr}{mU}\right) - U \\ -(aCf - bCr)/UIz & -(a^2Cf + b^2Cr)/UIz \end{bmatrix} \begin{bmatrix} V \\ r \end{bmatrix} + \begin{bmatrix} \frac{Cf}{m} \\ \frac{mCf}{Iz} \end{bmatrix} \delta$$

Bicycle Model state-space equations:

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

$$Y = Cx + Du$$

$$A = \begin{bmatrix} -(Cf + Cr)/mU & \left(-\frac{aCf - bCr}{mU}\right) - U \\ -(aCf - bCr)/UIz & -(a^2Cf + b^2Cr)/UIz \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{Cf}{m} \\ \frac{mCf}{Iz} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$D = [0 \quad 0]$$

Although much more complicated than kinematics, the dynamic model is also a simplified model since many assumptions were made, such as linear tires, no roll, small angles, etc. But it is still a powerful tool for processes such as lateral lane keeping controller design, which does not require extensive vehicle modeling. It can also provide a system-level response directed at the available parameters. A full Dynamics multibody model is required for component design and system-level vehicle design. [2]

1.4. PATH REPRESENTATION

The path is represented as a set of discrete points. (Must be saved.) Waypoints are typically specific driving scenarios that contain the following information:

1. X position in coordinates.
2. The position of Y on a coordinates.
3. Coordinates of the head point.
4. The curvature of the road at this time.
5. The distance between this point and the start point of the path.

1.5. PURE PURSUIT ALGORITHM

The implementation of the pure pursuit algorithm itself is quite simple. The pure tracking algorithm can be summarized as follows:

1. Determine the current location of the vehicle.
2. Find the closest waypoint to the vehicle
3. Find the target point.
4. Convert the target point to vehicle coordinates.
5. Calculate the camber and ask the vehicle to target that camber.
6. Update the vehicle position.



CHAPTER 2. BACKGROUND AND LITERATURE

REVIEW

2.1. BACKGROUND

Autonomous vehicles have attracted more and more attention, an interesting key research is stable route tracking for advanced driver assistance systems and with the rapid development of artificial intelligence, autonomous vehicles have received a lot of attention. Related technologies aim to reduce traffic congestion, improve road safety and reduce pollution. One of the most challenging tasks is to develop a stable vehicle control that fulfills the function of automatically following the desired routes at different speeds. To provide insight into state-of-the-art control strategies, four types of common control methods are discussed: Pure Pursuit, Stanley, Linear Quadratic Regulator (LQR) and Linear Model Predictive Control (MPC) with the Ackerman steering model are studied and these methods are tested on different shape trajectories in simulation experiments. LQR and MPC controllers are proven to perform better than Pure Pursuit and Stanley controllers. [3]

In R. Craig Conlter paper [4] the tackled describe in detail the implementation of the pure pursuit path tracking algorithm. Given the algorithm's overall success in recent years, it seems likely that it will be used again for land navigation problems and it also includes the geometric derivation of the procedure and provides some insight into the performance of the algorithm based on its parameters.

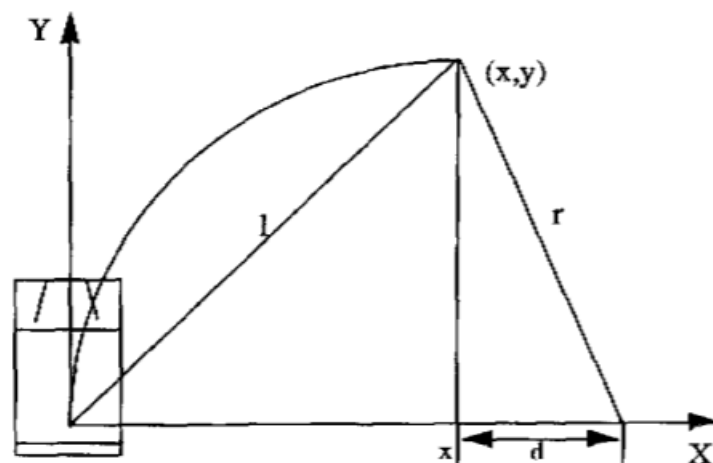


Figure 2.1: Geometry of the Algorithm

$$x^2 + y^2 = l^2 \quad (2.1)$$

$$x + d = r \quad (2.2)$$

Equation 1 describes a circle of radius l around the origin. This is the path of the possible destination point and Equation 2 describes the relationship between the radius of the arc connecting the origin and destination points. Y is the offset x between the target point and the vehicle. This equation simply states that the radius of the arc and the offset x are unrelated and there is a difference times

$$d \cdot d = r - x \quad (2.3)$$

$$(r - x)^2 + y^2 = r^2 \quad (2.4)$$

$$r^2 - 2rx + x^2 + y^2 = r^2 \quad (2.5)$$

$$2rx = l^2 \quad (2.6)$$

$$r = \frac{l^2}{2x} \quad (2.7)$$

$$\gamma = \frac{2x}{l^2} \quad (2.8)$$

The theoretical guidance discovered by the vehicle is plotted and the axis of the vehicle's coordinate system is drawn. The X axis runs through the rear axle of the vehicle. When the vehicle coordinate system is at the rear differential, as shown in the figure below, and the x -axis is collinear with the rear axle, the drive and steering are geometrically separated. [4]

Longer look-ahead distances tend to converge on the route more gradually with less variation. The response of a pure tracking tracker looks like the step response of a second-order dynamic system, shown in the figure below. And the value of l tends to act as a damping factor.

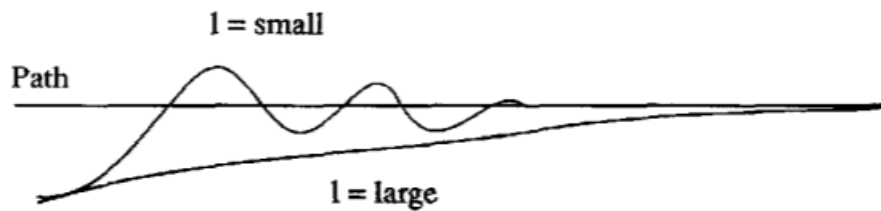


Figure 2.2: The Step Response of a Second-Order Dynamic System

The second problem is that the greater the distance ahead, the smaller the "curvature" in the path that can be followed. The algorithm calculates the curvature to allow the vehicle to make the curve. If the path between the vehicle and the destination point is "curvy" enough, then there is no single arc connecting those two points; each driving arc causes an error. Some limitations of this route tracking method are summarized. Two main limitations relate to kinetic effects.

This method does not simulate the capabilities of the vehicle or its actuators and therefore assumes a perfect response to the requested curvature. This leads to two problems:[5]

- 1) The camber may need to change sharply at high speeds, causing the rear of the vehicle to
- 2) The vehicle does not approach the path as quickly as expected due to the first-order rear-end sideslip.

In Jesus Morales, Jorge L. Mart'inez, Mar'ia A. Mart'inez, and Anthony Mandow work [7] the investigated focused on the problem of The pure search trajectory tracking method has been widely used for planned navigation of non-holonomic ground vehicles in order to tackle this issue To study the application of this technique for reactive tracking of routes implicitly defined by perceived environmental features. Target points are obtained by efficiently interpreting range data from an integrated 2D laser scanner to follow people, corridors and walls. In addition, this formulation allows a robotic mission to be composed of a combination of different types of track sections.

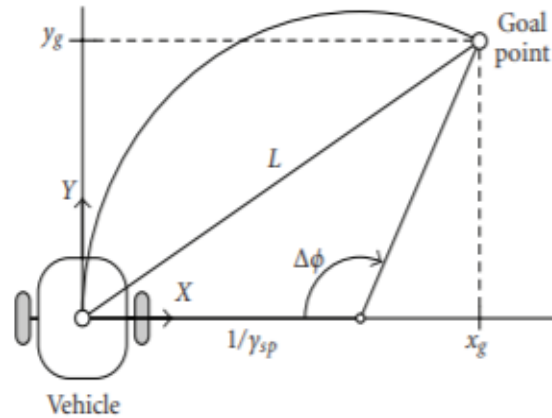


Figure 2.3: The Pure-Pursuit Strategy

In this article, pure tracking is a widely used technique and has been discussed by many authors. However, since this method was first proposed for mobile robots, implicit path tracking has received

little attention beyond our early efforts to track humans with rotating sonar and 2D laser scanners. Furthermore, previous work has not explored the possibility of combining different types of implicit.

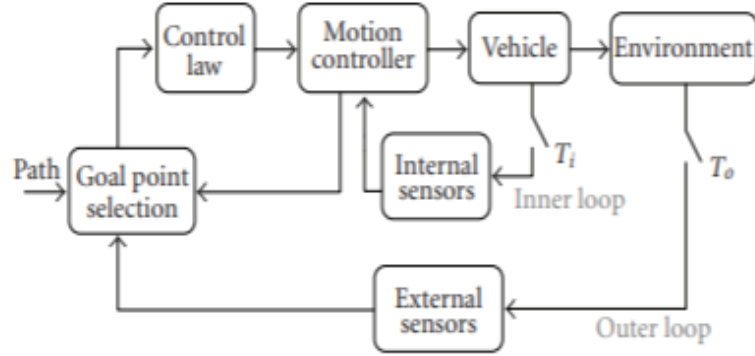


Figure 2.4: Block Diagram of the Pure-Pursuit Path Tracker

Nothing was mentioned about Y-axis vehicle tracking, but in this investigation there are also some other things that can be done, but at the same time Y-axis tracking is missing.

A master's thesis was read to solve a problem in hybrid cars while finding a way. This research was adopted by Istanbul Technical University, and the commonalities between the research and the dissertation were adopted. Master thesis name is “Hybrid Controller Approach for an Autonomous Ground Vehicle Route Tracking Problem” and student “Mertcan Cibooğlu” and this work was done in (2016).

Pure pursuit is one of the most commonly used methods for routing problems because of its simplicity and good performance. The main idea behind the pure tracking method is to always “look ahead” to the target point on the trajectory and find the steering angle that allows the vehicle to pass the target point. Hence the whole path tracing problem becomes a simple geometric problem. [6]

These equations were used:

$$\frac{l_d}{\sin(2\alpha)} = \frac{R}{\sin\left(\frac{\pi}{2} - \alpha\right)} \quad (2.9)$$

$$\frac{l_d}{2\sin(\alpha)\cos(\alpha)} = \frac{R}{\cos(\alpha)} \quad (2.10)$$

$$\frac{l_d}{2\sin(\alpha)} = R \quad (2.11)$$

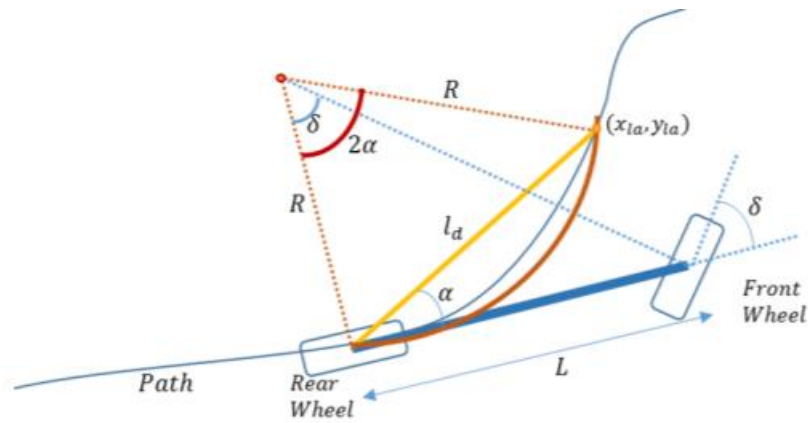


Figure 2.5: Geometric Explanation of Pure-Pursuit Method

And the geometric relationship between steering angle and the vehicle motion

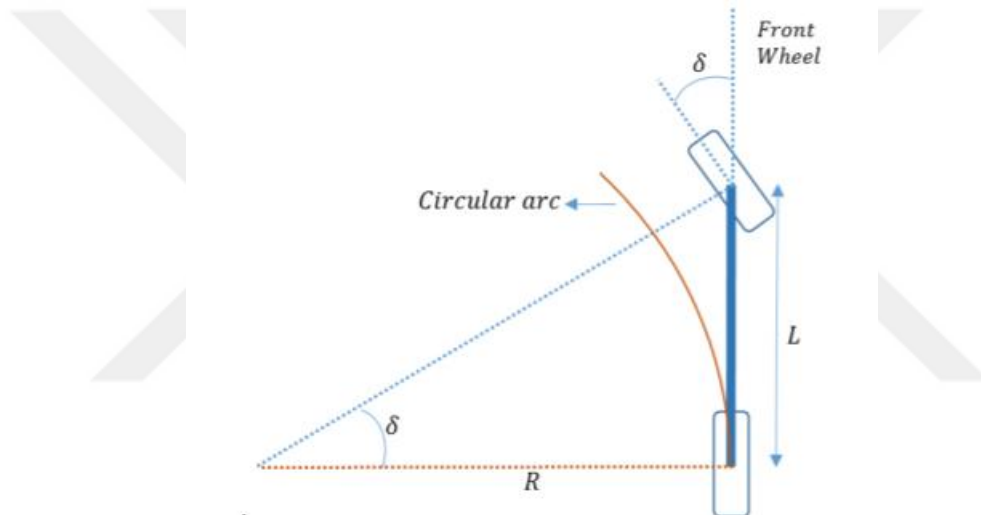


Figure 2.6: Geometric Relationship between Steering Angle and the Vehicle Motion

2.2. LITERATURE REVIEW

In Yang Yang, Yankai Li , Xing Wen , Gang Zhang , Qianglong Ma , Shangkun Cheng , Jian Qi , Liangyuan Xu and Liqing Chen research [8] the tackled is Pure pursuit algorithm as a popular automatic navigation technology control algorithm for agricultural machinery, the look-ahead distances the key to the consequential effect, while the look-ahead distance calculation has the problem that many influencing factors are difficult to determine accurately. Described by mathematical expressions, leading to the difficulty of selecting an appropriate waypoint tracing path.in order to tackle this problem they proposed a route tracking algorithm for agricultural machinery based on the optimal destination point. The algorithm simulates the look-ahead behavior of the driver and according to the evaluation function, searches for the optimal target point in the anticipation area. The aim of the study is to minimize the side error and the head error to achieve the adaptive optimization of the aiming point. Their approach were based on Tractor test platform, Mathematical model of agricultural machinery, Pure Pursuit algorithm, Path tracking stability analysis ,Path tracking algorithm based on optimal goal point . For determine the look-ahead area (Based on Python 3.7).

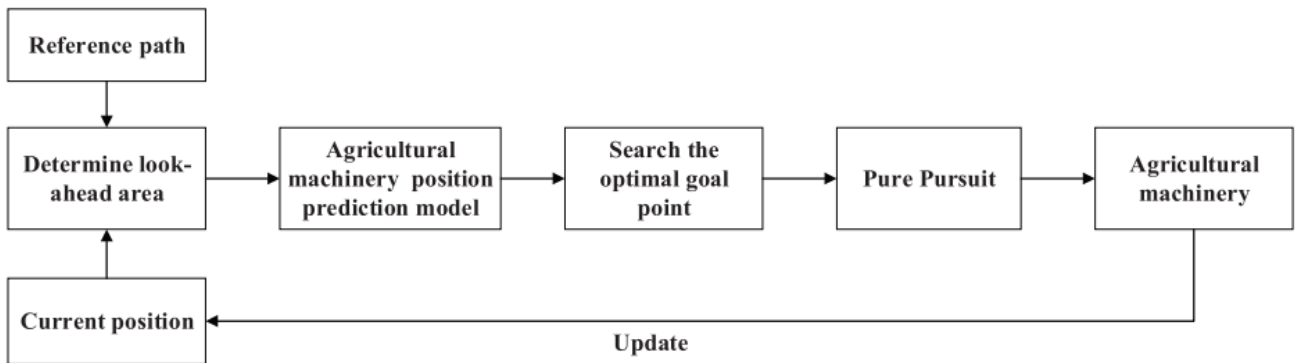


Figure 2.7: Path tracking algorithm framework.

In their implementation Huanghai Golden Horse 554 tractor was used as a test platform for field testing. Dry and slippery track conditions are loamy clay. The road conditions are bumpy was used Due to the irregularities within the field, the tractor experiences tiny oscillations in the front wheel angle throughout the chase supported each algorithms, that is usually swish and they found that the optimal aiming point algorithm proposed in this article can accurately and smoothly track the homing path, which is a significant improvement compared to the Pure Pursuit algorithm. Due to the bumpy road conditions, the lateral errors fluctuated, but the algorithm for this still kept good stability.

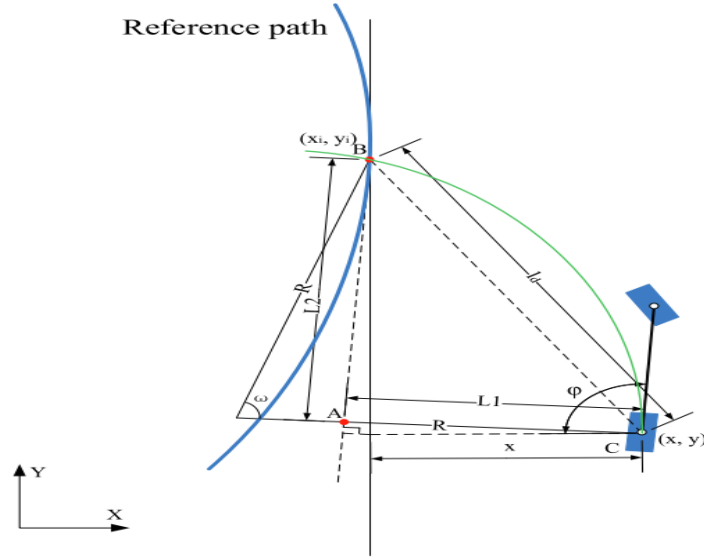


Figure 2.8: Pure Tracking Model, Where (x_i, y_i) Are Target Point Coordinates, (x, y) Are Rear Axle Center Coordinates, R Is Steering Radius.

The look-ahead space associated degree establishes an analysis operation to look for the optimum goal purpose in line with the agricultural machinery position prediction model. The distinction between it and my study was that I simulated a scenario driving to assume a road by using MATLAB to work out the look-ahead distance from it.

In Bingzhan Zhang, Zhiyuan Li, Yaoyao Ni and Yujie Li the approach represented [9] investigated to solve the curvature discontinuity problem in traditional combined arc-line-curve path planning, a quantic polynomial is used to smooth the path. In order to this issue they suggested the control based on a pure tracking algorithm is used as the comparison algorithm for route tracking and their methodology were based on. The practicability of the controller is verified by building a Simulink/CarSim co-simulation platform. In addition, the usefulness of the parking controller is additionally verified by exploitation the ROS intelligent automobile within the laboratory environment. And these algorithms included Parallel Parking Path Planning Based on Quantic Polynomial and motion the control of the path by depending on MPC to prove their contribution they used the algorithm based on MPC built by Simulink is duplicated in IPC by C++ language they concluded the designed route follower can better control the vehicle's following on the desired route. The ROS sensible automotive in an exceedingly laboratory atmosphere to verify the particular control impact of the control model and directly observe the take a look at results.

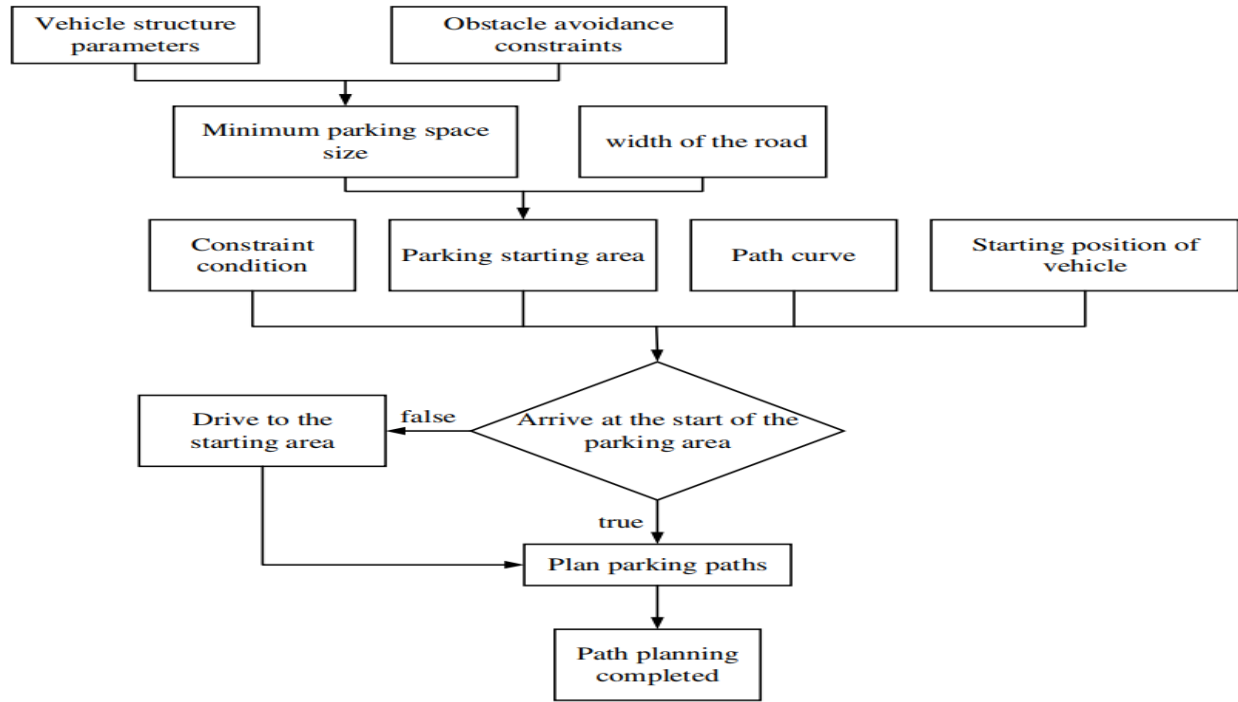


Figure 2.9 : Path Planning Structural Schematic Diagram

In this study was not concerned with the look-ahead distance, nor was it concerned with the geometry used to track the car to a point on the track, whether in the X or Y axes.

In Gabriel M. Hoffmann, Claire J. Tomlin Michael Montemerlo and Sebastian Thrun paper[10]. Targeted to nonlinear control law for an automobile to autonomously track a trajectory, provided in real-time, on rapidly varying methods may suffer from a lack of overall stability, a lack of tracking accuracy, or a reliance on slick road surfaces, all of which could result in the vehicle stalling off-road during autonomous driving due to they presented the tracking the vehicle's path in a new way by considering the orientation of the front wheels (not the vehicle's body) with respect to the desired path, allowing for instant system control. Using the kinematic equations of motion, a control law is designed for which the global asymptotic stability is tested. This control law is then extended to handle the dynamics of the pneumatic tires and power steering wheel. By a switching proportional integral (PI) controller. And their solution were based on the Kinematic Model, Dynamic Model and using these to make the lateral and longitudinal controls and using the Stanley model to study the Off-Road Tests and Endurance Test and they found the ability to accurately track trajectories in a variety of off-road environments, between hazards and on bumpy and winding roads, with a typical RMS cross-tracking error of less than 0.1m. Stanley was the only vehicle among 40 competitors. To avoid hitting an obstacle or missing a door. In addition the same strategy was used to represent the Stanley model using the MATLAB simulation program.

In Azza Elsayed Bayoumi Ibrahim the research represent. [11] The tackled of the Autonomous vehicle behavior in path following problems where steering performance decreases when driving on curved roads in order to tackle this problem they suggested used The pure pursuit controller is a good choice to solve this problem because of its simplicity, accuracy, and computational speed, but tuning its parameters to produce proper custom behavior takes a lot of effort. Autonomous calculation of the characteristic curve of a curve The procedure for managing the change in curvature of the roadway was offered as a new idea for self-adjustment of an anticipation distance (a look-ahead distance) and their methodology were based on first splits the path into lines and parts of circular arcs, and then extracts the information from the curve that will be used to later calculate the correct look ahead distances. In addition, the PI controller, in conjunction with the pure search controller, serves to reduce the lag error that is present. To demonstrate the efficiency of the proposed method, the full algorithms were implemented in the MATLAB/Simulink environment for different road shapes and the MATLAB Driving Scenario Designer was used to create roads. Fulfilment of their objectives three cases was studied in this article In the first case, the route segmentation algorithm splits the route into two curve segments and three straight line segments and they concluded that to proposed algorithms adjust look ahead distance in response to changes in road curvature, then calculate the correct steering angle for the vehicle to precisely follow the desired path. Additionally, the combination of a pure tracking controller with a feedback PI controller improves vehicle tracking accuracy. And automation. The results show that the autonomous vehicle maintains a high level of accuracy in its route. And the novel strategy can greatly minimize lateral errors on curved roads while maintaining acceptable tracking accuracy.

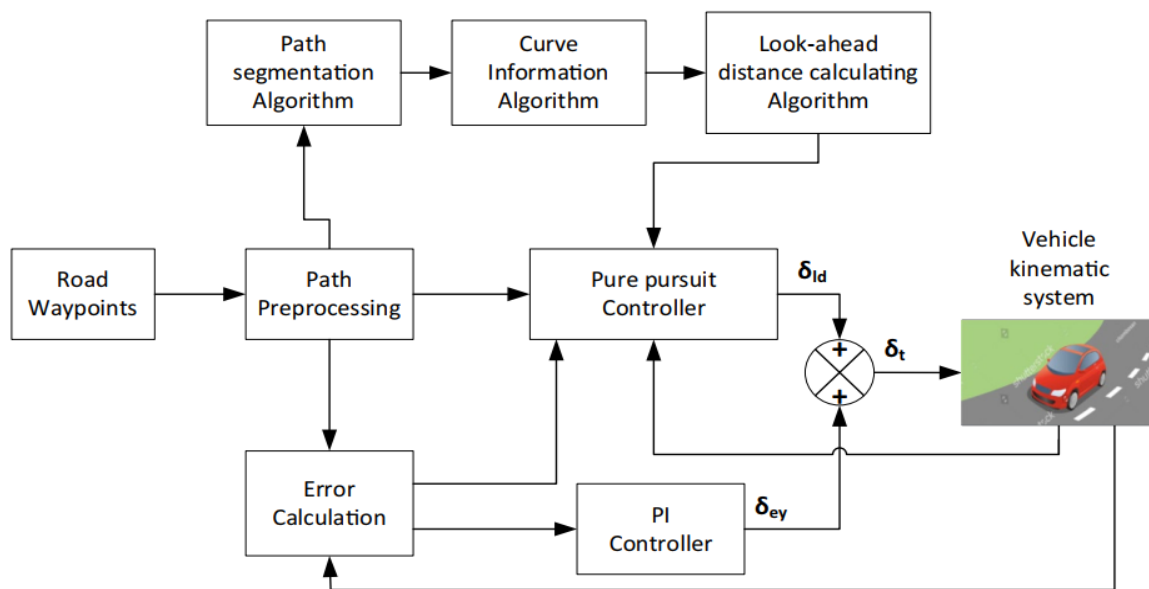


Figure 2.10 : Block diagram of the proposed steering control system

This method in the article is a similar method to the method I used in principle, but there is a difference in the shapes of the methods drawn using the MATLAB Driving Scenario, and there is a difference in the Curvature.



CHAPTER 3. PURE PURSUIT WHEN THE VEHICLE FOLLOWING THE X AND Y AXIS

3.1. INTRODUCTION

The performance of the geometric trajectory tracking controller depends on the look ahead distance. The forward distance is similar to the distance between the vehicle and a point in front of the vehicle, and a human driver can look carefully to follow the road. Therefore, the robotic vehicle must have a variable forward visibility in relation to speed. The faster the robotic vehicle moves, the greater the forward visibility should be. Since the longer forward view allows the robotic vehicle to turn earlier than the shorter forward view, a shorter forward view is required for accurate route navigation. However, if the robotic vehicle has a high speed and a short distance ahead, the robotic vehicle will have a high angular velocity when following the path, which may cause the robotic vehicle to skid. [5]

3.2. PURE PURSUIT WHEN THE VEHICLE FOLLOEING THE X AXIS

Simulink has been created to show you how to track the vehicle to the X-Axis, and this is shown in the figure below.

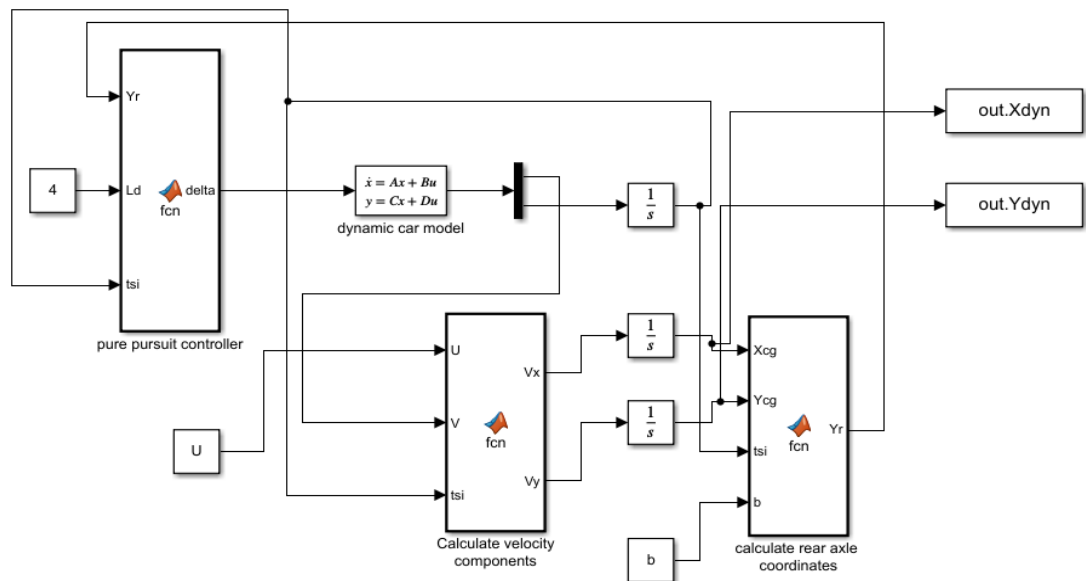


Figure 3.1: Pure Pursuit Stimulant. When the Vehicle Following the X Axis

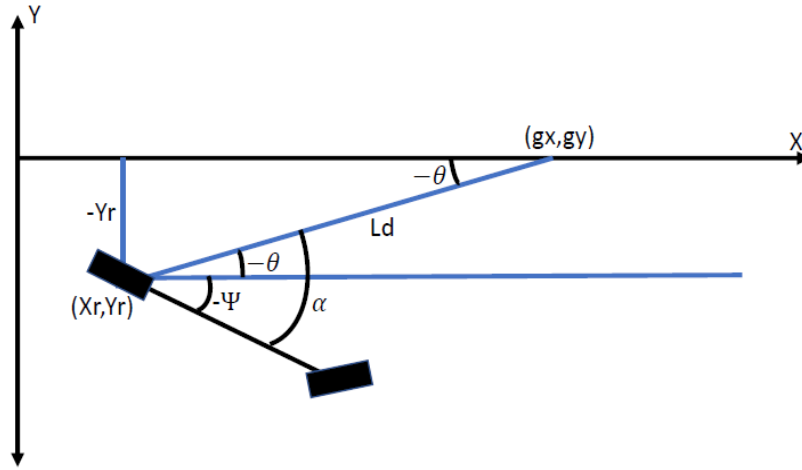


Figure 3.2: Simple Test When the Vehicle Following a Straight Line in X Axis

$$\alpha = -\theta - \Psi = -\sin^{-1}\left(\frac{Y_r}{L_d}\right) - \Psi$$

The MATLAB function code in Stimulant (**Pure Pursuit Controller**):

```
function delta= fcn(Yr,Ld,tsi)
L=2.2;
alpha=-asin(Yr/Ld)-tsi;
delta=atan(2*L*sin(alpha)/Ld);
```

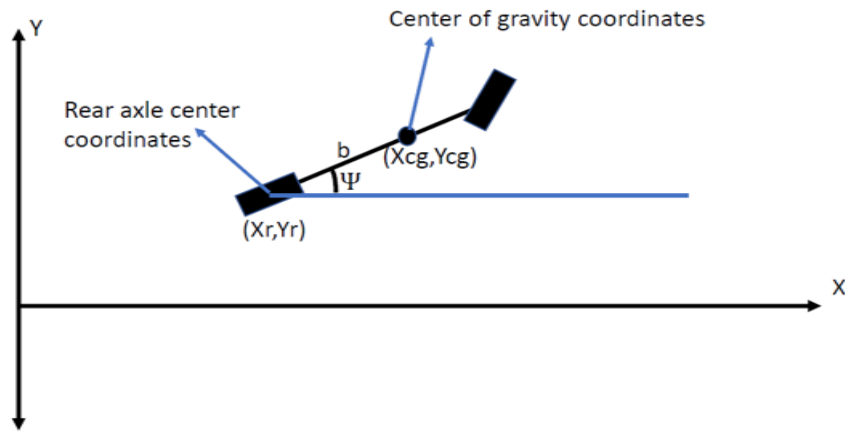


Figure 3.3: Finding the Coordinates of Rear Axle Center When the Vehicle Following the X-Axis

$$X_r = X_{cg} - b \cos(\psi)$$

$$Y_r = Y_{cg} - b \sin(\psi)$$

The MATLAB function code in Stimulant (**calculate the rear axle coordinates**):

```
function [Xr,Yr] = fcn(Xcg,Ycg,tsi,bd)
Xr=Xcg-bd*cos(tsi);
Yr=Ycg-bd*sin(tsi);
```

The last MATLAB function in the Stimulant to **calculate the velocity coordinates** and the code shown below explain this:

```
function [Vx,Vy]= fcn(U,V,tsi)
```

```
Vx=U*cos(tsi)-V*sin(tsi);
```

```
Vy=U*sin(tsi)+V*cos(tsi);
```

3.3. PURE PURSUIT WHEN THE VEHICLE FOLLOEING THE Y AXIS

Simulink has been created to show you how to track the vehicle to the Y-Axis, and this is shown in the figure below.

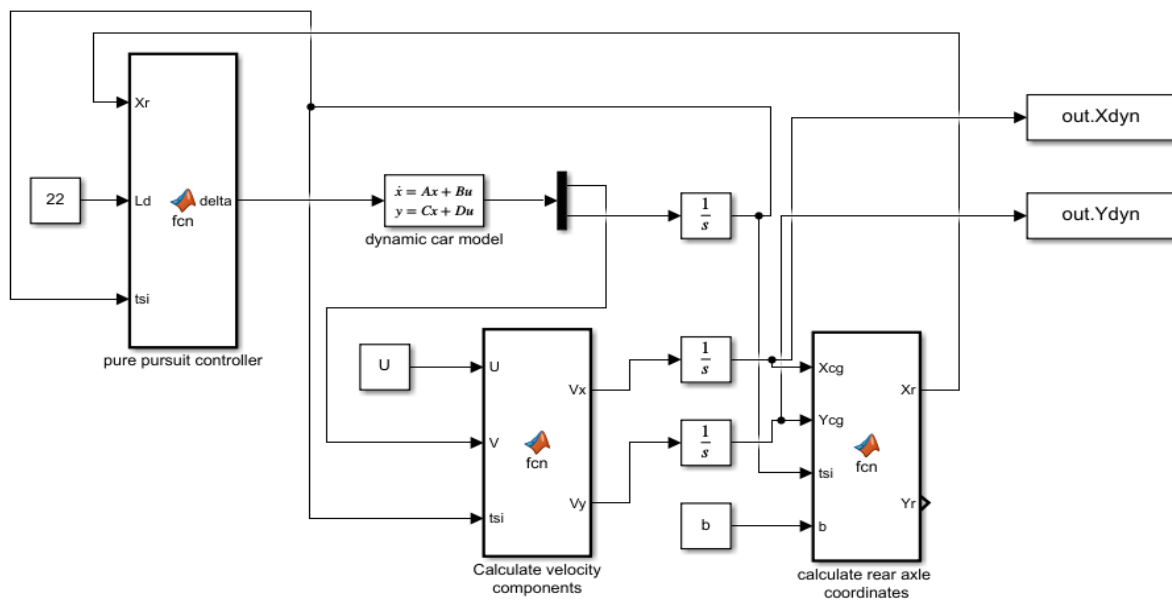


Figure 3.4: Pure Pursuit Stimulant When the Vehicle Following the Y Axis

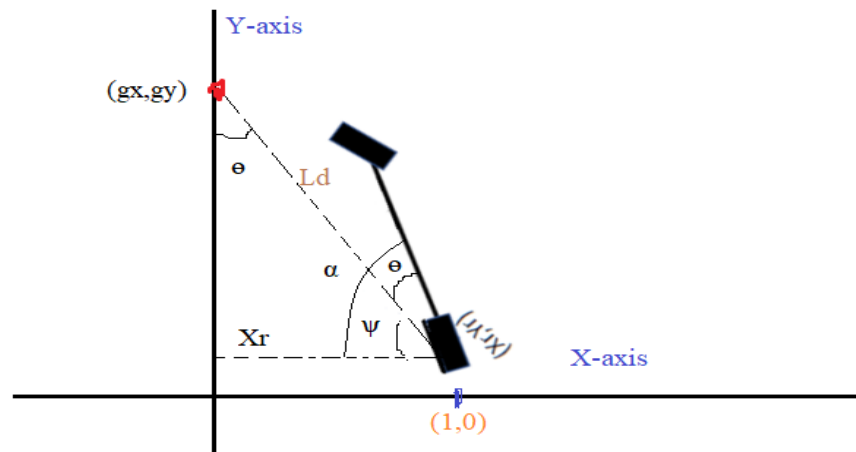


Figure 3.5: Simple Test When the Vehicle Following a Straight Line in Y Axis

$$\alpha = \theta + \Psi = \cos^{-1} \left(\frac{X_r}{L_d} \right) + \Psi$$

The MATLAB function code in Stimulant (**Pure Pursuit Controller**):

```
function delta= fcn(Xr,Ld,tsi)
L=2.2;
alpha=acos(Xr/Ld)+tsi;
delta=atan(2*L*sin(alpha)/Ld);
```

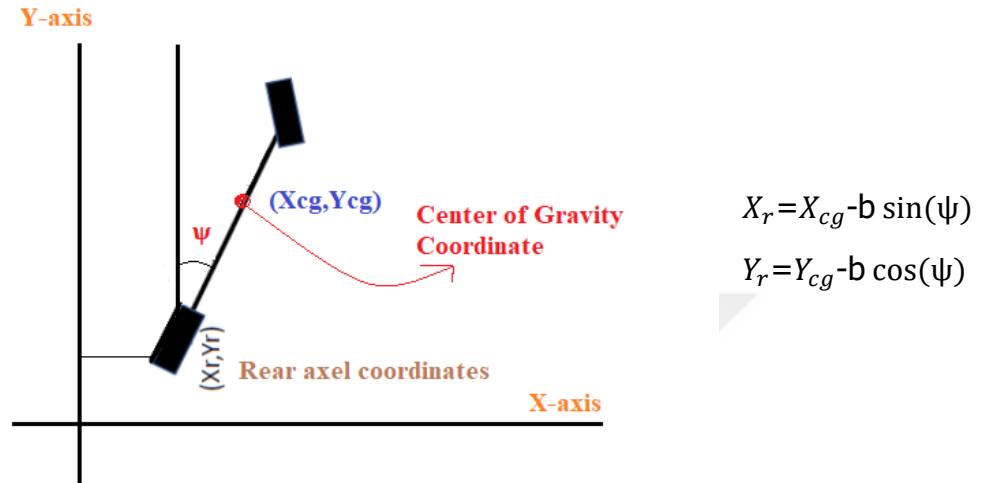


Figure 3.6: Finding the Coordinates of Rear Axle Center When the Vehicle Following the Y-Axis

The MATLAB function code in Stimulant (**calculate the rear axle coordinates**):

```
function [Xr,Yr] = fcn(Xcg,Ycg,tsi,bd)
Xr=Xcg-bd*sin(tsi);
Yr=Ycg-bd*cos(tsi);
```

The last MATLAB function in the Stimulant to **calculate the velocity coordinates** and the code shown below explain this:

```
function [Vx,Vy]= fcn(U,V,tsi)
Vx=U*cos(tsi)-V*sin(tsi);
Vy=U*sin(tsi)+V*cos(tsi);
```

The parameter code below was written to apply it in the Stimulant at U=15m/s with 3 different look ahead distance to draw how **the vehicle following the path in X and Y axis** as the equations explained previously.

```
%The parameters of the vehicle
```

```
clear all
```

```
mass=1250 %mass of vehicle
```

```
Iz=1550;
```

```
L=2.2;
```

```
ad=1;
```

```
bd=1.2;
```

```
Cfront=50000;
```

```
Crear=50000;
```

```
U=15;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

Note that the value of look ahead distance has been changed to (22, 12, and 4), and the results of these changing was explained below by Stimulant and plots. The vehicle starts from initial coordinates (1, 0) and an initial angle of 30 degree with X and Y axis that should be change from integrators in the Stimulant and the figures below explains that.

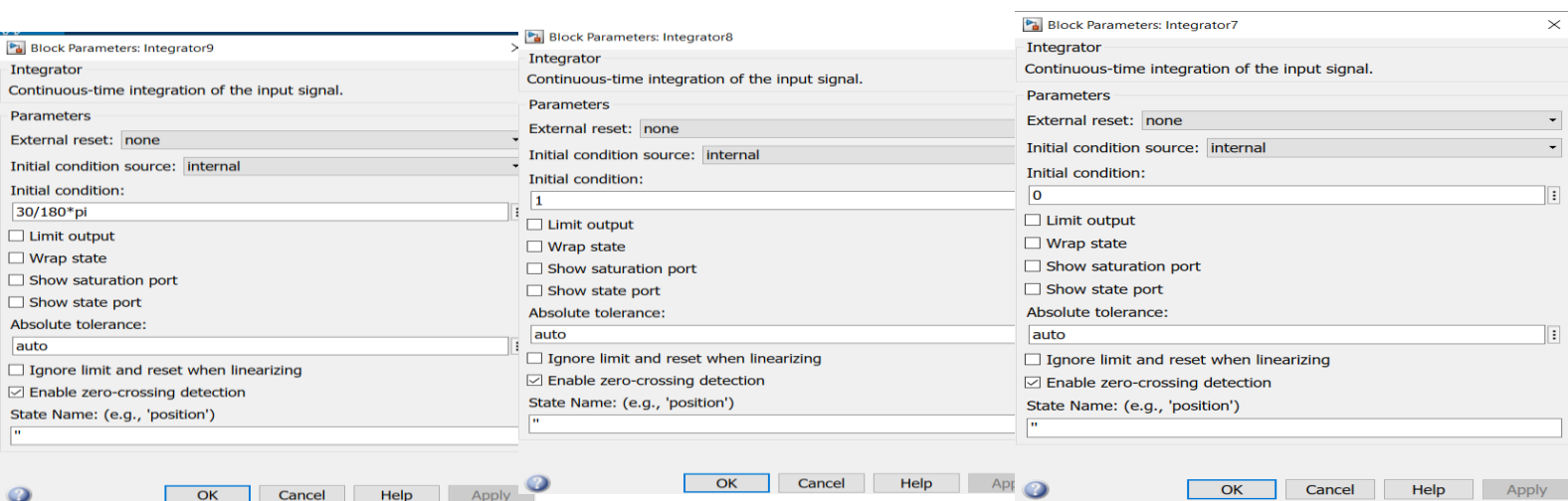


Figure 3.7: The Initial Conditions When the Vehicle Starts From Initial Coordinates (1, 0) and an Initial Angle of 30 Degree with X, Y Axis

3.4. APPLY THE PURE PURSUIT MODEL WHEN VEHICLE FOLLOWING THE Y-AXIS

The tracking of looking head distance has been applied with different values of l_d in meters

3.4.1. For l_d is equal 22 m when the vehicle following the Y-Axis

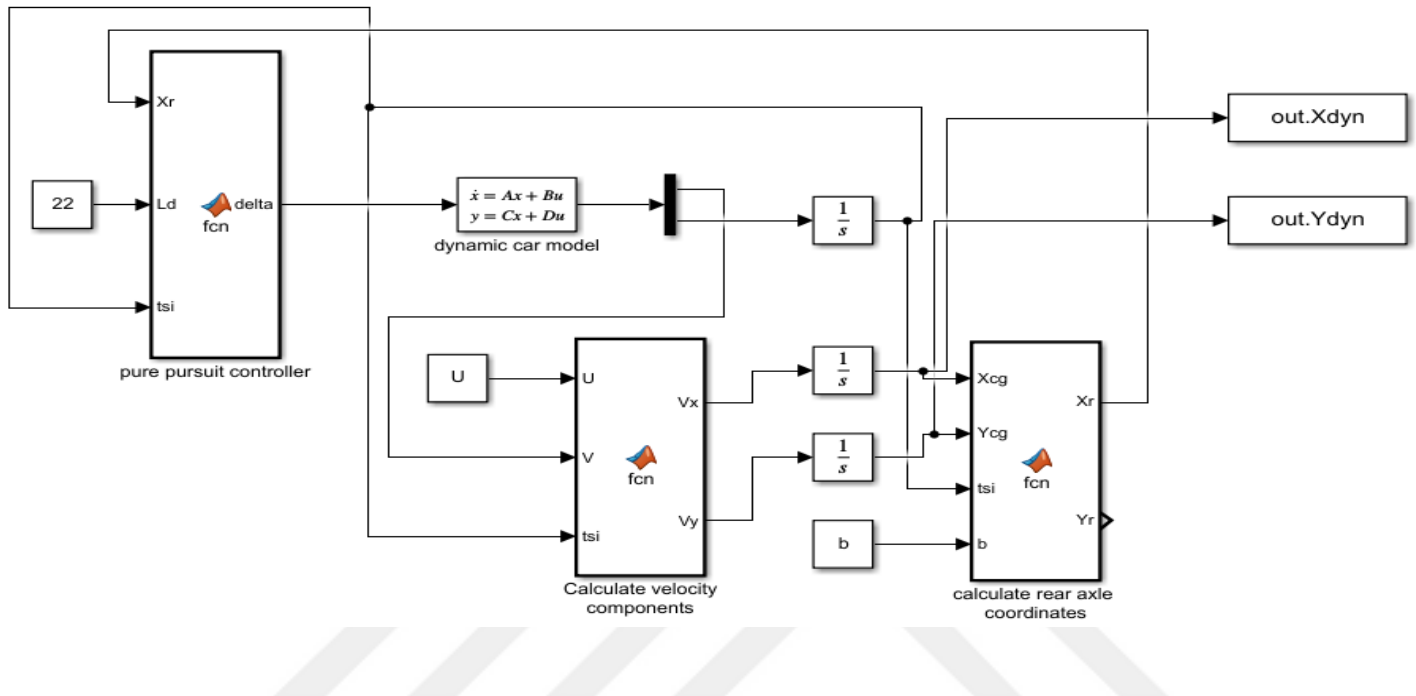


Figure 3.9: Simulant When ($L_d=22$ m) When Vehicle Following the Y- Axis

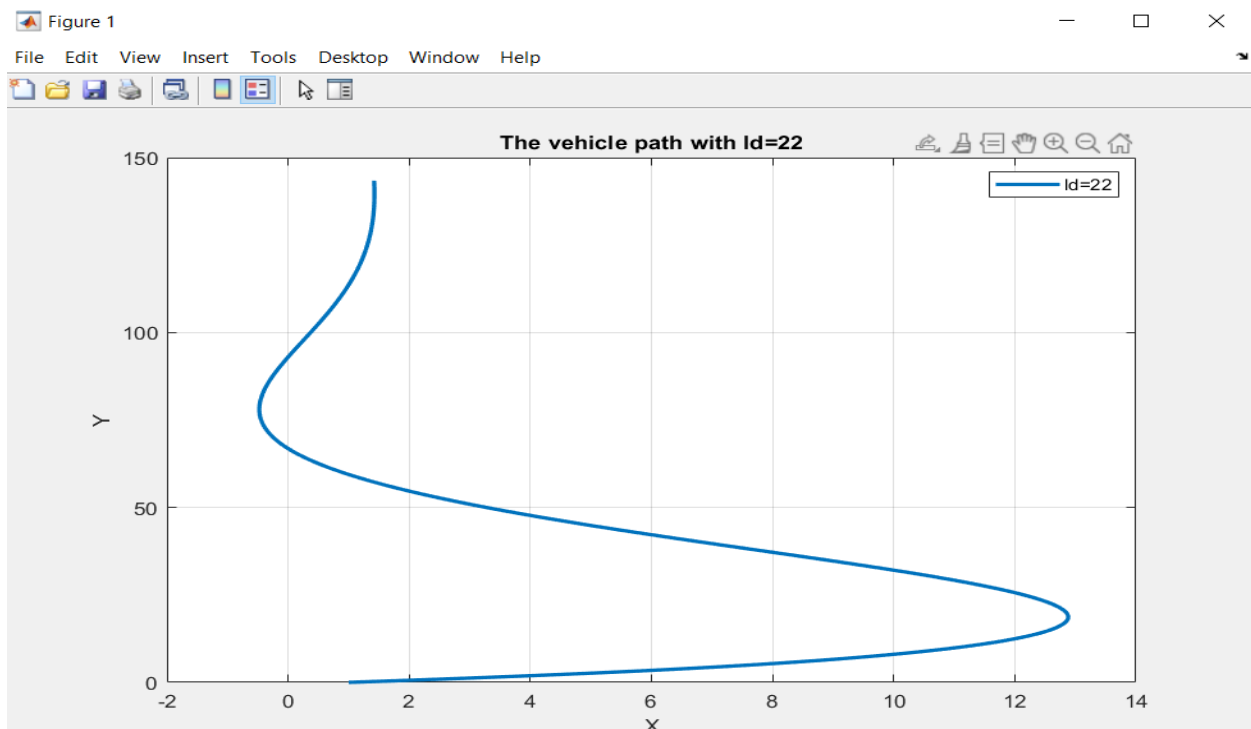


Figure 3.8: The Vehicle Path When ($L_d=22$ m) When Vehicle Following the Y- Axis

3.4.2. For l_d is equal 12 m when the vehicle following the Y-Axis

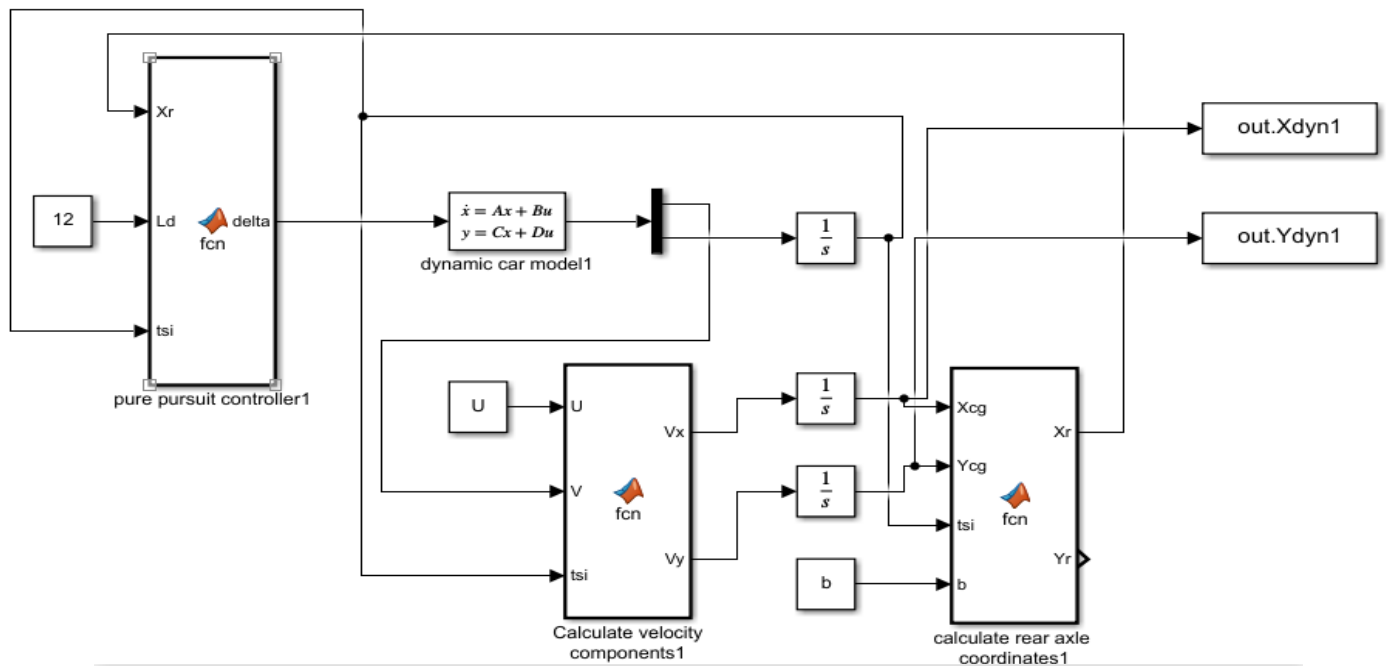


Figure 1

Figure 3.11: Stimulant When ($L_d=12$ m) When Vehicle Following the Y- Axis

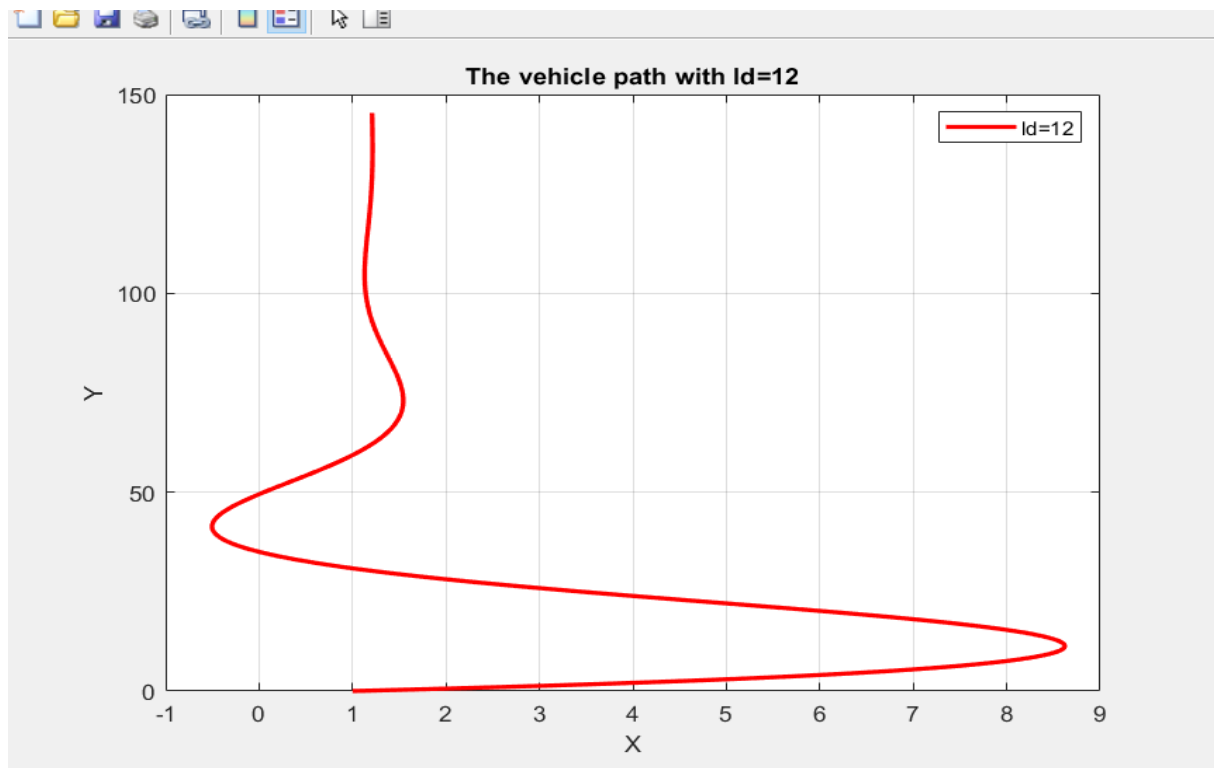


Figure 3.10: The Vehicle Path When ($L_d=12$ m) When Vehicle Following the Y- Axis

3.4.3. For l_d is equal 4 m when the vehicle following the Y-Axis

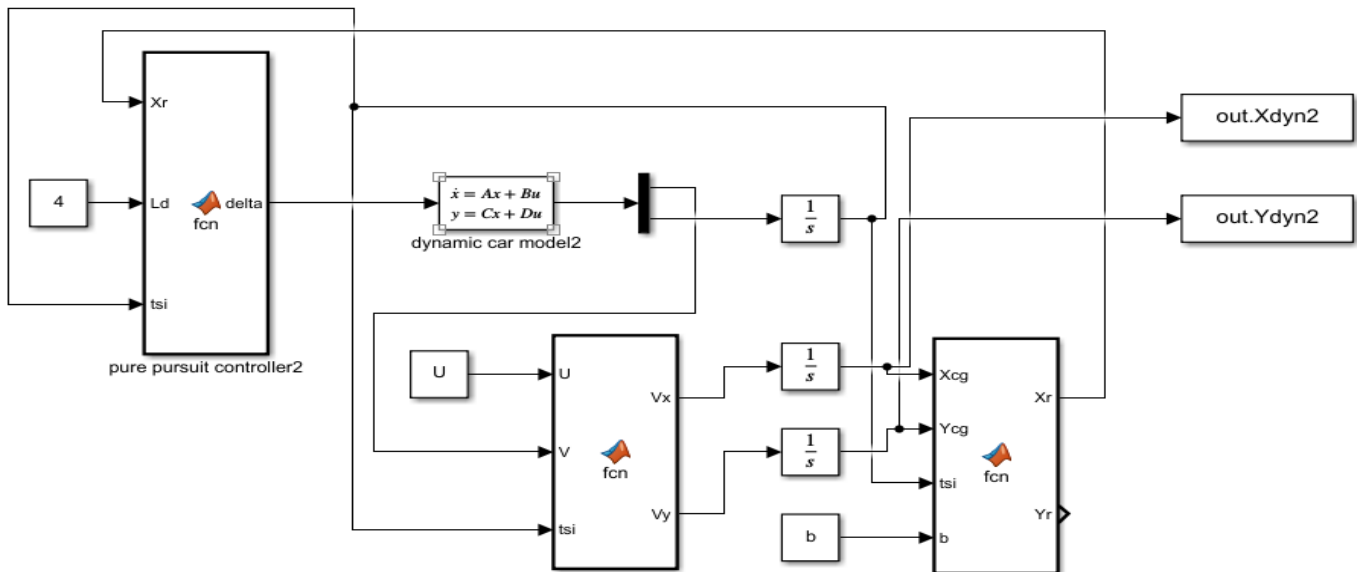


Figure 3.13: Stimulant When ($L_d=4$ m) When Vehicle Following the Y- Axis

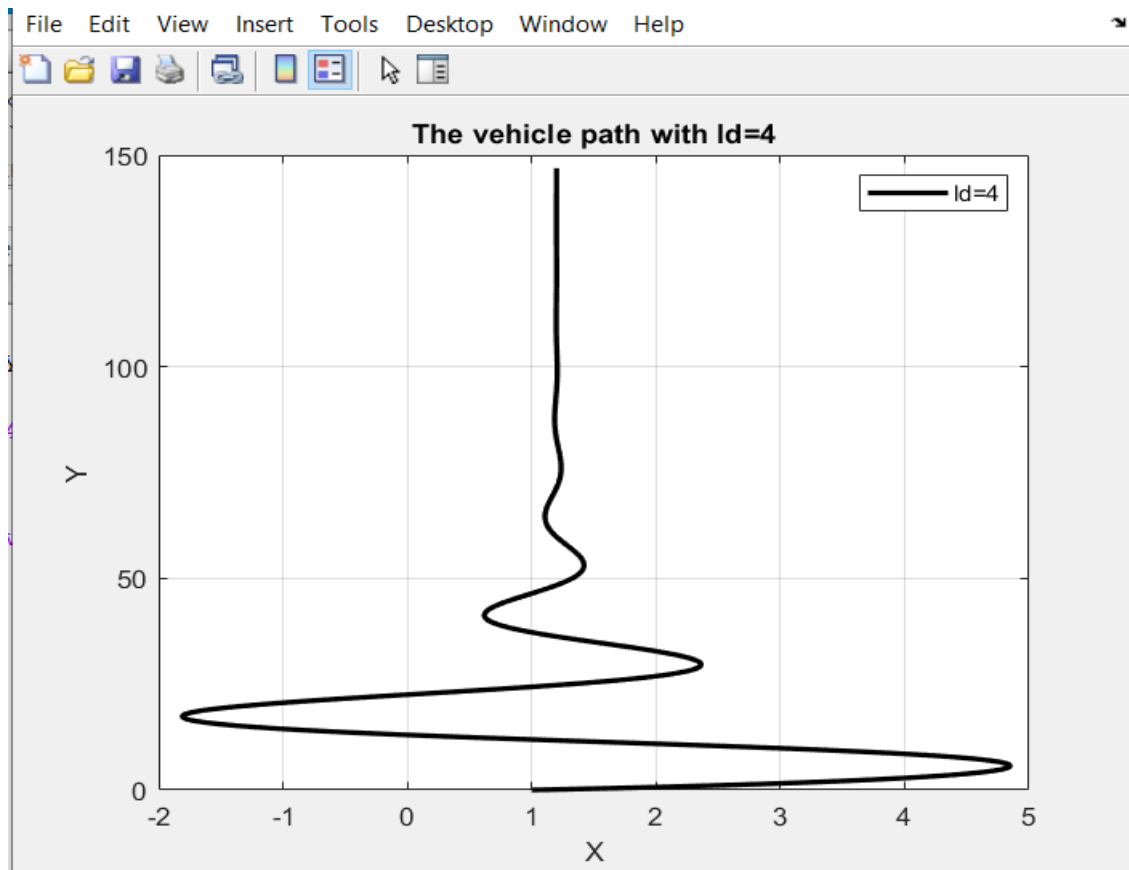


Figure 3.12: The vehicle path when ($L_d=4$ m) when vehicle following the Y- axis

The previous drawings were collected into one drawing through a specific code, that was written as shown below, to facilitate the study of the three cases, at 3 different looking ahead distance to the vehicle.

```
%The vehicle path plotter
plot(out.Xdyn,out.Ydyn,out.Xdyn1,out.Ydyn1,out.Xdyn2,out.Ydyn2,'LineWidth',2)
grid on
legend('ld=22','ld=12','ld=4')
xlabel('X')
ylabel('Y')
title('The vehicle path for 3 Different look ahead distances')
```

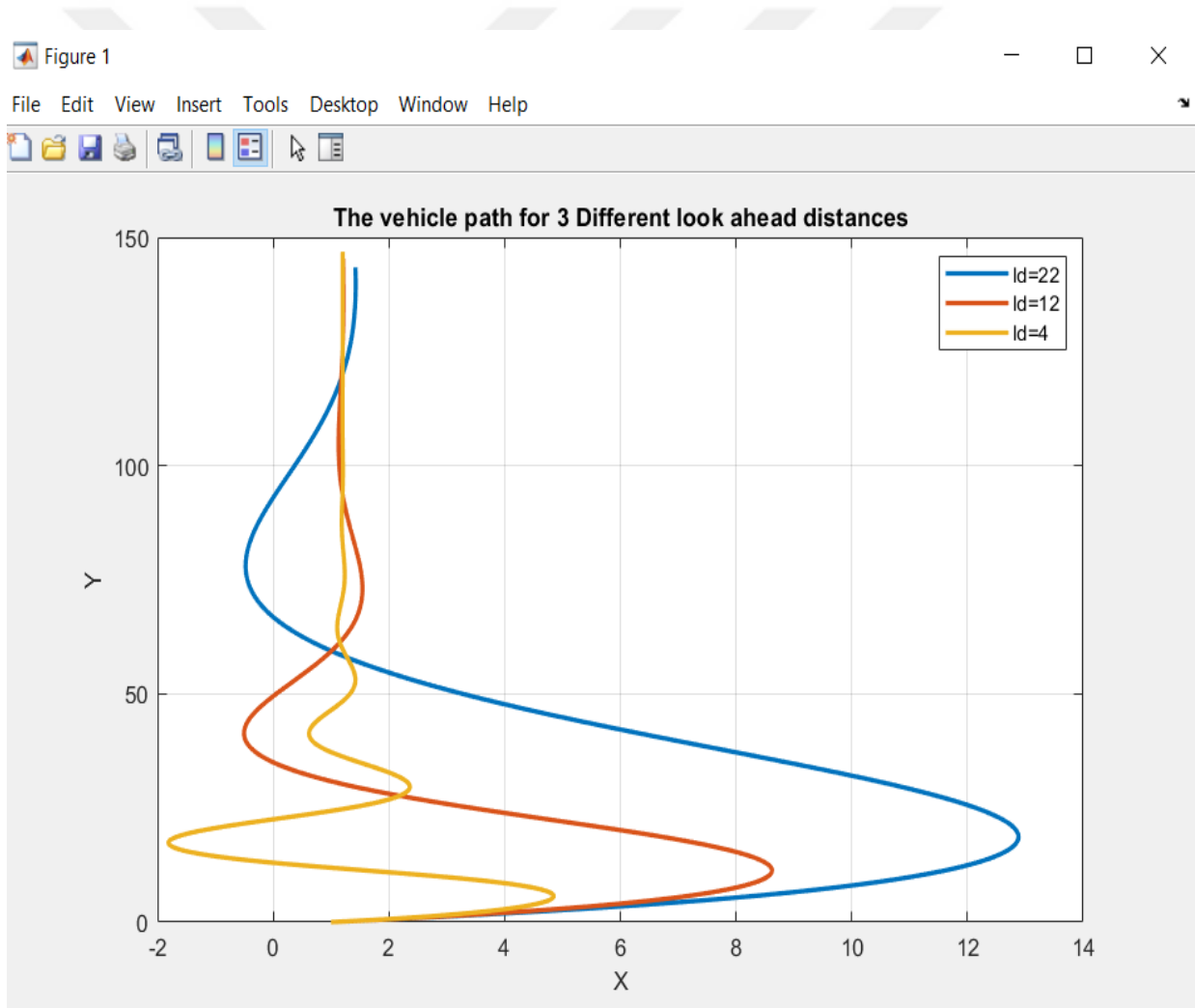


Figure 3.14: The Vehicle Path with 3 Different Looking Ahead Distance (22, 12 And 4 m) When Vehicle Following the Y-Axis

3.4.4. Discuss the Results When the L_d Has Been Changed (22, 12 and 4) When Vehicle

Following the Y- Axis

In this test, the speed of the vehicle is kept constant ($U=15\text{m/s}$). The results are shown in the previous figures, and the pure pursuit tracking controller is tested for different looking ahead distance values. The looking ahead distance basically used as a parameter of the controller, which tells how far the vehicle should find a reference point on the vehicle path? If the look ahead distance decreased (shorter L_d) that's making a turn much faster and the response of it is faster than larger L_d so the shorter L_d has a fast response to follow the Y-axis. However the shorter L_d has too much of oscillations that's mean the system is marginal stable (it seems almost unstable). Whereas when L_d increased gives more stability to the vehicle to tracking the path but it still has large amplitude of wave compared with the shorter L_d . Therefore, looking ahead distance is one of the important factors that greatly affect the vehicle's trajectory and thus greatly affect the controller. Therefore, the appropriate the looking ahead distance must be taken to track the lane well and obtain good stability for the vehicle while it is on the track. The last thing we can said when L_d increased the response became smoother and the system doesn't make a large overshoot.

3.5. APPLY THE PURE PURSUIT MODEL WHEN VEHICLE FOLLOWING THE X-AXIS

The parameter code was used previously has been apply it in the Stimulant at $U=15\text{m/s}$ with 3 different look ahead distance to draw how **the vehicle following the path in X-axis** with different values of l_d in meters

3.5.1. For l_d is equal 22 m when the vehicle following the X-Axis

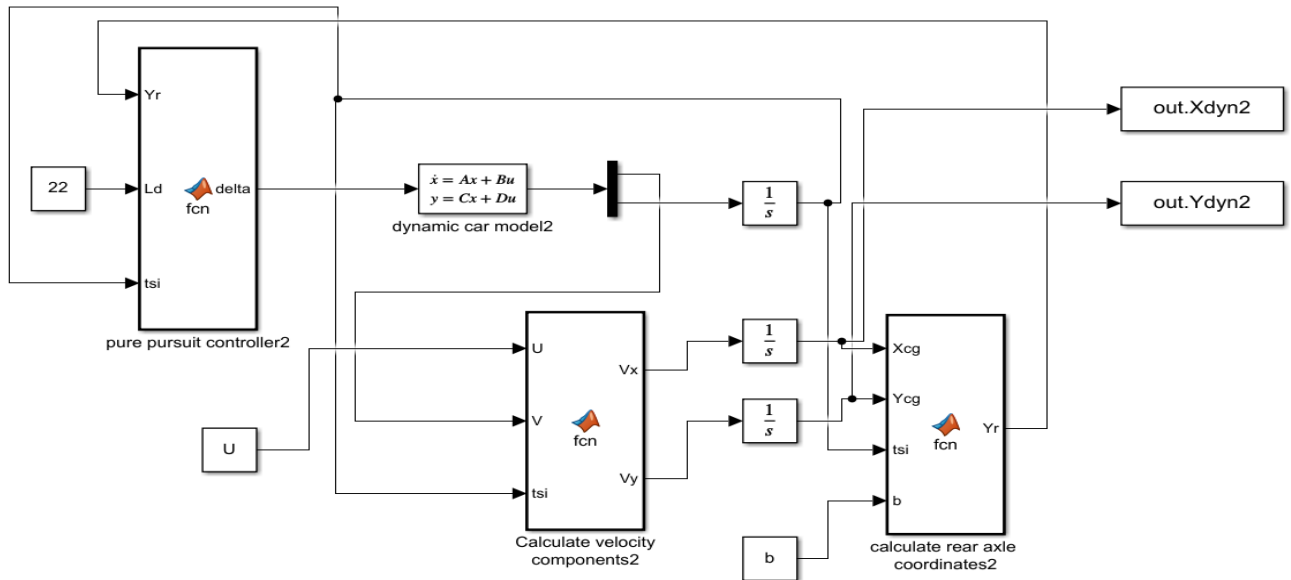


Figure 3.15: Stimulant When ($L_d=22$ m) When Vehicle Following the X- Axis

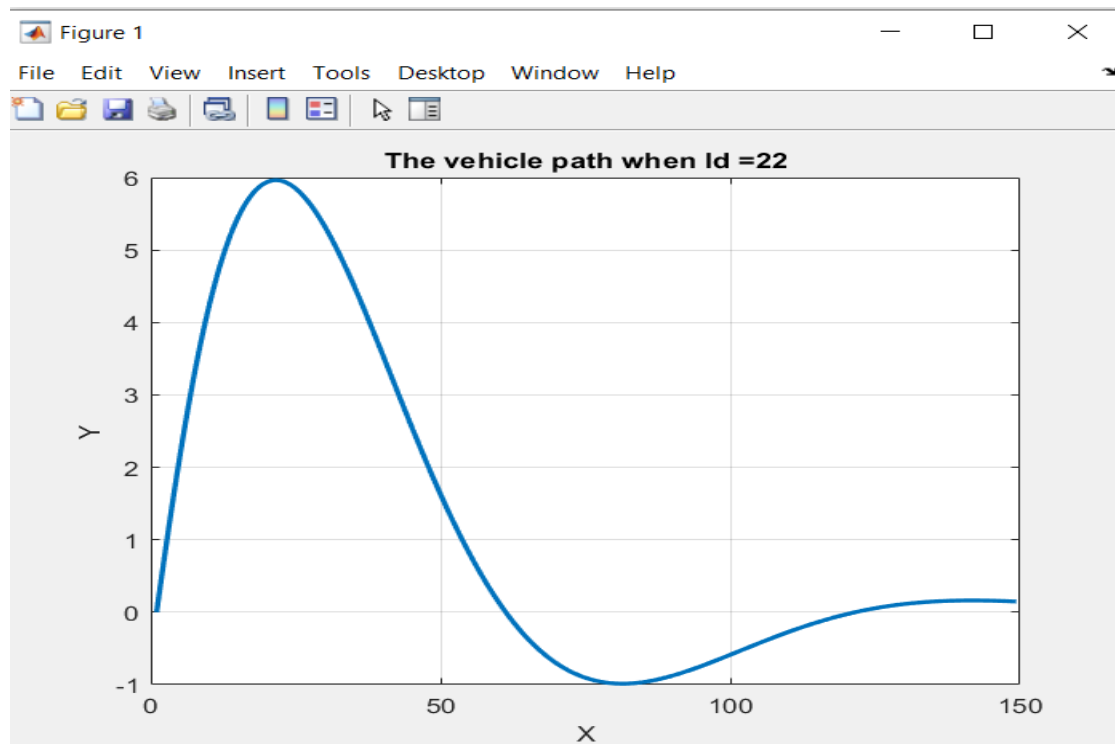


Figure 3.16: The Vehicle Path When ($L_d=22$ m) When Vehicle Following the X- Axis

3.5.2. For L_d is equal 12 m when the vehicle following the X-Axis

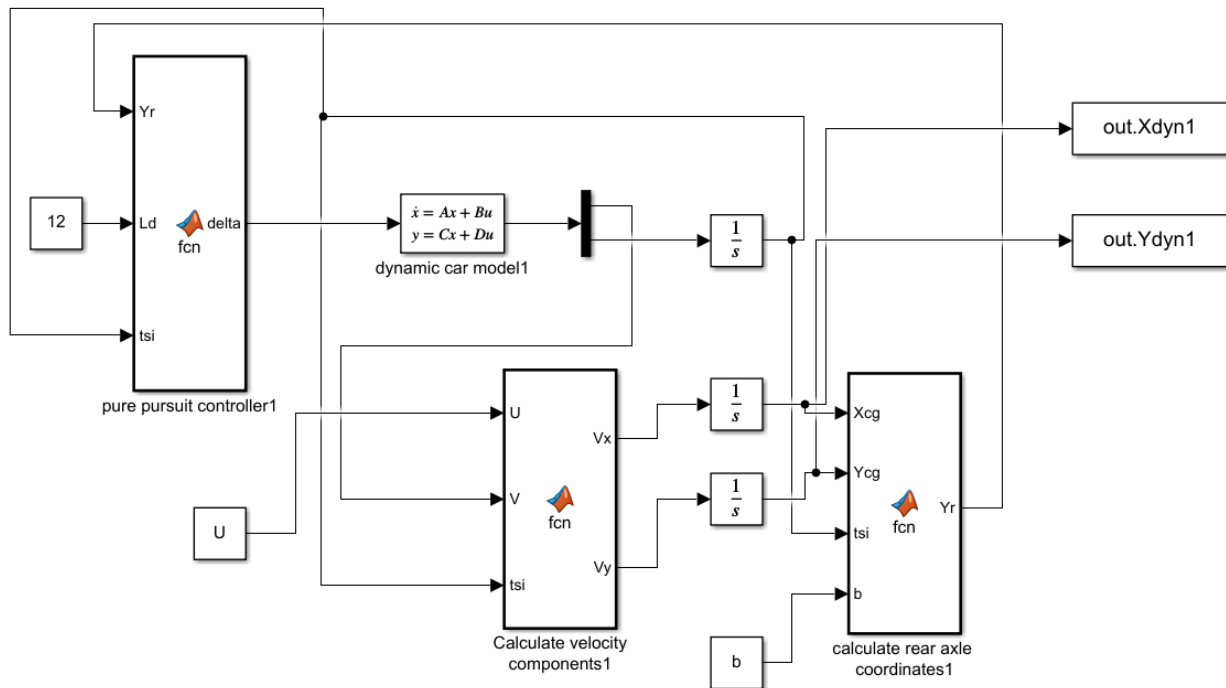


Figure 3.17: Stimulant When ($L_d=12$ m) When Vehicle Following the X- Axis

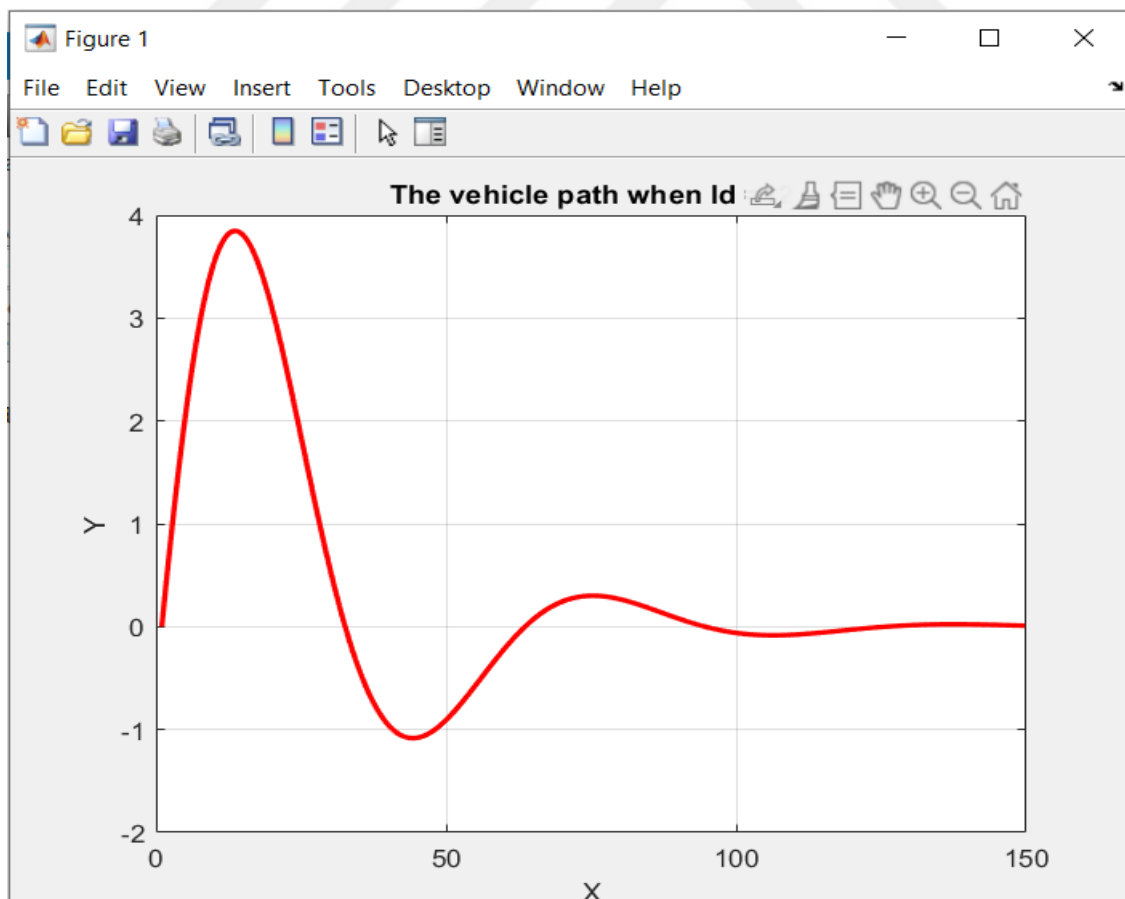


Figure 3.18: The Vehicle Path When ($L_d=12$ m) When Vehicle Following the X- Axis

3.5.3. For L_d is equal 4 m when the vehicle following the X-Axis

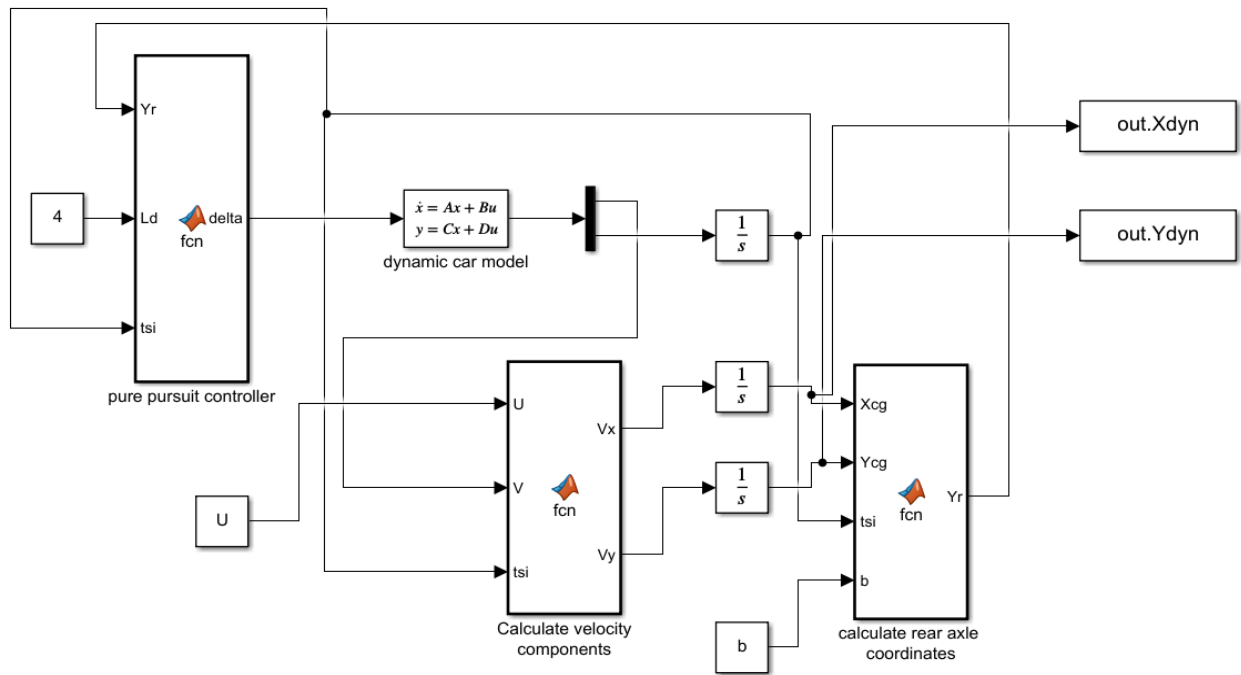


Figure 3.19: Simulink When ($L_d=4$ m) When Vehicle Following the X- Axis

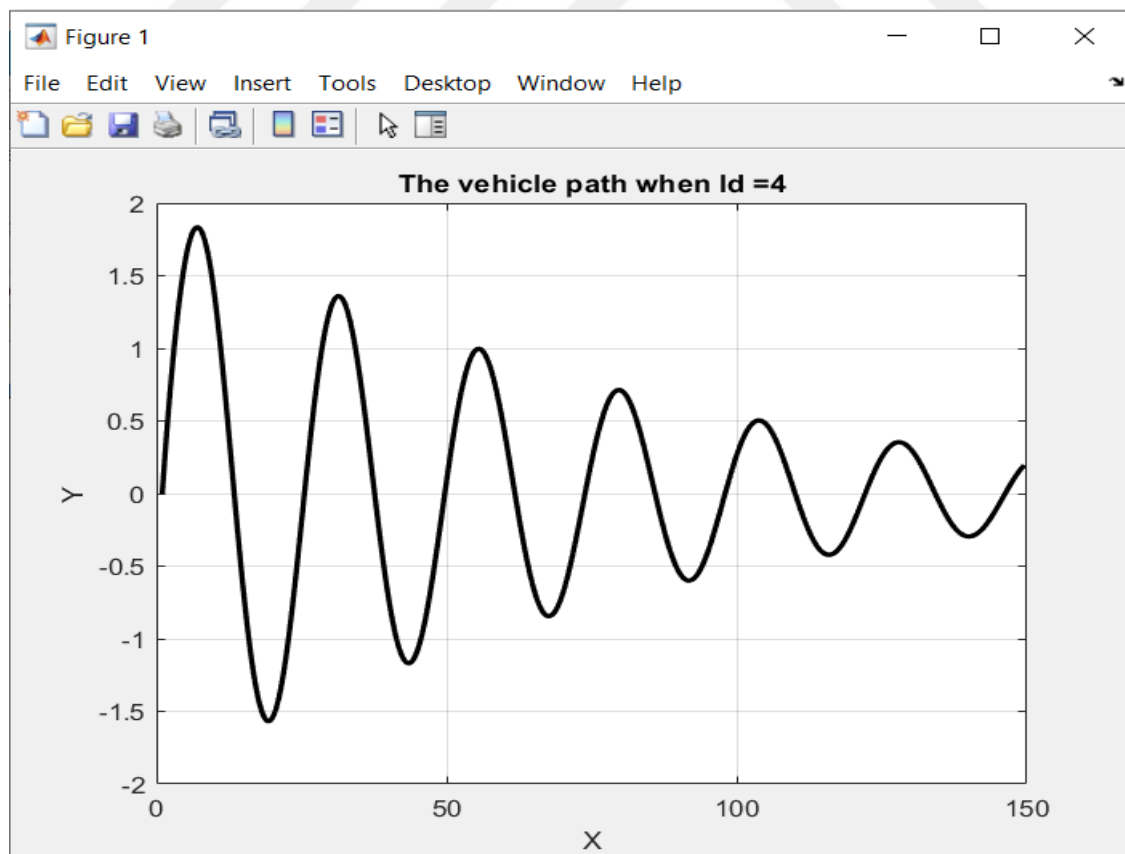


Figure 3.20: The Vehicle Path When ($L_d=4$ m) When Vehicle Following the X- Axis

The previous drawings were collected into one drawing through a specific code that was written as shown below to facilitate the study of the three cases at 3 different looking ahead distance to the vehicle.

```
%The vehicle path plotter
plot(out.Xdyn,out.Ydyn,out.Xdyn1,out.Ydyn1,out.Xdyn2,out.Ydyn2,'LineWidth',2)
grid on
legend('ld1=4','ld2=12','ld3=22')
xlabel('X')
ylabel('Y')
title('The vehicle path for different values of ld when vehicle following the x axis')
```

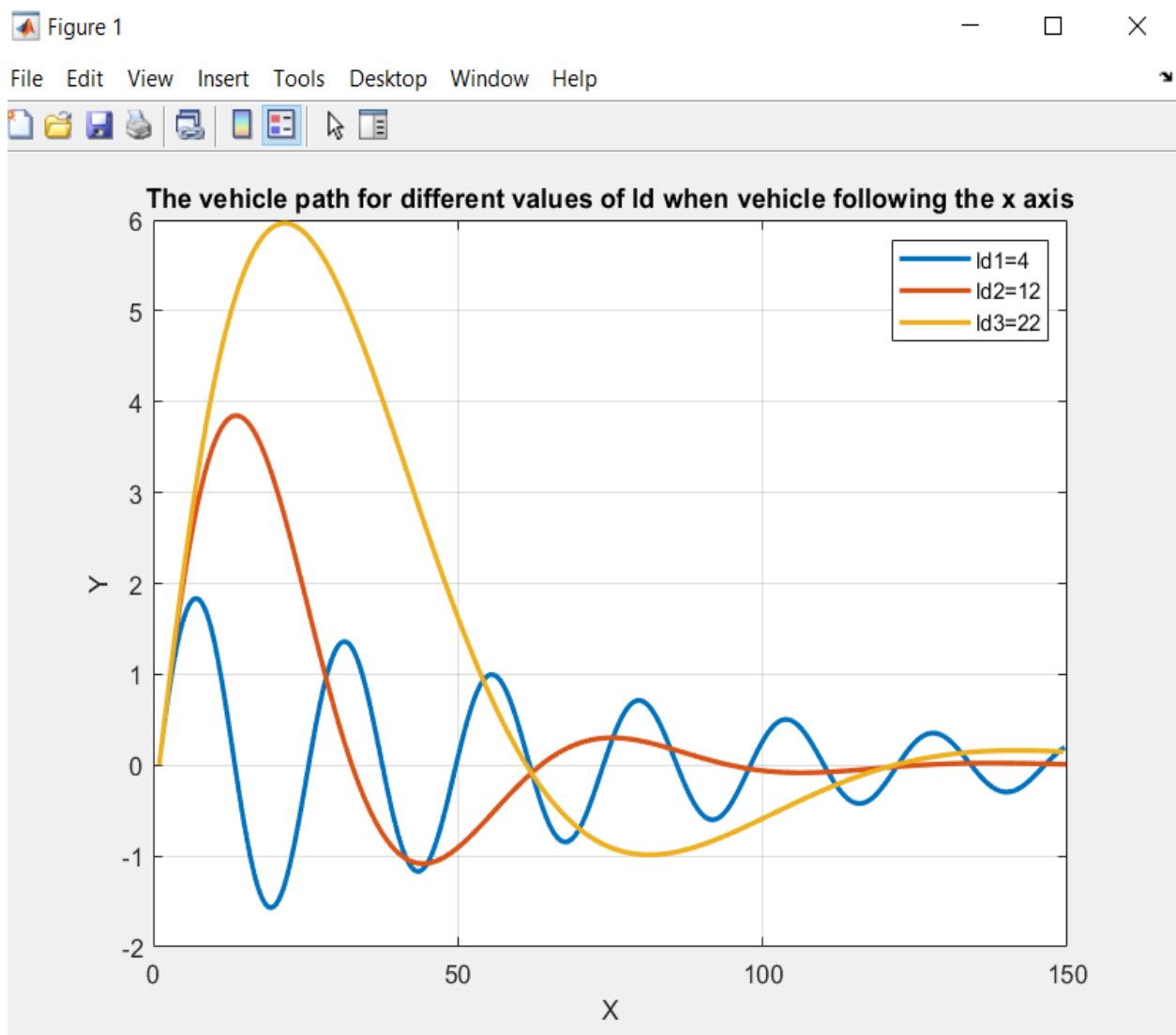


Figure 3.21: The Vehicle Path with 3 Different Looking Ahead Distance (22, 12 and 4 m) When Vehicle Following the X- Axis

3.6. CHANGING THE LONGITUDINAL VEHICLE VELOCITY:

This test of the control is by changing the longitudinal velocity to three different values in order to see the effect of the speed on the control, and this test is important to answer the following question: Will the vehicle remain in a stable state or not with the change of speed to three different values of L_d ? and through the codes for changing the speed in each case and testing it on the simulation in MATLAB, we come to the most important question, which is :when will the control crash happen and what is the effect of that?. The drawings below explain in detail each of the cases, knowing that each of the cases has a special code in which we change the speed of the vehicle. The figure shown below is the Stimulant for the vehicle that follows the Y-axis and X-axis with 3 different L_d .

3.6.1. Changing Longitudinal velocity for pure pursuit model when the vehicle following Y axis

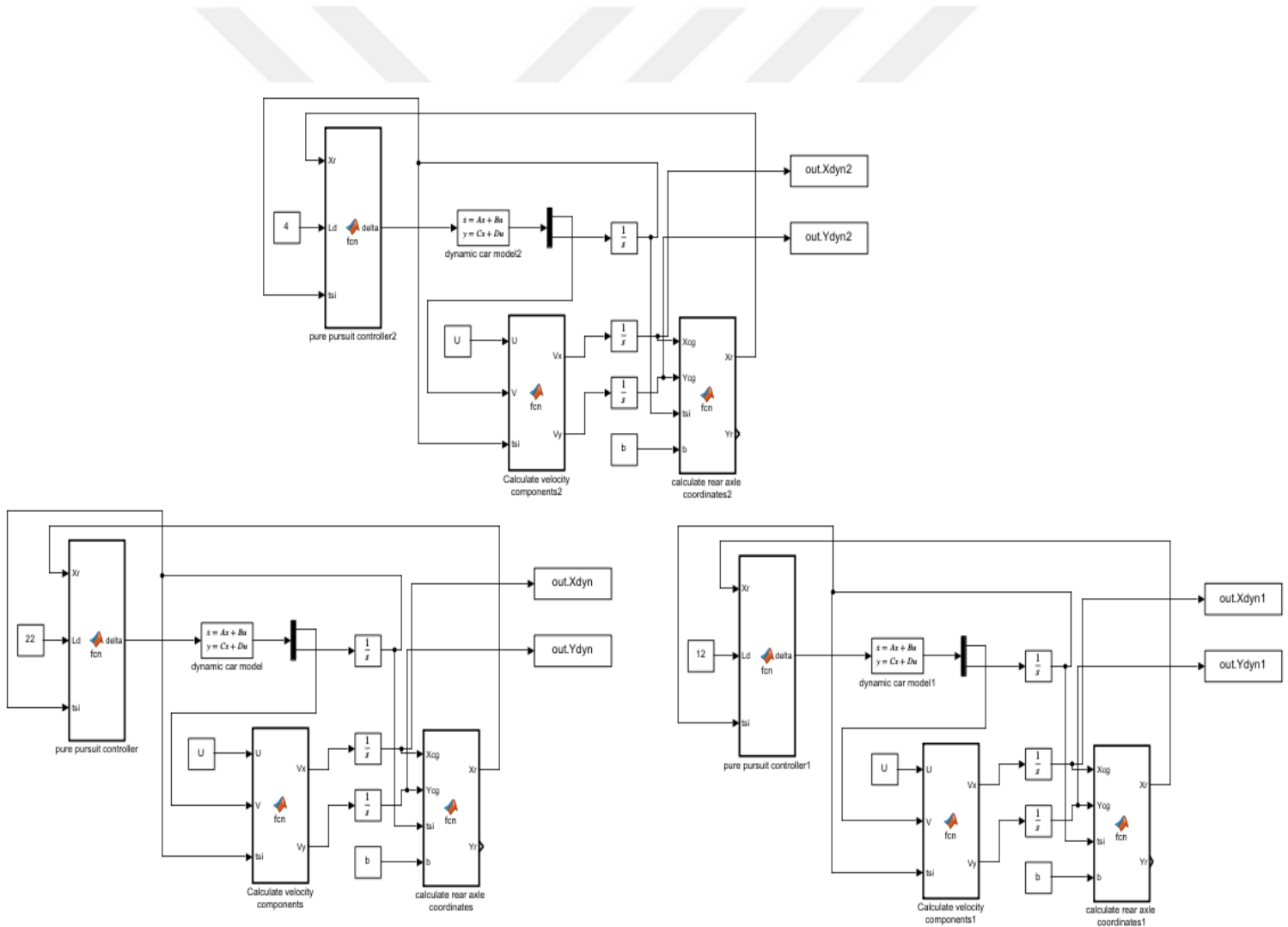


Figure 3.22: Stimulant for the Vehicle That Follows the Y-Axis with 3 Different Values for L_d

When the longitudinal velocity is equal ($U=3\text{m/s}$):

%The parameters of the vehicle

```
clear all
```

```
mass=1250;
```

```
Iz=1550;
```

```
L=2.2;
```

```
ad=1;
```

```
bd=1.2;
```

```
Cfront=50000;
```

```
Crront=50000;
```

```
U=3;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

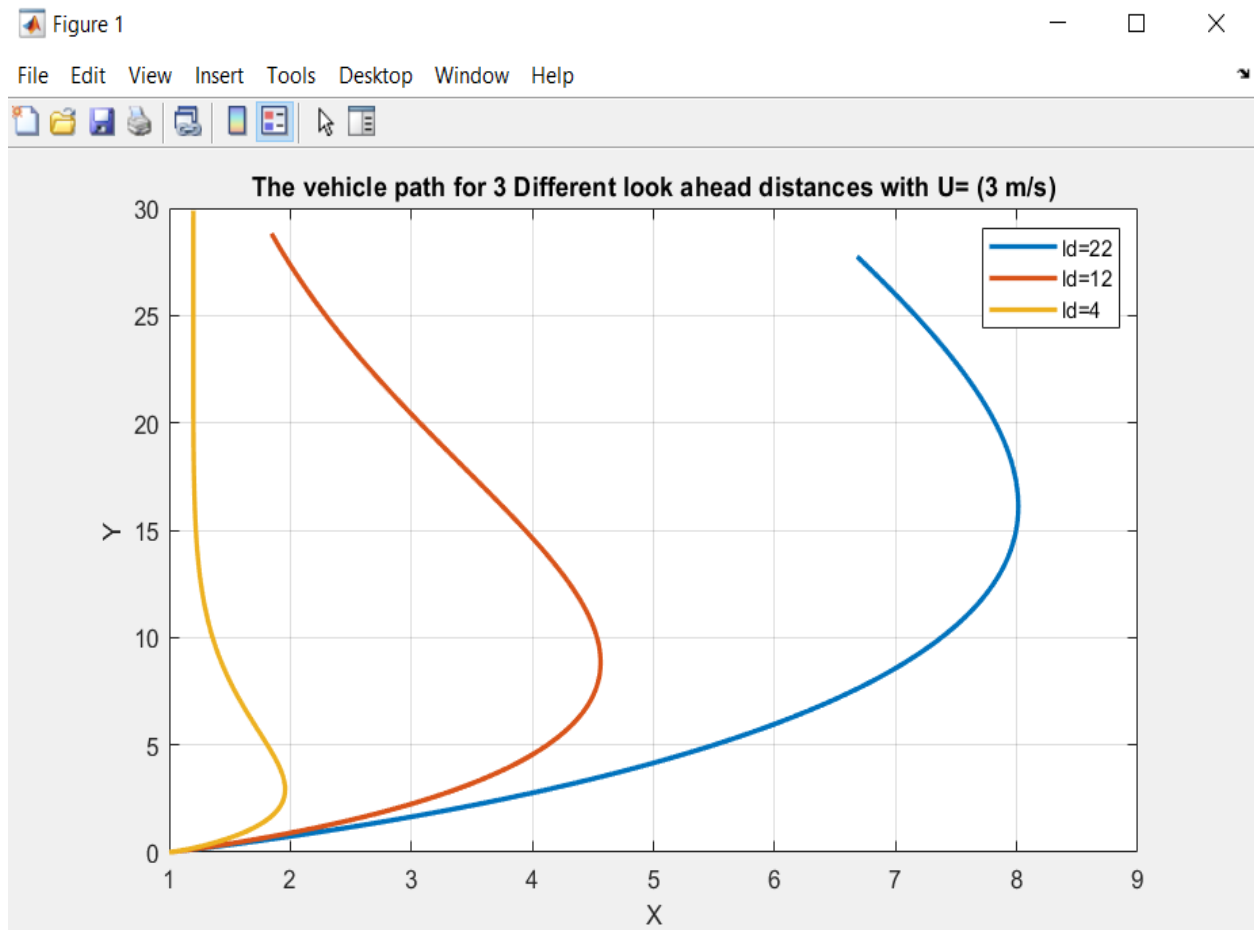


Figure 3.23: The Vehicle Path for 3 Different Look Ahead Distance with ($U=3\text{m/s}$)

When the longitudinal velocity is equal (U=8m/s):

%The parameters of the vehicle

```
clear all
mass=1250;
Iz=1550;
L=2.2;
ad=1;
bd=1.2;
Cfront=50000;
Crear=50000;
U=8;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

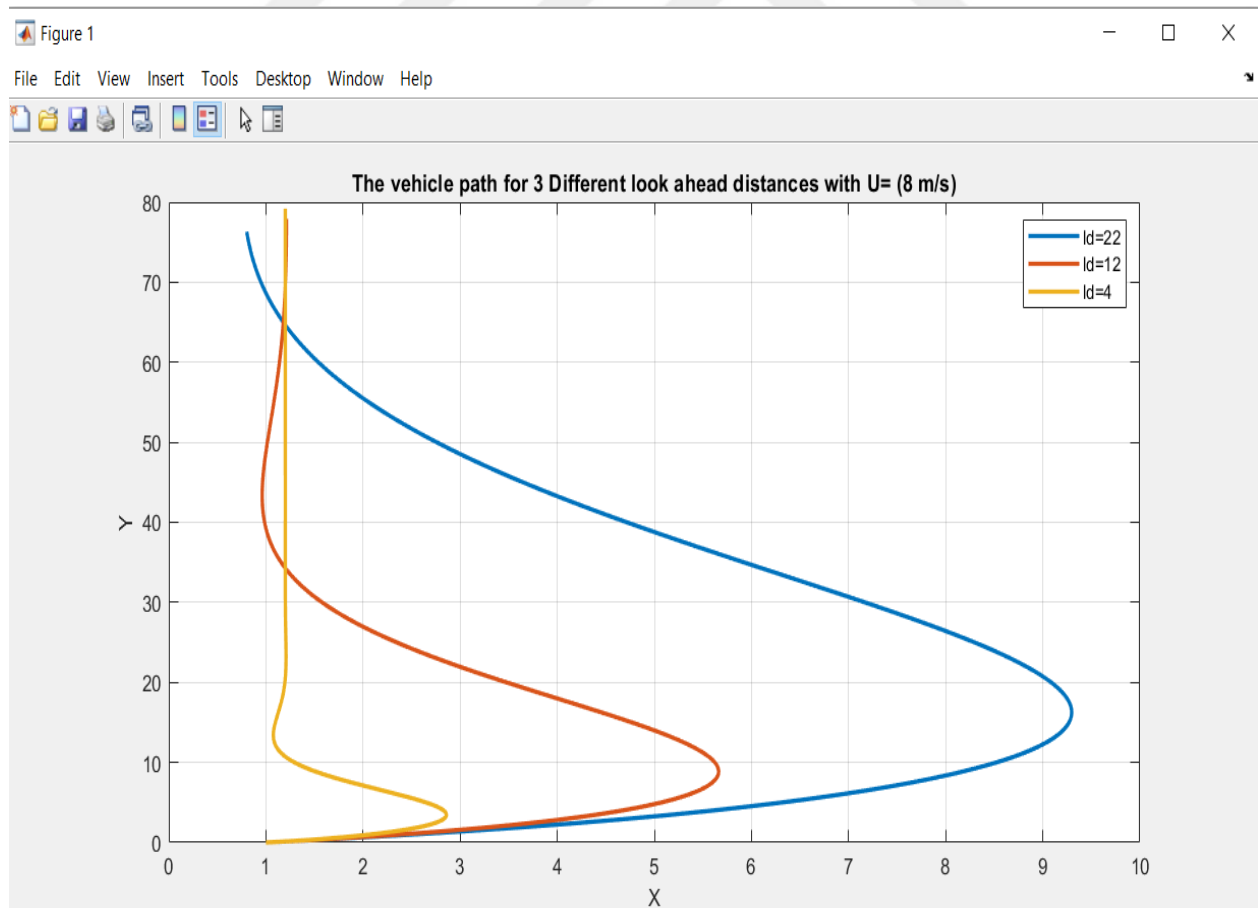


Figure 3.24: The Vehicle Path for 3 Different Look Ahead Distance with (U=8m/s)

When the longitudinal velocity is equal ($U=13\text{m/s}$):

%The parameters of the vehicle

```
clear all
mass=1250;
Iz=1550;
L=2.2;
ad=1;
bd=1.2;
Cfront=50000;
Crear=50000;
U=13;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

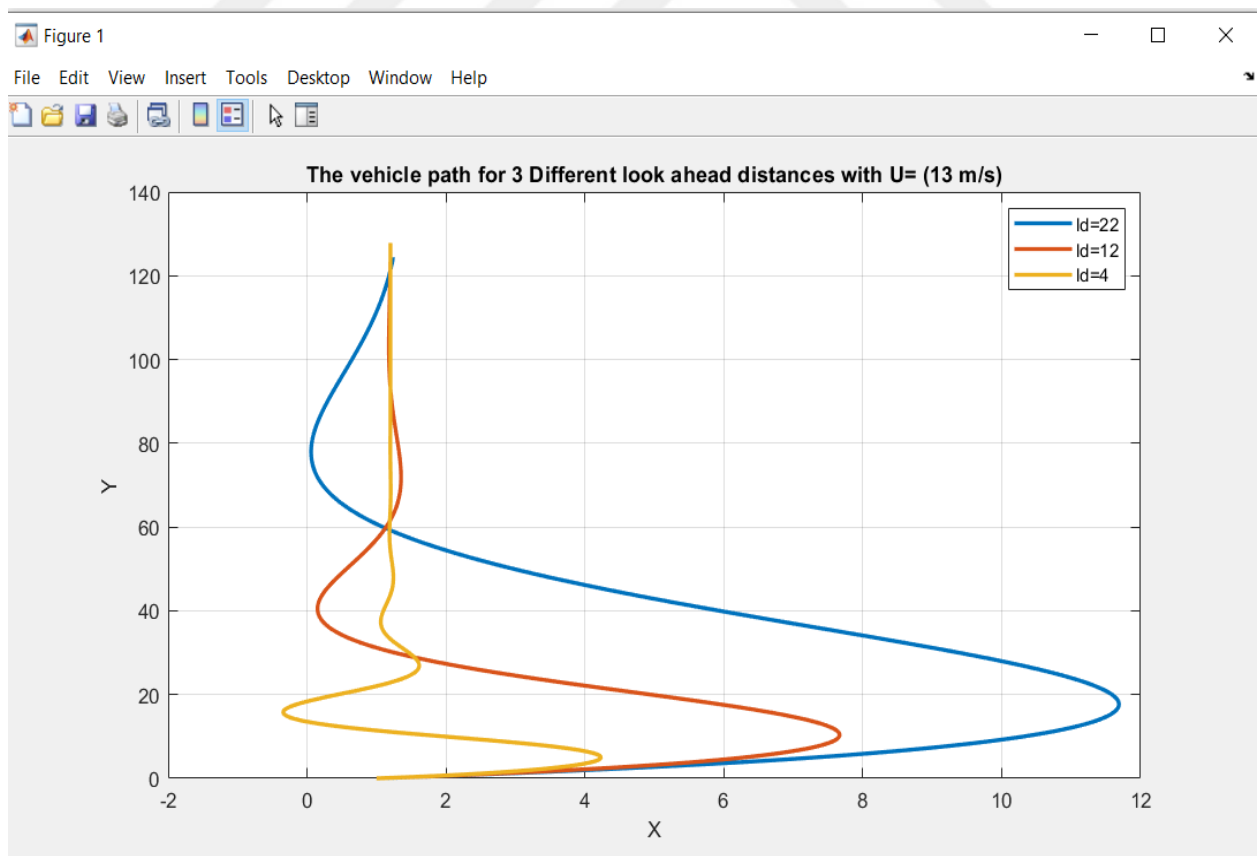


Figure 3.25: The Vehicle Path for 3 Different Look Ahead Distance with ($U=13\text{m/s}$)

When the speed became over than 15 m/s the controller will crash. This is because the controller is affected by high speed and while working at high speed, a control the crash occurs due to increased instability and overshoots for the system. The longitudinal velocity ($U=25$) was assumed and the figures shown below explain that.

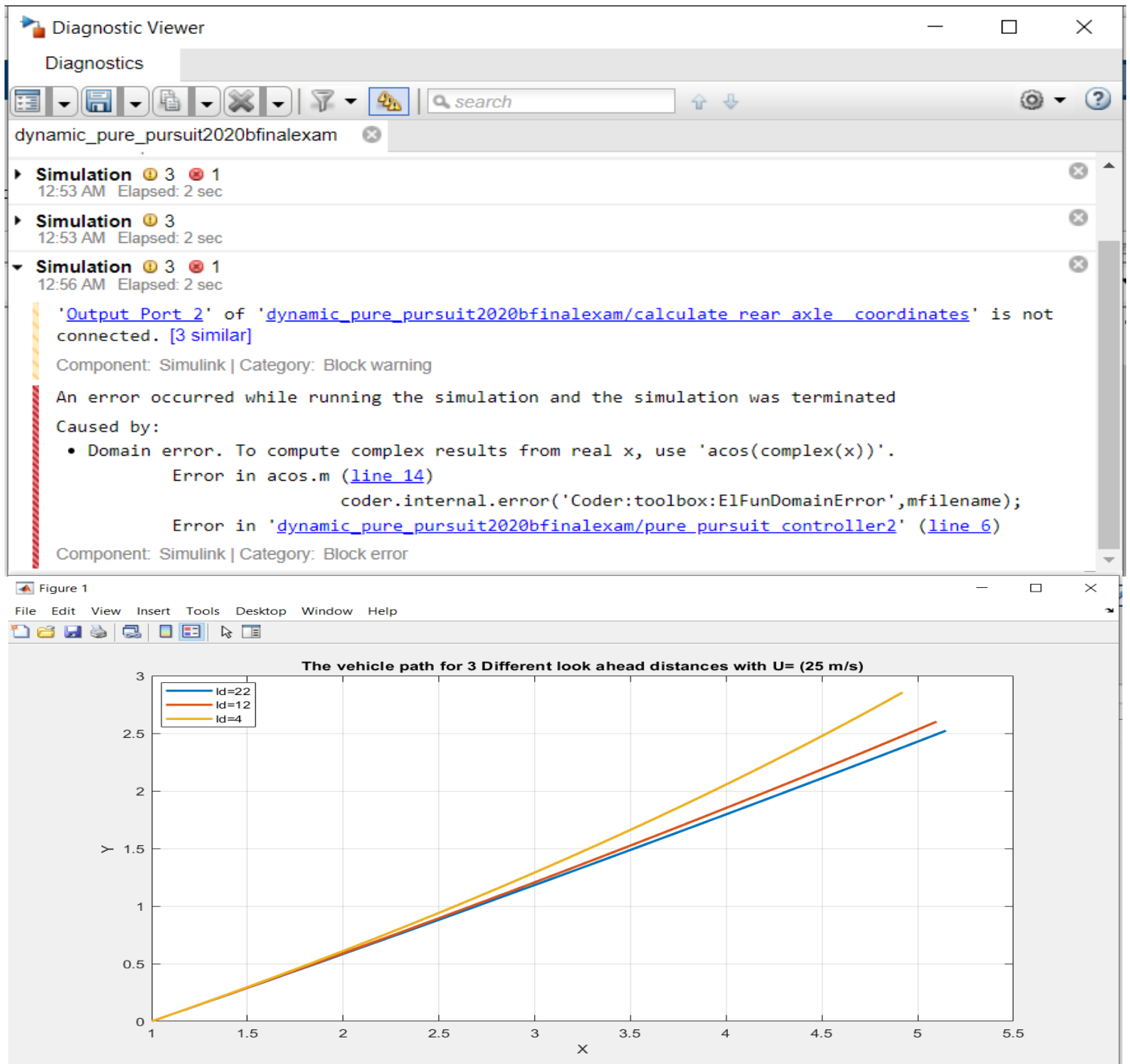


Figure 3.26: The Vehicle Path for 3 Different Look Ahead Distance with ($U=25\text{m/s}$) When the System Crashed

Discuss the results when speed of the vehicle has been changed (3, 8, 13 and 25):

In this test, the vehicle speed was changed to three different values, as when the speed decreases, it greatly affects the stability of the vehicle and actually negatively affects the increase in instability, and this is what Figure 10 shows, where three different values of L_d were shown, and we can see the same previous results. Which was talked about in the first section. When you increase the speed of the control as Figures 11 and 12 shown , this makes the system stable and reduces the overshoot of the system and it make the system following the Y-axis smoothly , but it's not better than the lower speed also because we can see the instability in high speed in another hand if we increase the speed and getting bigger and bigger that's will affect directly in the controller and it will be not valid anymore and this happens when the speed becomes over 15 in this controller in another meaning the controller will crashed according the high speed and the Figure 13,14 shows that .Therefore, the speed is one of the important factors affecting the stability of the vehicle and the controller, and there are limits to the maximum speed that the system reaches the state of crash, and for this, the appropriate speed must be chosen, which is commensurate with the L_d to obtain the least effect and the best stability and follow the system well and to make the response smooth.

3.6.2. Changing Longitudinal velocity for pure pursuit model when the vehicle following X axis

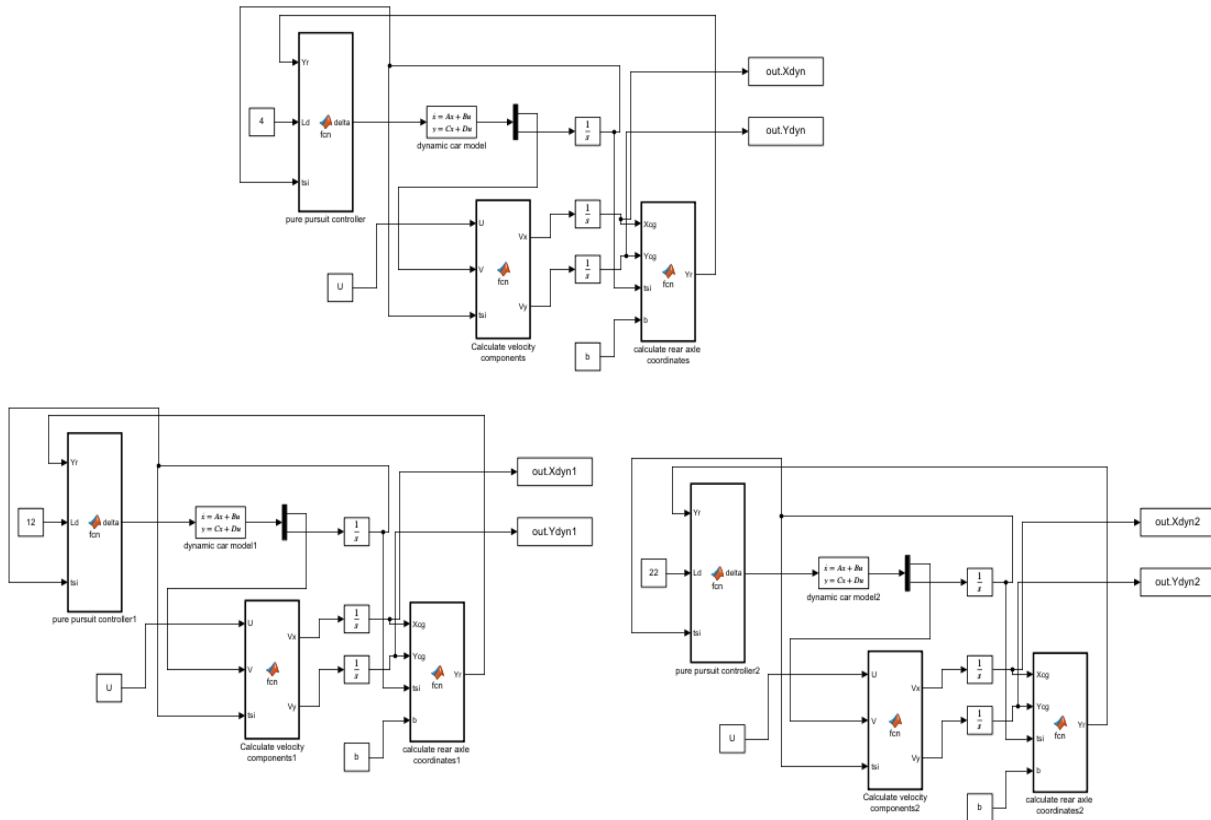


Figure 3.27: Stimulant for the Vehicle That Follows the X-Axis with 3 Different Values for L_d

When the longitudinal velocity is equal ($U=3\text{m/s}$) :

%The parameters of the vehicle

```
clear all
mass=1250;
Iz=1550;
ad=1;
bd=1.2;
Cfront=50000;
Crear=50000;
U=3;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

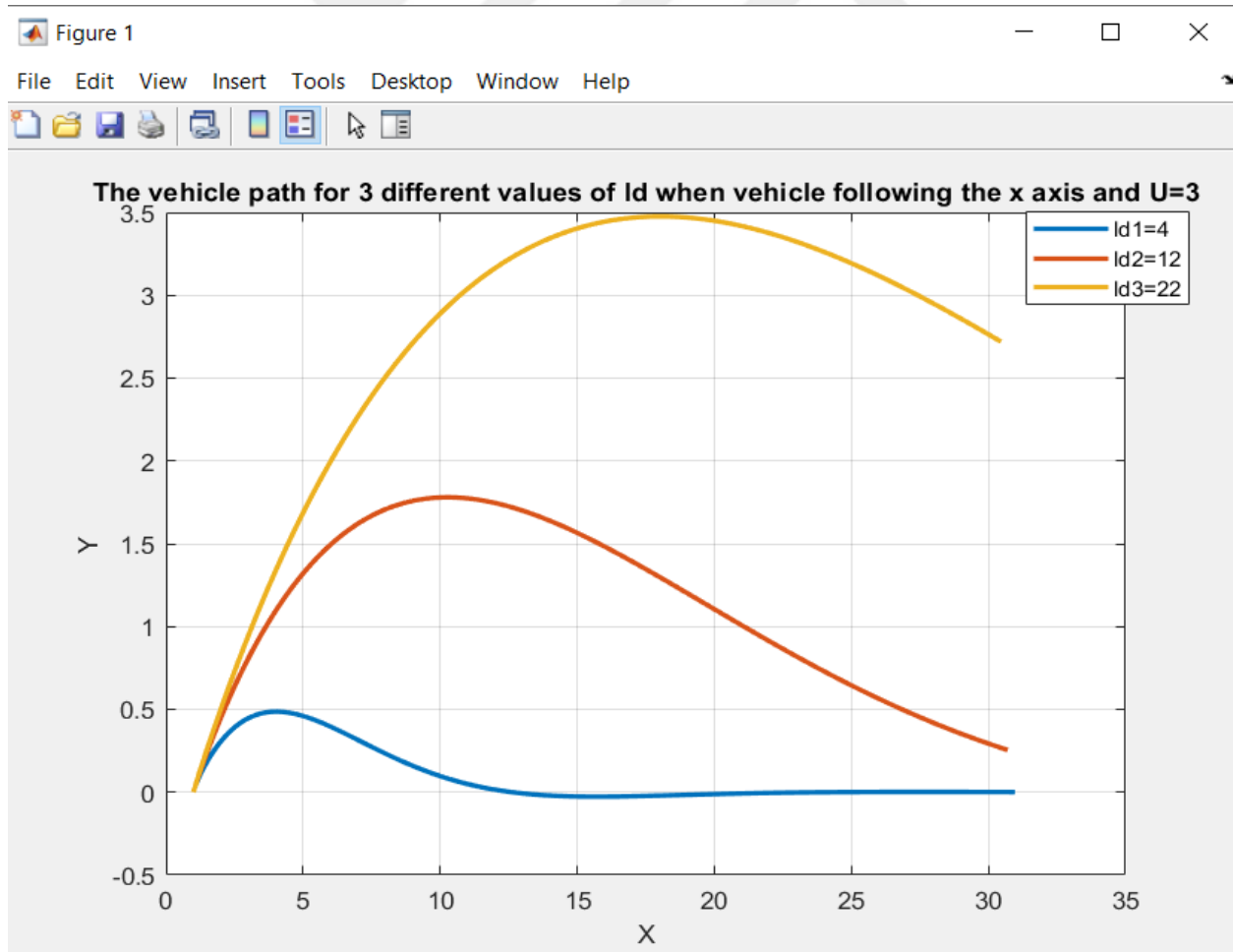


Figure 3.28: The Vehicle Path for 3 Different Look Ahead Distance with ($U=3\text{m/s}$)

When the longitudinal velocity is equal ($U=8\text{m/s}$):

%The parameters of the vehicle

```
clear all
```

```
mass=1250;
```

```
Iz=1550;
```

```
ad=1;
```

```
bd=1.2;
```

```
Cfront=50000;
```

```
Crear=50000;
```

```
U=8;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
```

```
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

Figure 1

File Edit View Insert Tools Desktop Window Help

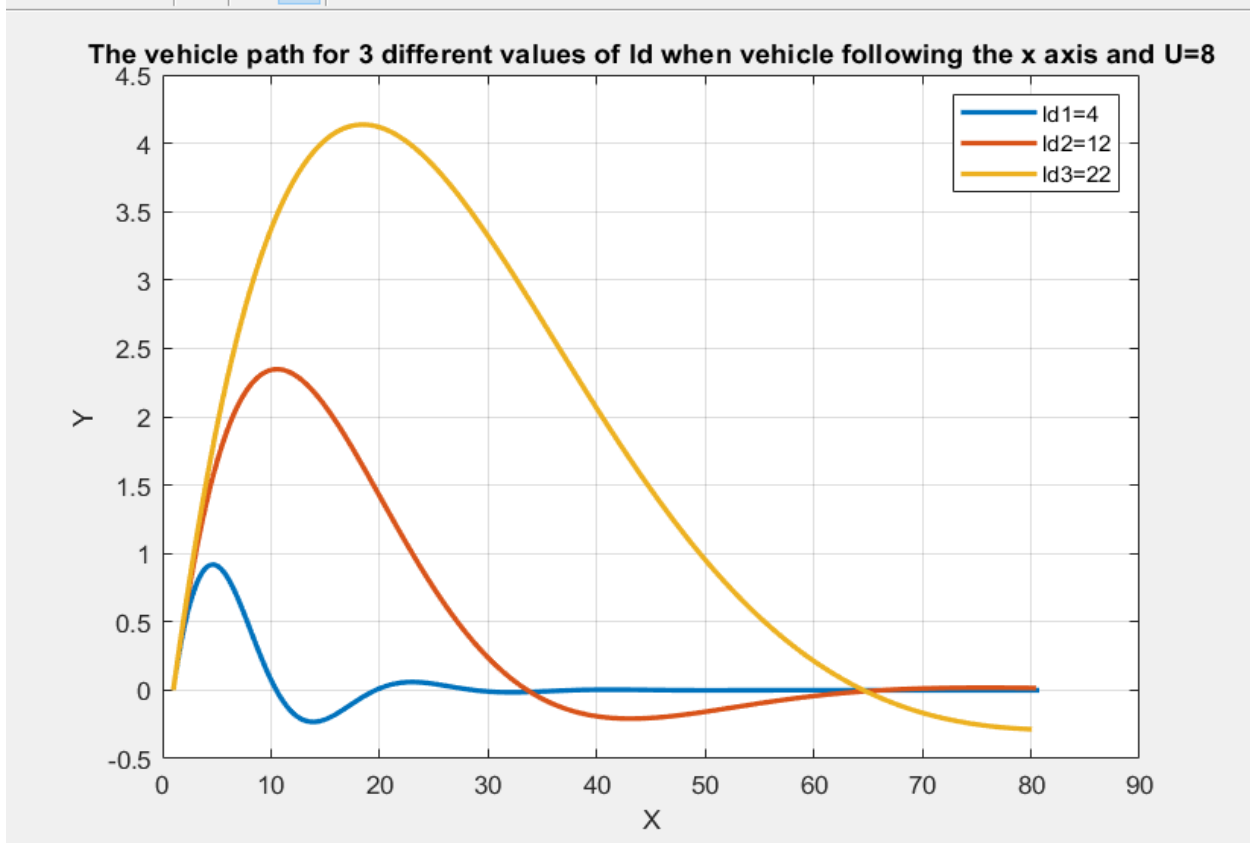


Figure 3.29: The Vehicle Path for 3 Different Look Ahead Distance with ($U=8\text{m/s}$)

When the longitudinal velocity is equal ($U=13\text{m/s}$):

%The parameters of the vehicle

```
clear all
```

```
mass=1250;
```

```
Iz=1550;
```

```
ad=1;
```

```
bd=1.2;
```

```
Cfront=50000;
```

```
Crear=50000;
```

```
U=13;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
```

```
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

Figure 1

File Edit View Insert Tools Desktop Window Help



The vehicle path for 3 different values of l_d when vehicle following the x axis and $U=13$

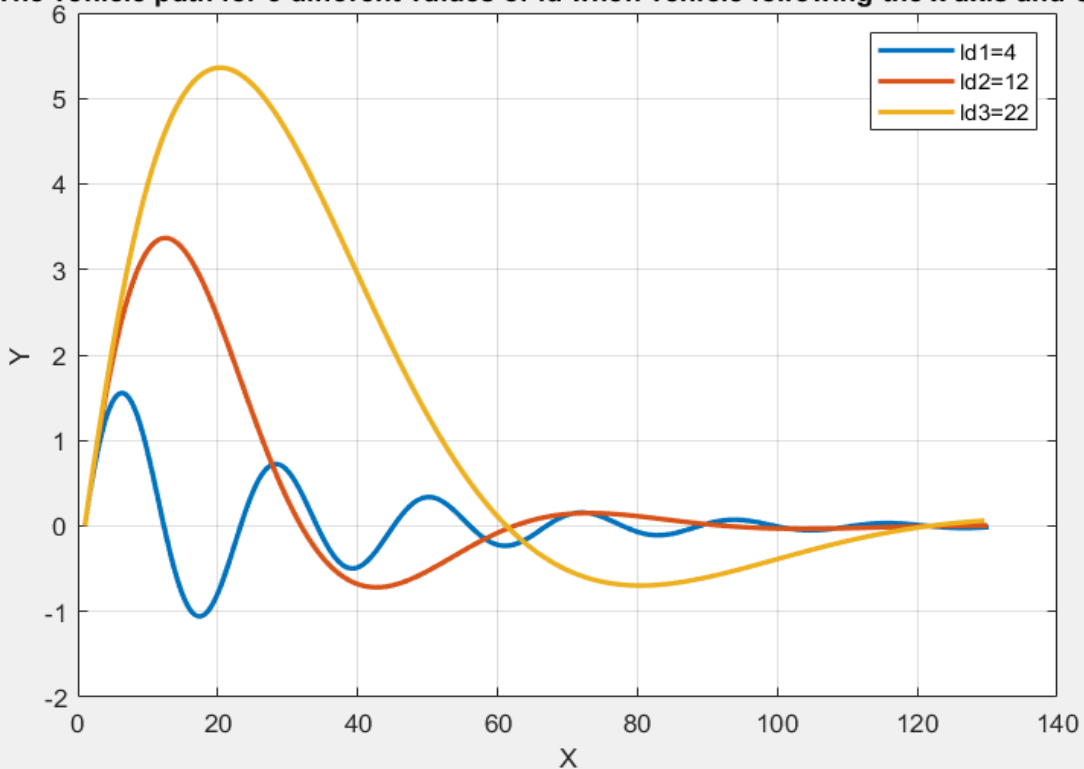


Figure 3.30: The Vehicle Path for 3 Different Look Ahead Distance with ($U=13\text{m/s}$)

3.7. STANLEY METHOD

Before getting started with the Stanley controller, it helps to know an easier way. Stanley is an improvement in this approach, turning a pure pursuit into a more intuitive control law. In contrast to the pure tracking method, where the rear axle is the reference point, the Stanley method uses the front axle as the reference point. Observe course errors and crossing errors at the same time. In this method, the cross path error is defined as the distance between the closest point on the path and the front axle of the vehicle.

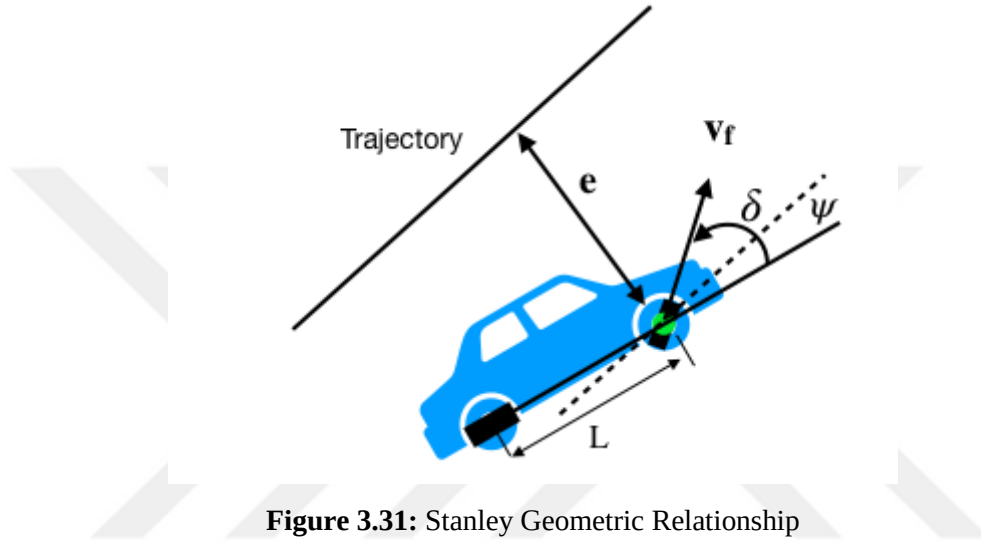


Figure 3.31: Stanley Geometric Relationship

As shown in Figure 3.31, ψ is the angle between the lane direction and the vehicle direction. The steering angle is expressed as δ . Stanley's method has three intuitive spin laws: [7]

1. Eliminating the heading errors. $\delta(t) = \psi(t) = \theta_e(t)$
2. Elimination of crossing errors. This step consists of finding the closest point between the route and the vehicle, denoted as $e_{fa}(t)$. The steering angle can be corrected as follows:

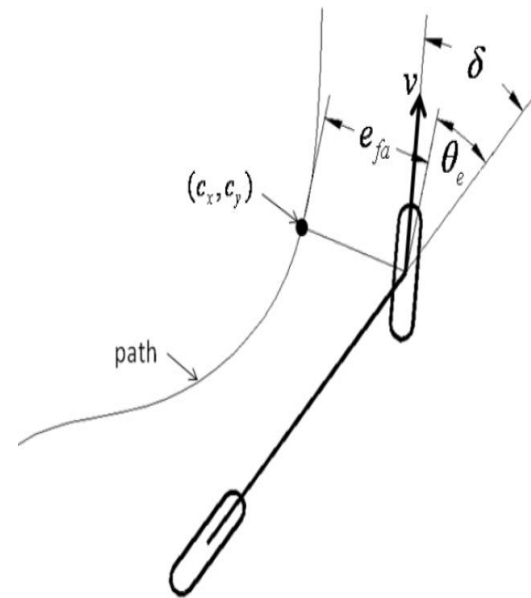
$$\delta(t) = \tan^{-1} \left(\frac{ke_{fa}(t)}{v_x(t)} \right)$$

3. The steering control method that the max steering angle bounds. That means $\delta(t) \in [\delta_{min}, \delta_{max}]$

$$\delta(t) = \theta_e(t) + \tan^{-1} \left(\frac{ke_{fa}(t)}{v_x(t)} \right) \quad \delta(t) \in [\delta_{min}, \delta_{max}]$$

Noting that the Heading error is equal:

$$\theta_e = \theta - \theta_p$$



Where $\theta = \text{yaw}$ and $\theta = \text{the heading of the curve}$. Instead of calculating course error in terms of guidance distance, we calculate course error in terms of trajectory curvature. The approximate curvature of the trajectory can be calculated as follows:

$$\tan(\text{last point.}y - \text{first point.}y, \text{last point.}x - \text{first point.}x)$$

Why is the Stanley controller effective and stable?

If the course error is large and the cross-track error is small, it means that ψ is large, so the steering angle δ also increases and turns in the opposite direction to correct the course error so that the vehicle is in the same direction as the trajectory.

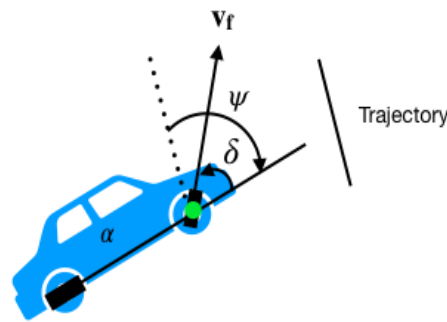


Figure 3.32: Large Heading Error (Course Error)

The process of this scenario can be drawn as follows:

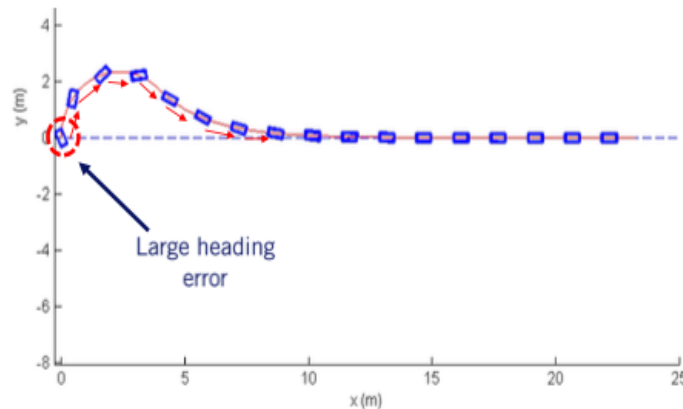


Figure 3.33: Large Heading Error Correction Process

If the course offset error is large with a small heading error, it can do the following:

$$\tan^{-1}\left(\frac{ke_{fa}(t)}{v_x(t)}\right) \approx \frac{\pi}{2}$$

So the steering angle will be:

$$\delta(t) = \psi(t) + \frac{\pi}{2}$$

Assuming that course error $(t) = 0$, then $(t) \pi/2$. This causes the vehicle to enter the track as follows:

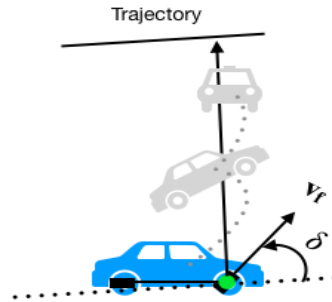


Figure 3.34: Large Course Offset, Turn Onto Track First

If steering changes course, course correction overrides lateral toe correction and resets steering to zero. As the vehicle approaches the road, the course offset decreases and the steering angle begins to correct course alignment as follows:

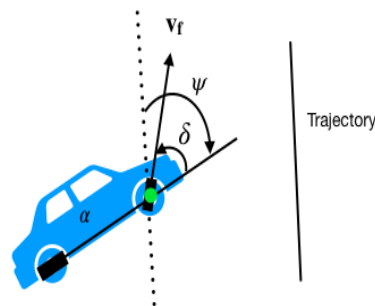


Figure 3.35: As The Cross-Track Error Decreases, The Direction Aligns The Route.

The process of this scenario can be drawn as follows:

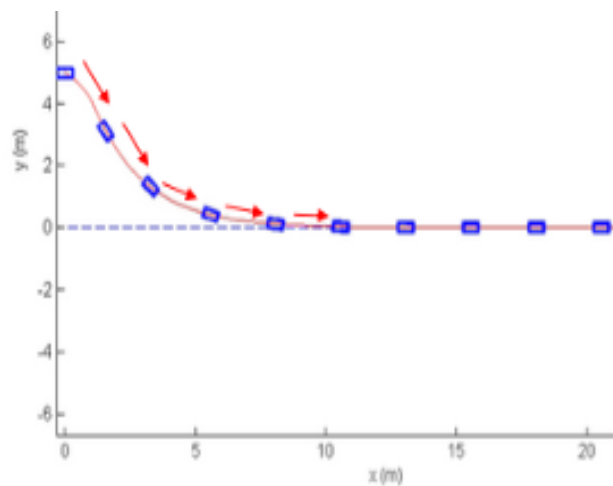


Figure 3.36: The Large Cross-Track Error Correction Process

The Stanley controller is simple but it's an effective and constant method for subsequent control. This method was created using MATLAB, as shown in the following figure, using all of the previous conditions

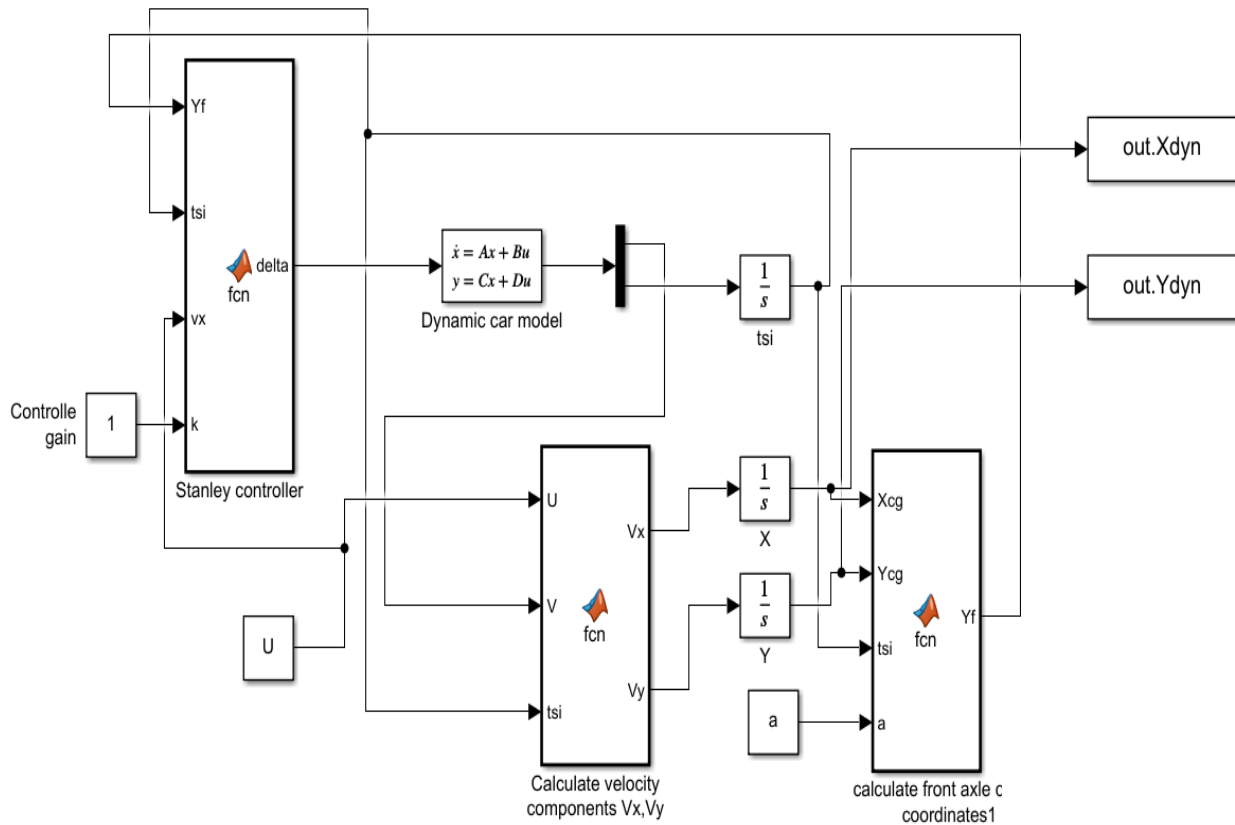


Figure 3.37: Stanley Controller Stimulant

Finding the coordinates of front axle center when the vehicle following the X-axis:

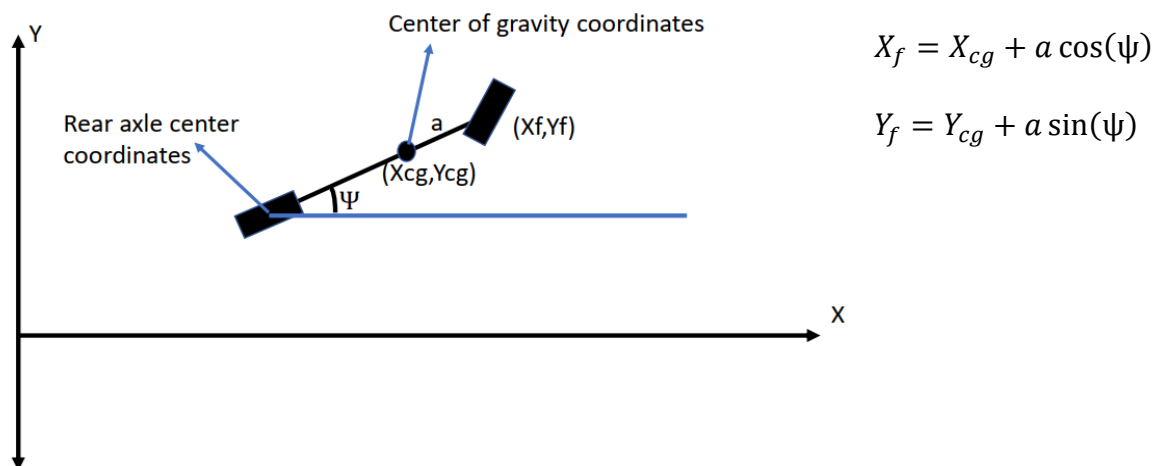


Figure 3.38: Finding the Coordinates of Front Axle Center When the Vehicle Following the X-Axis

Since the Stanley method uses the front axle center coordinates in the formula, we must calculate these coordinates based on the center of gravity coordinates used in the dynamic model:

The MATLAB function code to finding front axle center coordinates:

```
function Yf = fcn(Xcg,Ycg,tsi,ad)
Xf=Xcg+ad*cos(tsi);
Yf=Ycg+ad*sin(tsi);
```

The MATLAB function code for Stanley Controller:

```
function delta= fcn(Yf,tsi,vx,k)
%steering error
thetap=0;
theta=tsi;
thetae=theta-thetap; %heading error
Cp1=thetae;
%front axle lateral path error calculation
efa=Yf;
Cp2=atan(k*efa/vx);
delta=-(Cp1+Cp2);
```

The last MATLAB function in the Stimulant to calculate the velocity coordinates and the code shown below explain this:

```
function [Vx,Vy]= fcn(U,V,tsi)
Vx=U*cos(tsi)-V*sin(tsi);
Vy=U*sin(tsi)+V*cos(tsi);
```

This model was applied to three different speeds in order to make a comparison between it, and the pure pursuit model where it was found that the results are very close. And this is what the drawings show below. Where the calculation was made for the same initial conditions, and for the same parameters of the car. And it was put in the conclusions section of the consequences of the use of this method in tracking path.

The parameter code that was used when $U=3$ m/s:

```
%The parameters of the vehicle
```

```
clear all
```

```
mass=1250;
```

```
Iz=1550;
```

```
ad=1;
```

```
bd=1.2;
```

```
Cfront=50000;
```

```
Crear=50000;
```

```
U=3;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
```

```
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

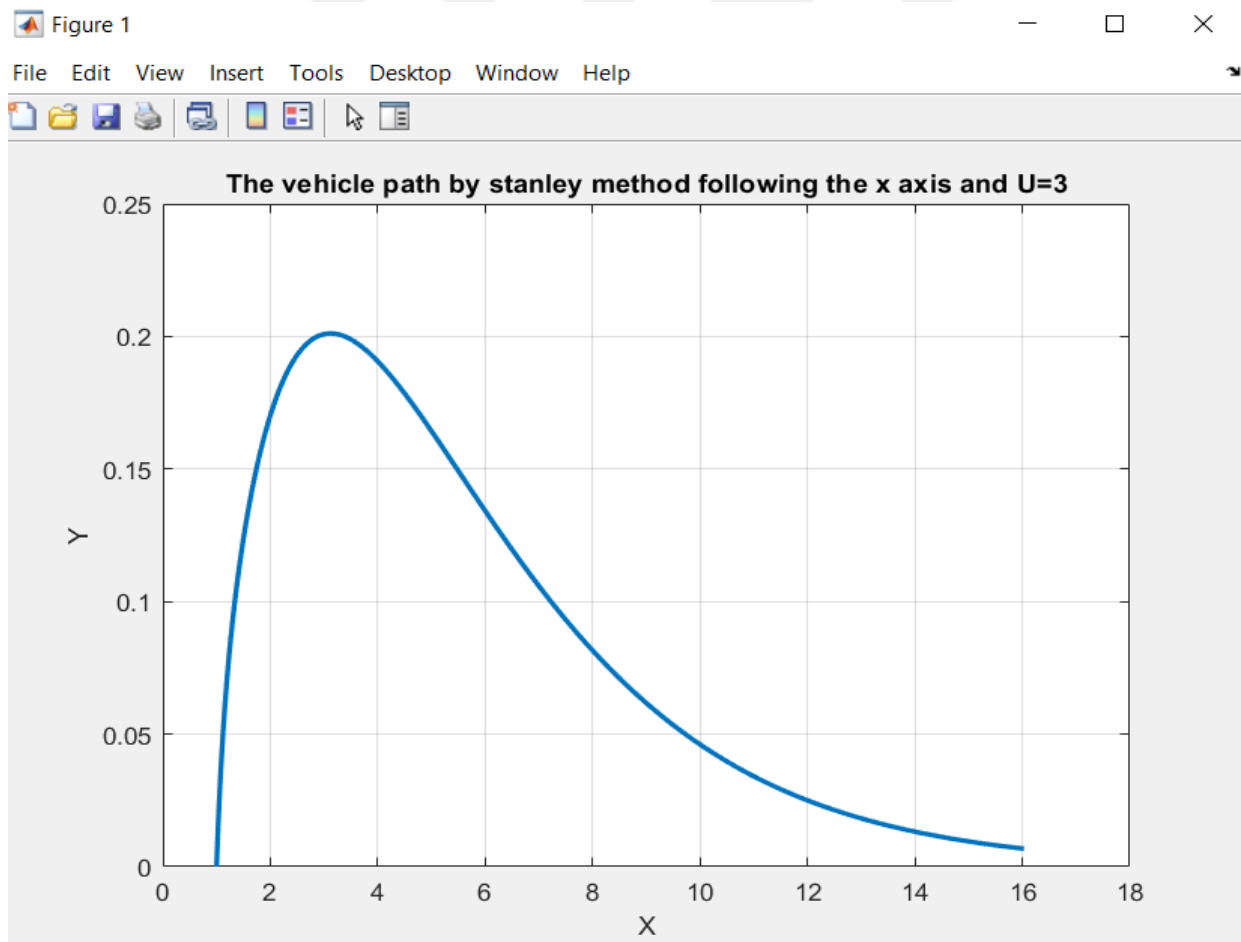


Figure 3.39: Stanley Method When $U = 3$ m/s

The parameter code that was used when $U=8$ m/s:

```
%The parameters of the vehicle
```

```
clear all
```

```
mass=1250;
```

```
Iz=1550;
```

```
ad=1;
```

```
bd=1.2;
```

```
Cfront=50000;
```

```
Crear=50000;
```

```
U=8;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
```

```
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

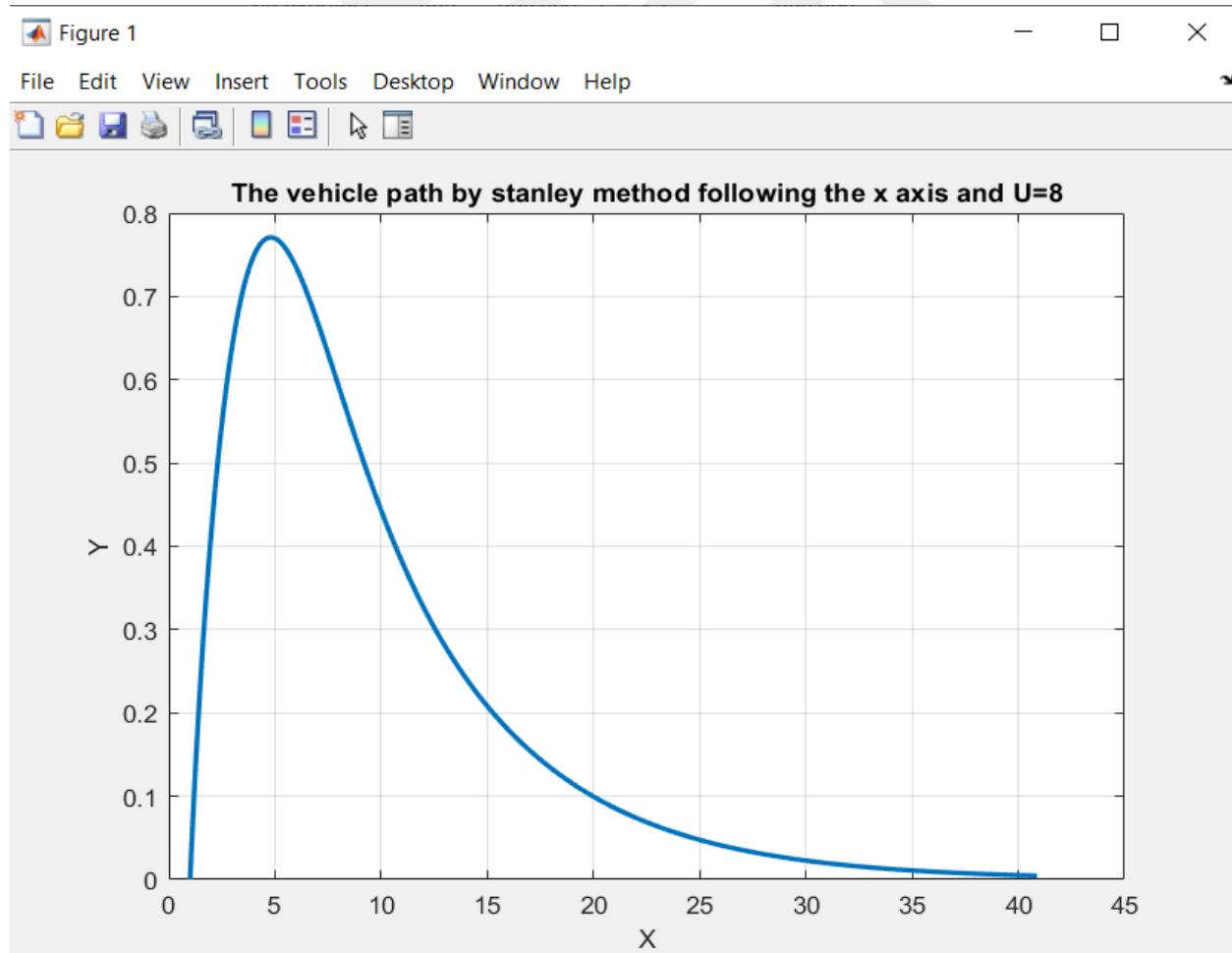


Figure 3.40: Stanley Method When $U = 8$ m/s

The parameter code that was used when U=13 m/s:

`%The parameters of the vehicle`

```
clear all
```

```
mass=1250;
```

```
Iz=1550;
```

```
ad=1;
```

```
bd=1.2;
```

```
Cfront=50000;
```

```
Crear=50000;
```

```
U=13;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
```

```
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

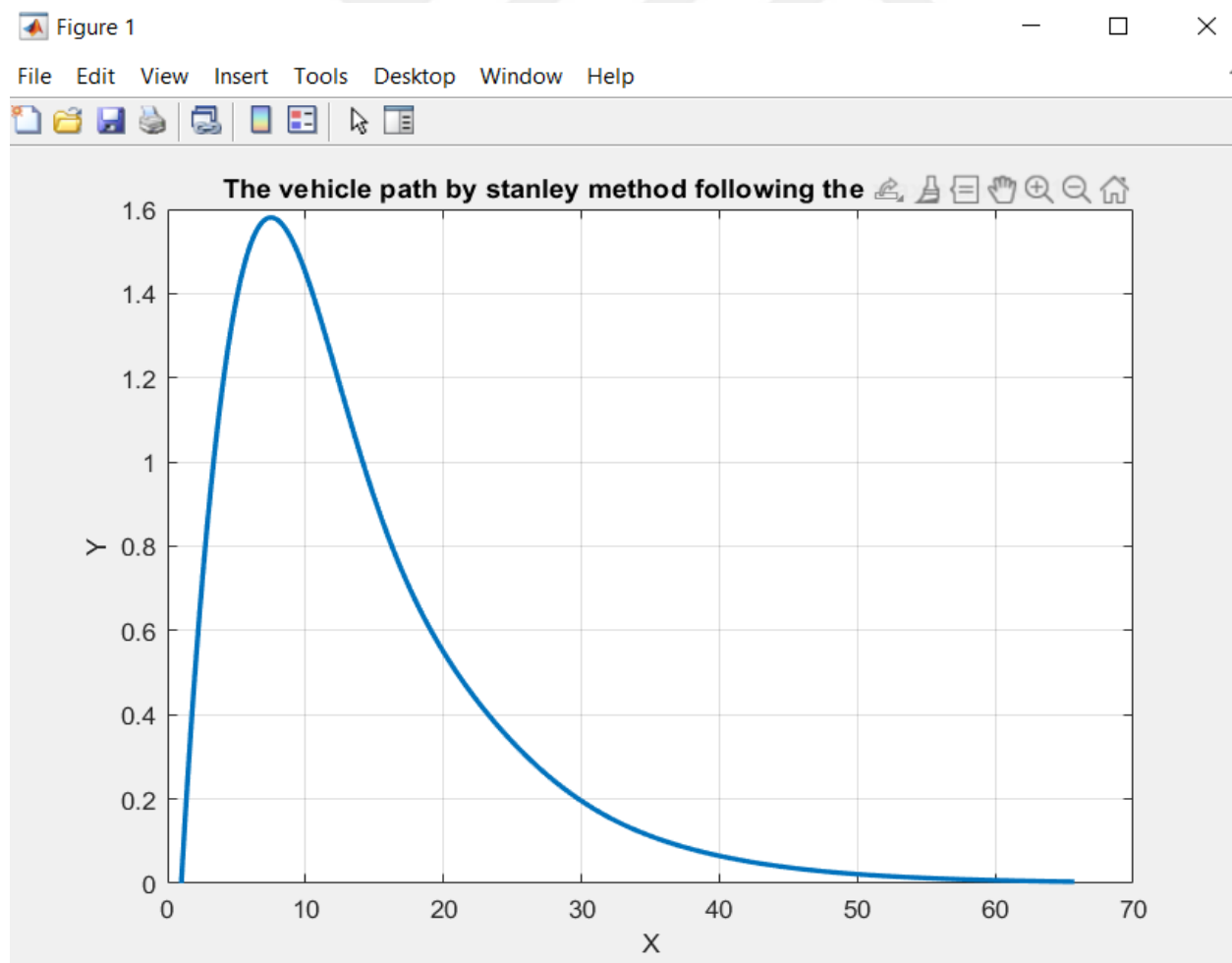


Figure 3.41: Stanley Method When U =13 m/s

3.8. CALCULATE THE LOOK-AHEAD DISTANCE AUTOMATICALLY

The method that was used to calculate the look-ahead distance automatically by creating a driving scenario through the Driving Designer in MATLAB and using a code and running it after that the results could be taken by using Simulink .Where 2 driving scenarios were made according to the curvature of the drawn trajectory, as the radius of the turn differs from the curvature. The figure below illustrates the method used during this section. The results and the procedure followed are also explained in detail.

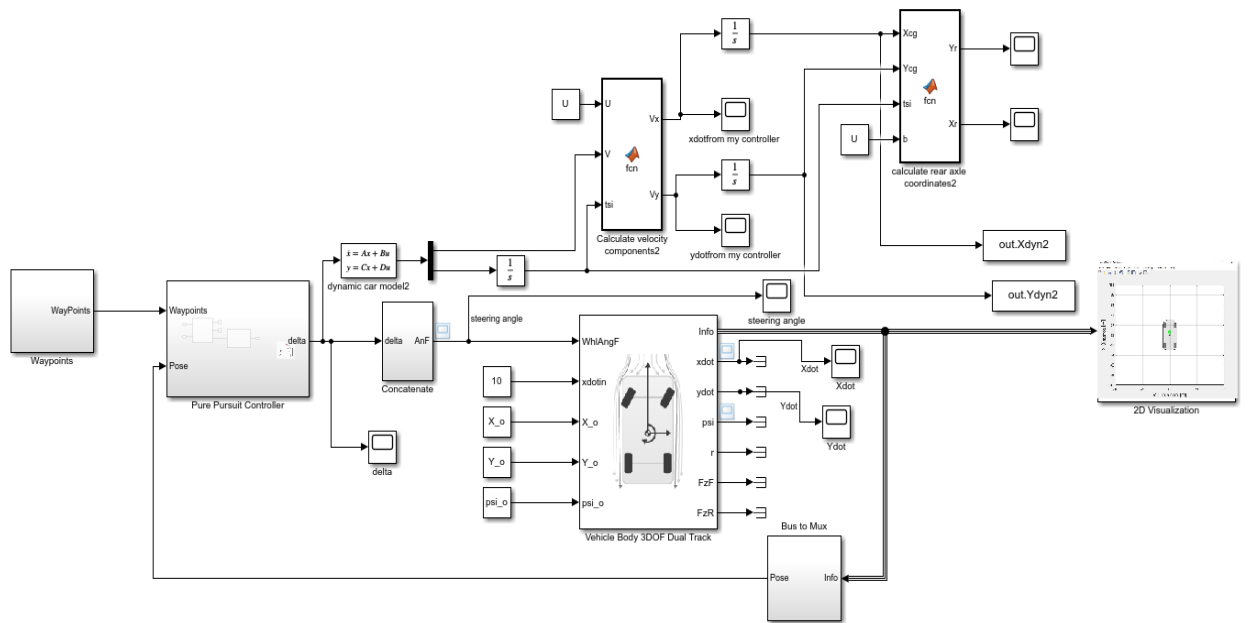


Figure 3.42: Pure Pursuit Controller for an Autonomous Vehicle with Two Different Driving Scenario

The subsystem implements a Pure Pursuit controller. The "Pure Pursuit" block calculates the target direction and the "Wheel Steering Angle Formulation" block calculates the wheel steering angle that's shown in figure below.

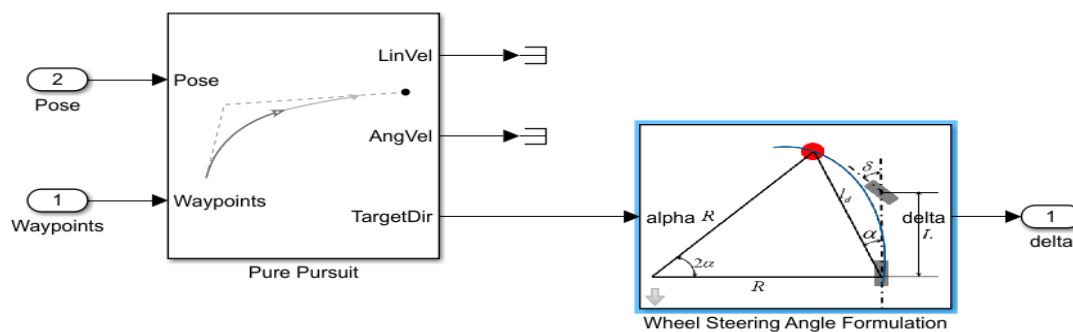


Figure 3.43: Pure Pursuit Controller and the "Wheel Steering Angle Formulation"

The results of this Simulink find out the current location of the vehicle. And that offers functions that report the current position of the vehicle as (x, y, heading). The position is reported relative to the vehicle's position at the time of initialization. This initial position is the general frame of reference for the path. Moreover to find the closest waypoint to the vehicle. In the geometric derivation it was stated that the target point would be a look ahead distance in front of the vehicle. There may be multiple points within a look ahead distance from the current vehicle. Location. The vehicle should be steered to the nearest point within forward gaze distance of your current location. Therefore, first the closest waypoint to the vehicle is found and a point (g_x, g_y) at look ahead distance is sought. Removal from the vehicle begins at this point and begins to climb the route furthermore, find the target point. The waypoint is found by moving up the route and calculating the distance between that waypoint and the vehicle's current position. Waypoint locations are recorded on a global scale; this calculation is performed on the coordinates. Convert the target point to vehicle coordinates. Once the target point has been found, it must be transformed into the local coordinates of the vehicle even more, the geometric derivation for the curvature was carried out in the vehicle coordinates and the curvature commands for the vehicle in the vehicle coordinates. Calculate the curvature. The curvature equation derived from (Theoretical Derivation) calculates the desired curvature of the vehicle. The camber is converted into the steering wheel angle by the vehicle's onboard control unit. Additionally, update the location of the vehicle. During the simulation, it must be determined what effects the command has on the position and direction of the vehicle.

The driving scenario must add to the parameter file after that the Simulation would applied:

```
mass=1250; %mass of vehicle
Iz=1550;
L=2.2;
ad=1;
bd=1.2;
Cfront=50000;
Crear=50000;
U=42;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
%% add the Image to the path in the matlab simulink
addpath(genpath('Images'));
%% load the scen data file that generated from Driving Scenario
%% we can use any scenario to be applayied in simulink
load('saifscenariotwo');
%% defining the reference points:
refPose = data.ActorSpecifications.Waypoints; %and make it as a
subsystem in the simulink
xRef = refPose(:,1); %reference x
yRef = -refPose(:,2); %reference Y
%% reference time of plot
Tst = 22; % the simulation time
s = size(xRef); % the size of X referance as matrix
tRef = (linspace(0,Tst,s(1)))'; % this time variable is used for
plotting the reference points.
%% define parameters used in the models
L = 2.2; % bicycle length of the vehicle
X_o = refPose(1,1); % initial car position for X
Y_o = -refPose(1,2); % initial car position for Y
psi_o = 0.02; % the initial yaw angle
```

Two driving scenario was created and applied to the controller the first radius ($R=114.3\text{m}$) the curvature is equal ($K= 0.00874890638$)

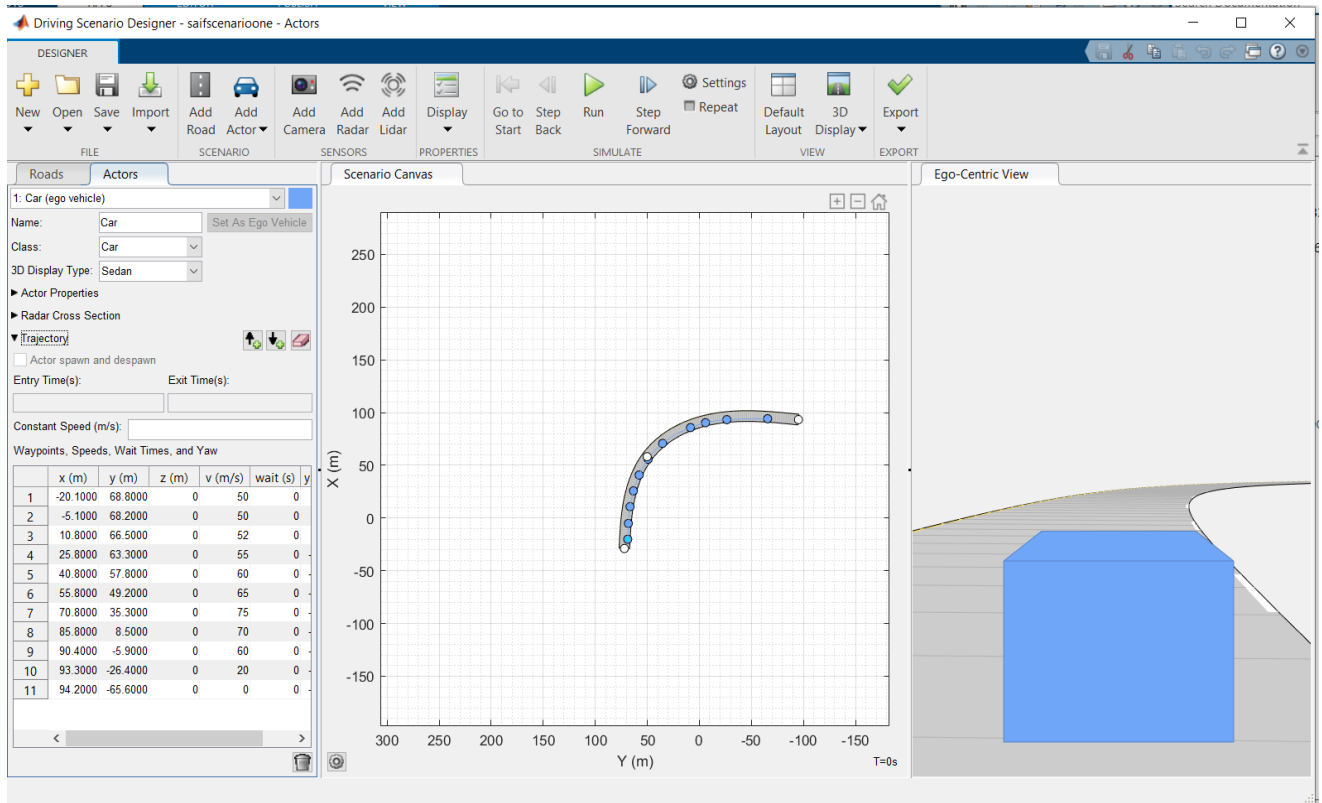


Figure 3.44: Saif scenario 1 for path tracking with radius =114.3m



Figure 3.45: 3D Animation for tracking a path for scenario 1

The results when the look ahead distance is small the response of vehicle is too bad as well as the wheels moving very fast with high response as the figures below shows

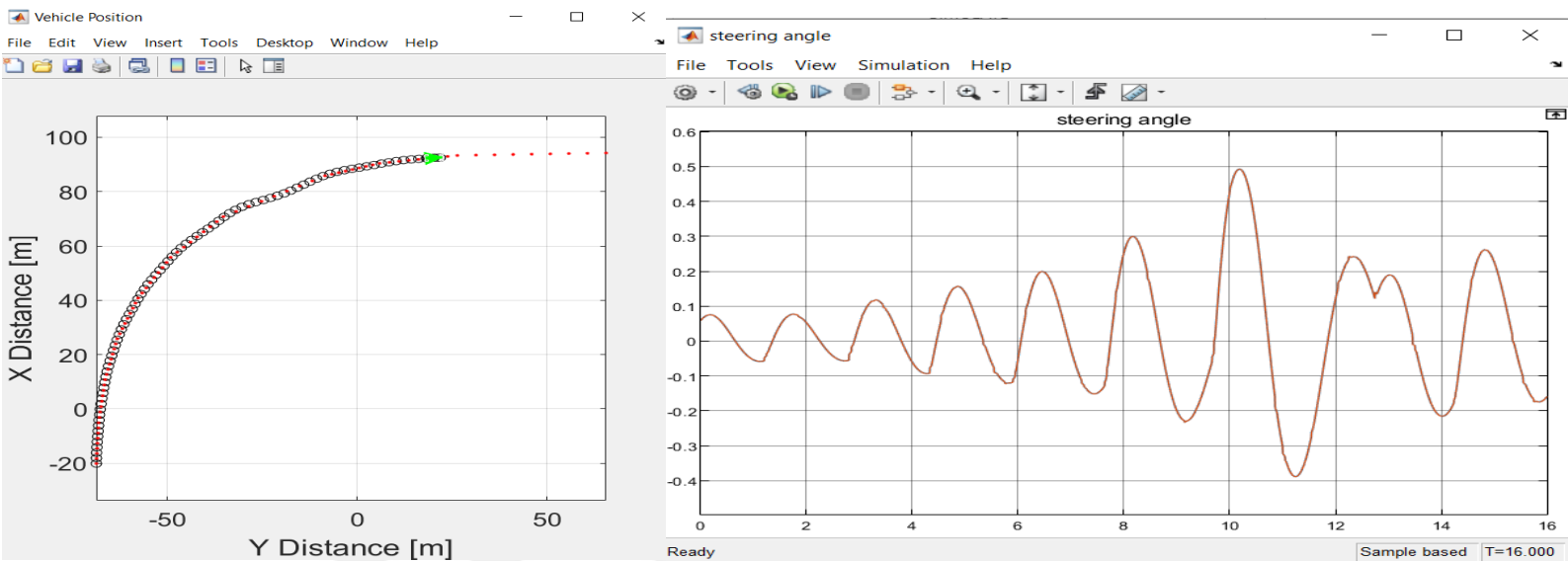


Figure 3.46: Vehicle Position By 2D-Visualiza For Saif Scenario 1with Small Look Ahead Distance And Vehicle Steering Angle (δ)

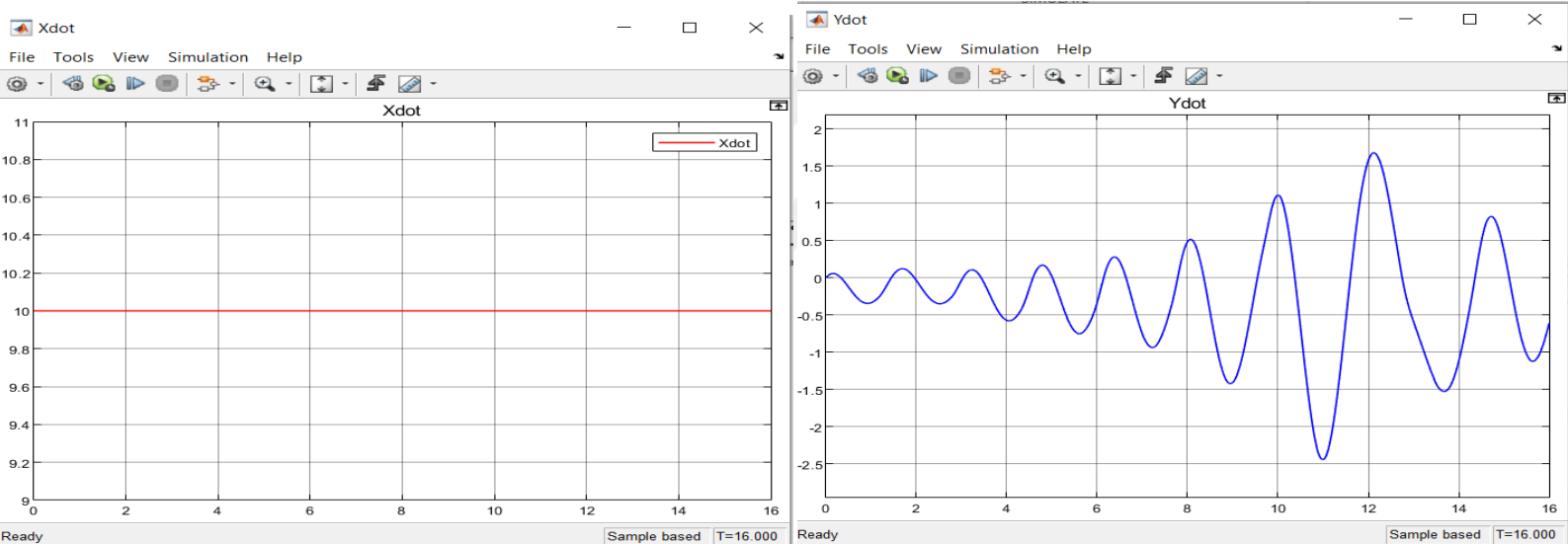


Figure 3.47: Vehicle Velocity At The (X, Y) Direction ($\dot{X}$, $\dot{Y}$) For Saif Scenario 1 For Small Look Ahead Distance
For pure pursuit controller the results of Simulink was:

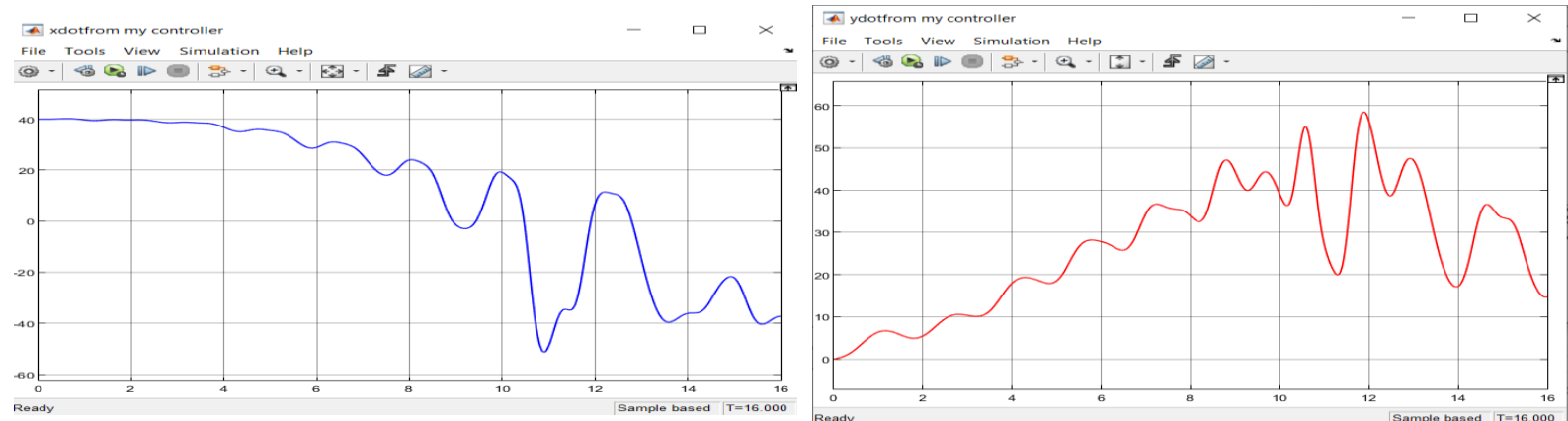


Figure 3.48: Pure Pursuit Controller The Results For Small Look Ahead Distance

The results when the look ahead distance is large the response is good for vehicle wheel as the figures below shows

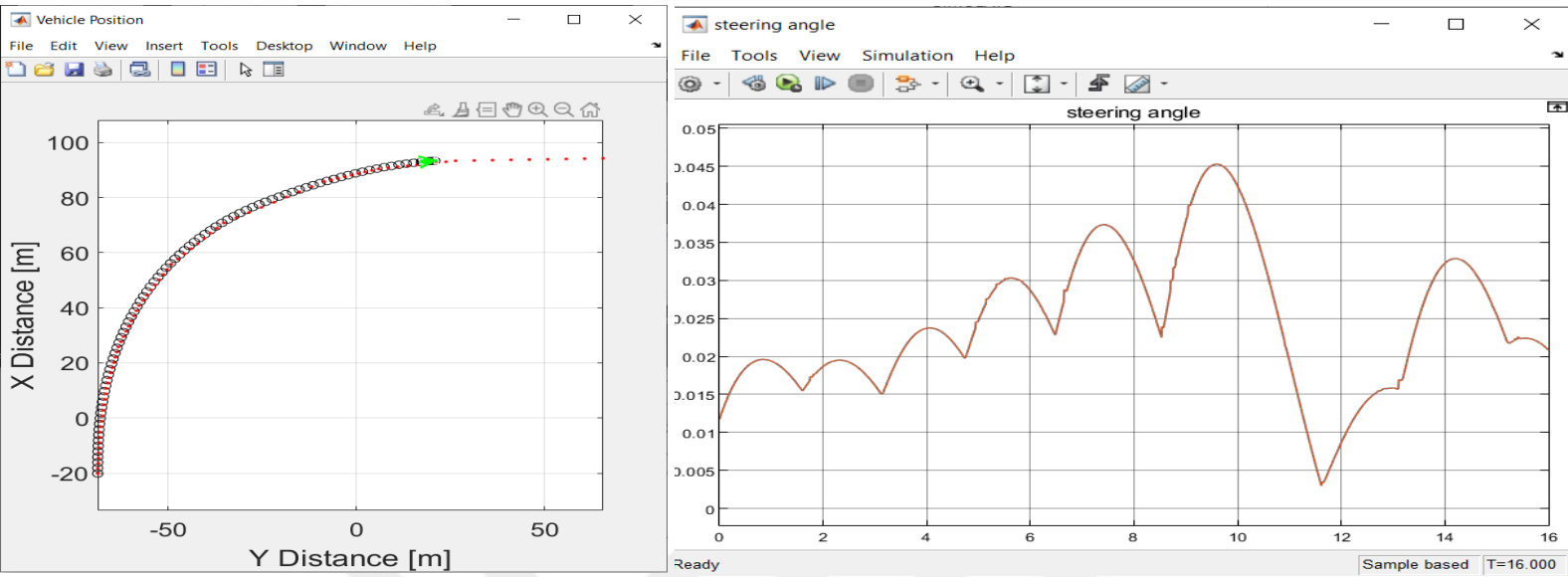


Figure 3.51: Vehicle Position By 2D-Visualiza For Saif Scenario 1with large Look Ahead Distance And Vehicle Steering Angle (δ)

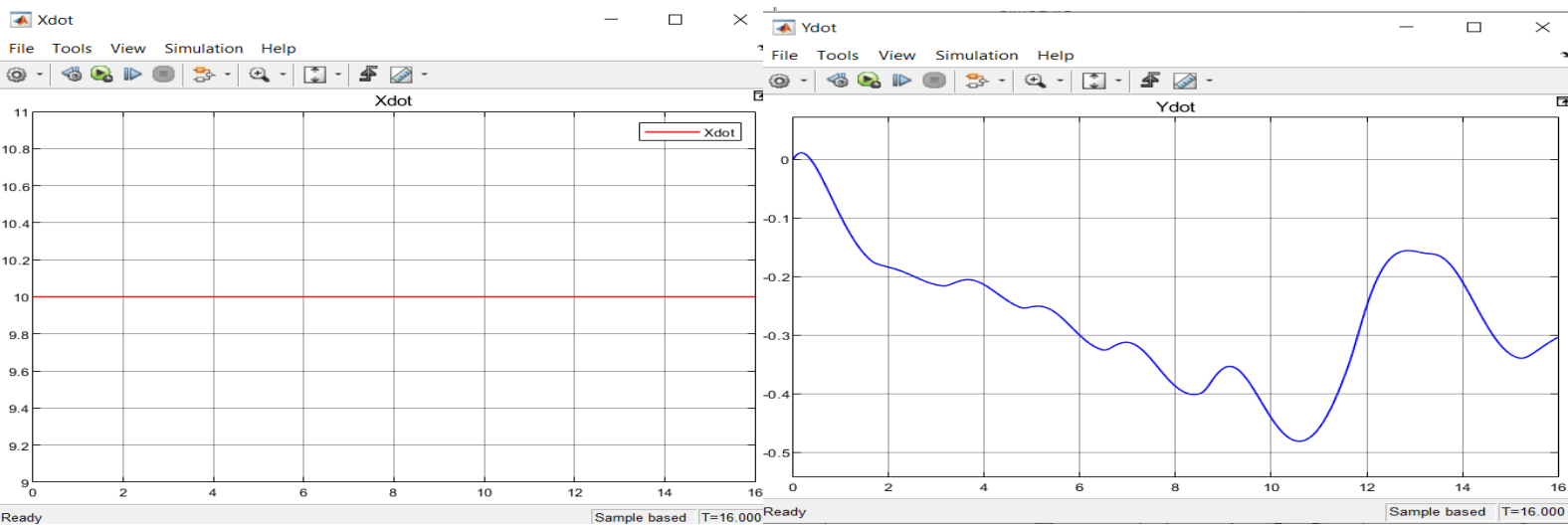


Figure 3.50: Vehicle Velocity At The (X, Y) Direction ($\dot{X}, \dot{Y}$) For Saif Scenario 1 for large look ahead distance

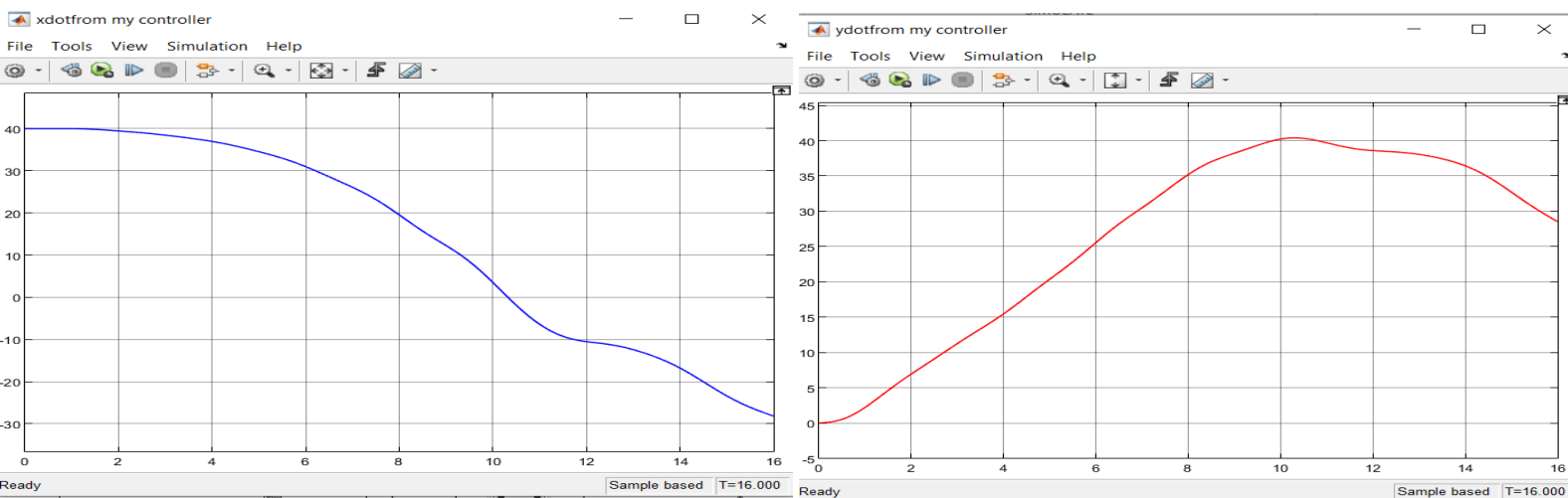


Figure 3.49: Pure Pursuit Controller The Results For Large Look Ahead Distance

The second driving scenario was created and applied to the controller the radius ($R=135.6\text{m}$) the curvature is equal ($K=0.0073746313$)

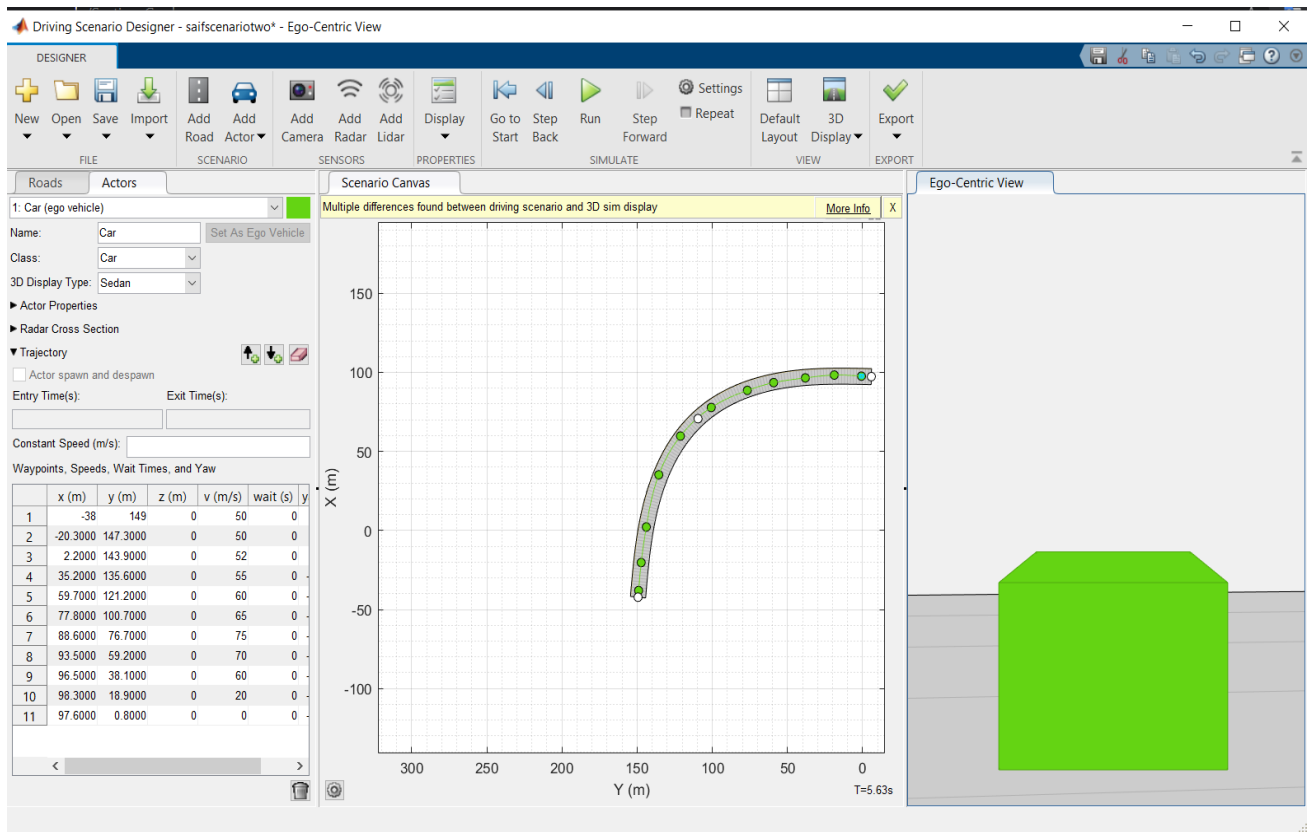


Figure 3.52:Saif Scenario 2 For Path Tracking With Radius =135.6 m

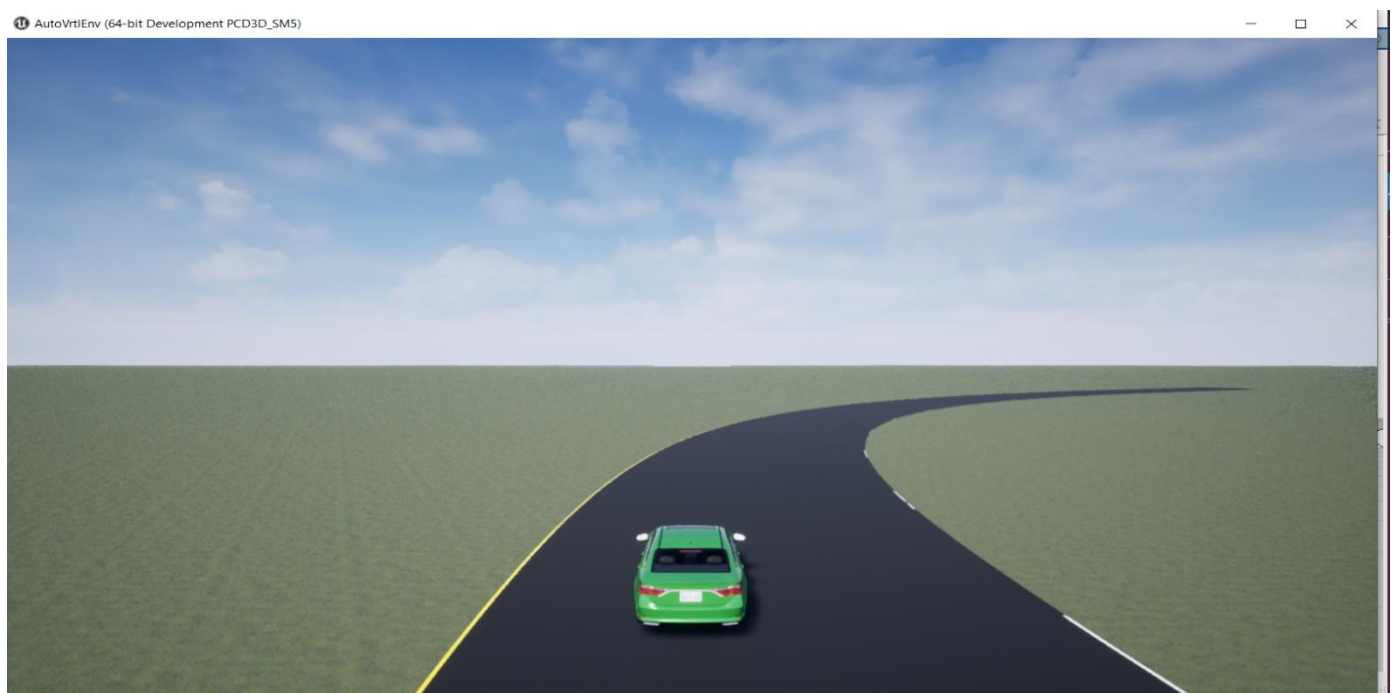


Figure 3.53: 3D Animation For Tracking A Path For Scenario 2

The results when the look ahead distance is small the response of vehicle is too bad as well as the wheels moving very fast with high response as the figures below shows

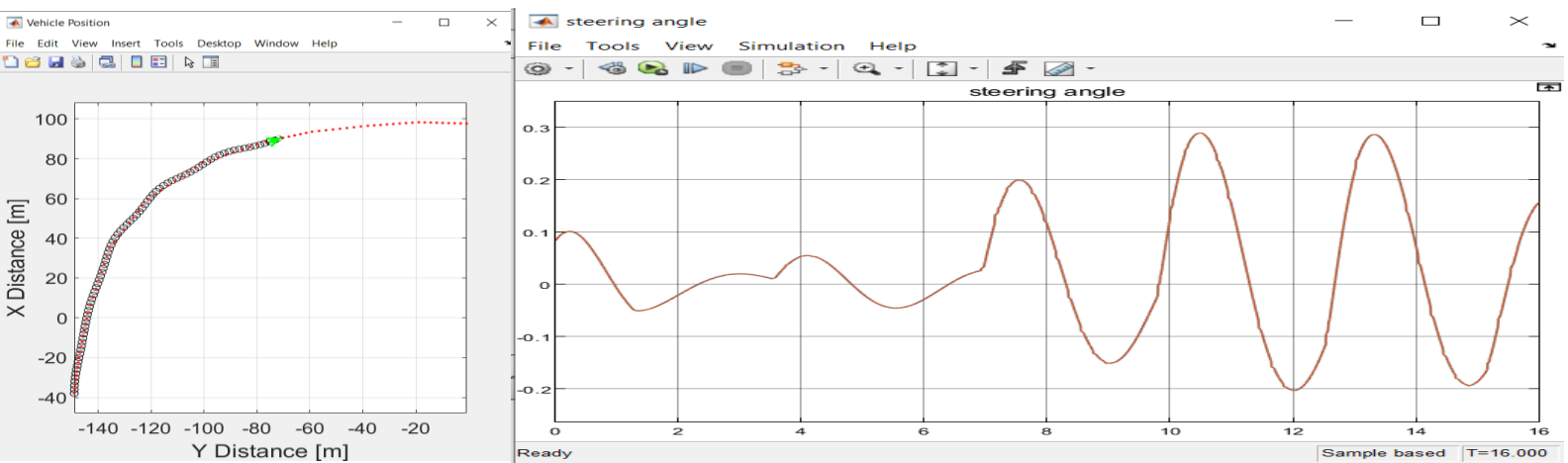


Figure 3.54: Vehicle Position By 2D-Visualiza For Saif Scenario 2 with Small Look Ahead Distance And Vehicle Steering Angle (δ)

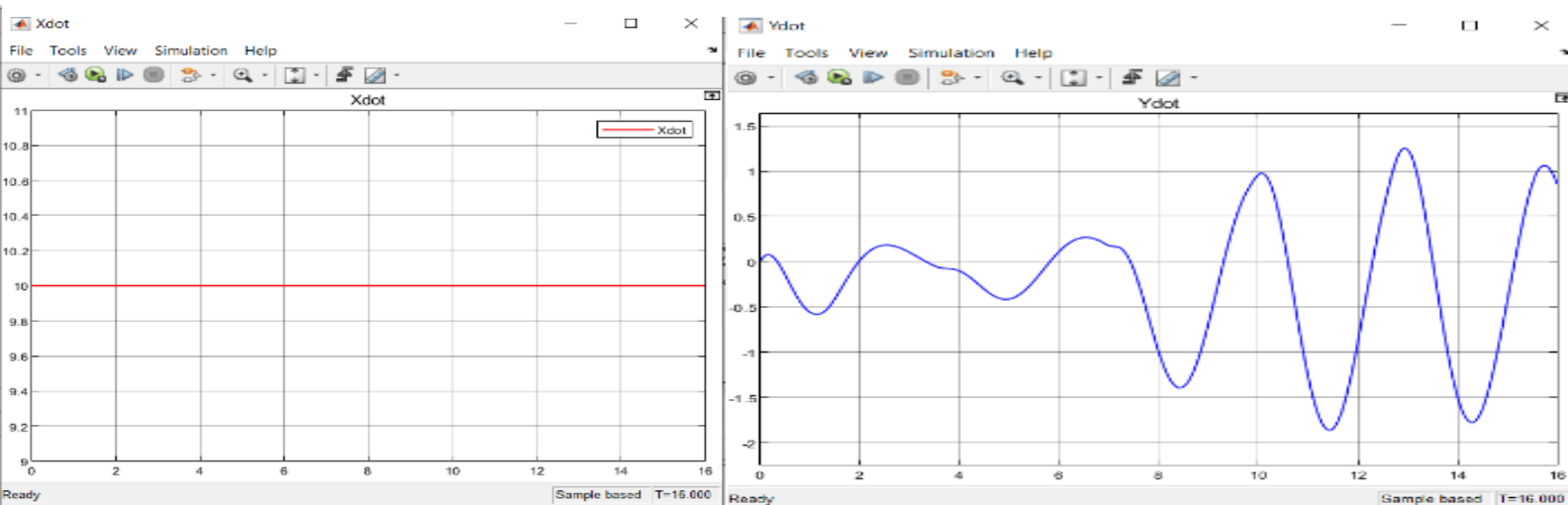


Figure 3.55: Vehicle Velocity At The (X, Y) Direction ($\dot{X}, \dot{Y}$) For Saif Scenario 2 For Small Look Ahead

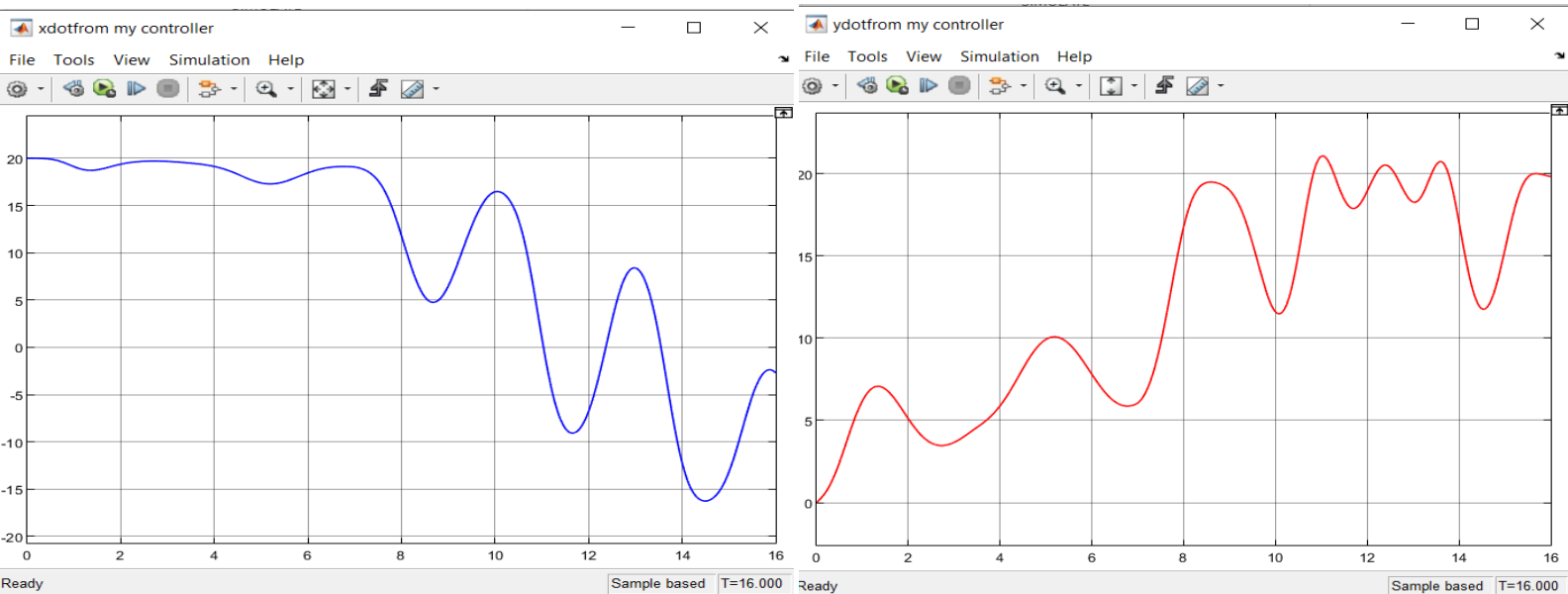


Figure 3.56: Pure Pursuit Controller the Results for Small Look Ahead Distance

The results when the look ahead distance is large the response is good for vehicle wheel as the figures below shows

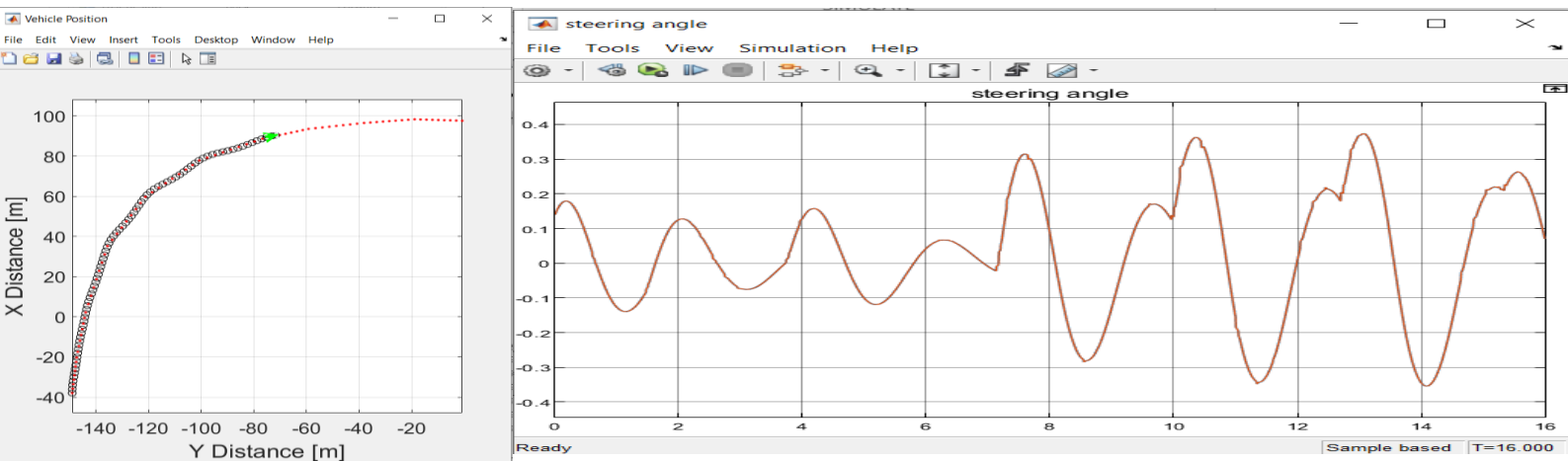


Figure 3.58: Vehicle Position By 2D-Visualiza For Saif Scenario 2 with large Look Ahead Distance And Vehicle Steering Angle (δ)

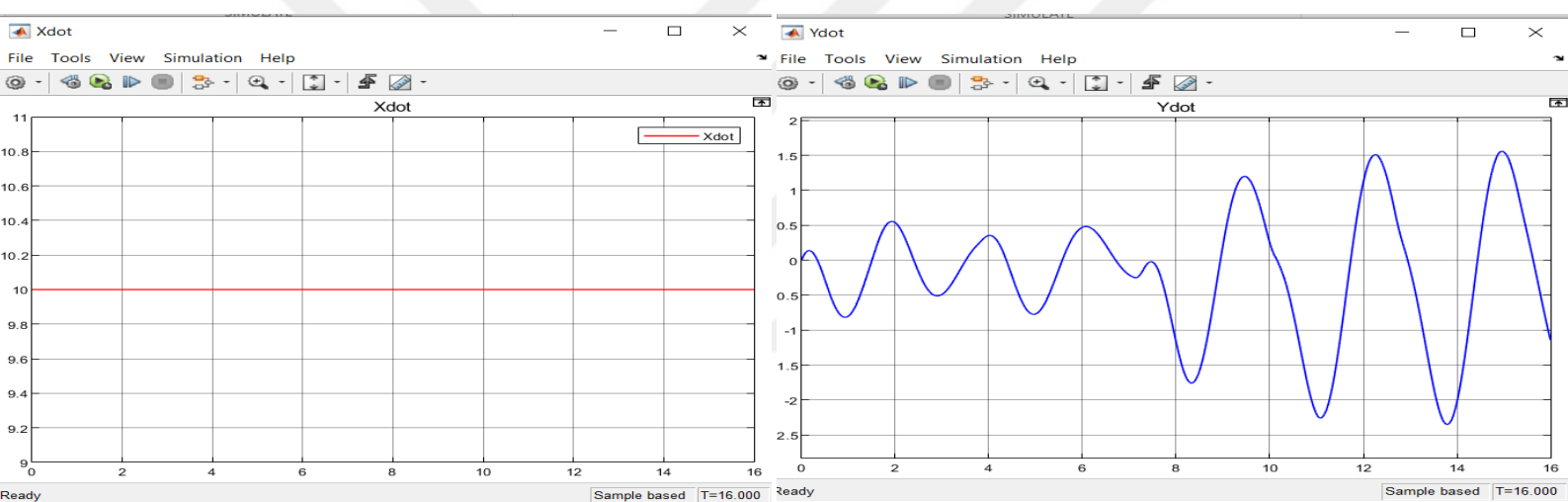


Figure 3.57: Vehicle Velocity At The (X, Y) Direction ($\dot{X}, \dot{Y}$) For Saif Scenario 2 for large look ahead distance

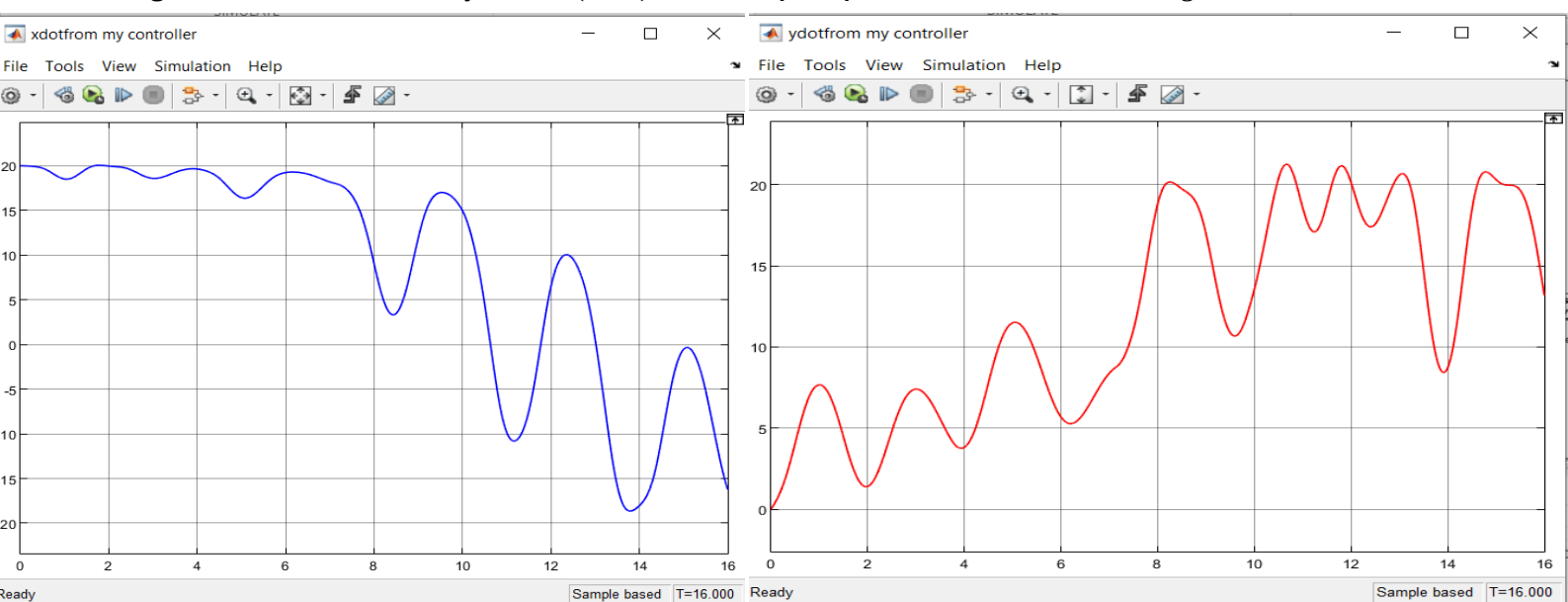


Figure 3.59: Pure Pursuit Controller the Results for Large Look Ahead Distance

CHAPTER 4. CHANGE THE PARAMETERS THAT WAS USED TO SEE ITS EFFECT

In this chapter, the parameter was changed to study the effect in details, as it was changed as shown in the arrangement of the sub-titles of this chapter. Where the study was done in the following manner, when the value of the distance between the front and rear wheels of the vehicle is changed by large and small values for the x and y axes. The center of gravity for the two axes, as well as the study of changing the center of gravity for the Stanley method.

4.1. STUDY THE EFFECT OF CHANGING THE DISTANCE BETWEEN THE FRONT AND REAR WHEELS (WHEELBASE):

Now the study revolved around changing the wheelbase of the vehicle wheels, noting the difference when changing the looking head distance, as well as with changing the speeds of the car, when the car follows three different points to see the difference between them with the change in the distance between the wheels of the car. The subject was studied in the event that the car follows the x axis and the y axis in the same initial conditions.

4.1.1. Changing the Wheelbase When the Vehicle Following the Y-Axis (High L)

The first case is if the car follows the Y axis with a change in the value of $L=3.4\text{m}$ and the $(U=10\text{m/s})$, $L_d=(4,12,22)$:

```

%The parameters of the vehicle
clear all
mass=1250;
Iz=1550;
ad=1.7;
bd=1.7;
Cfront=50000;
Crear=50000;
U=10;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];

```

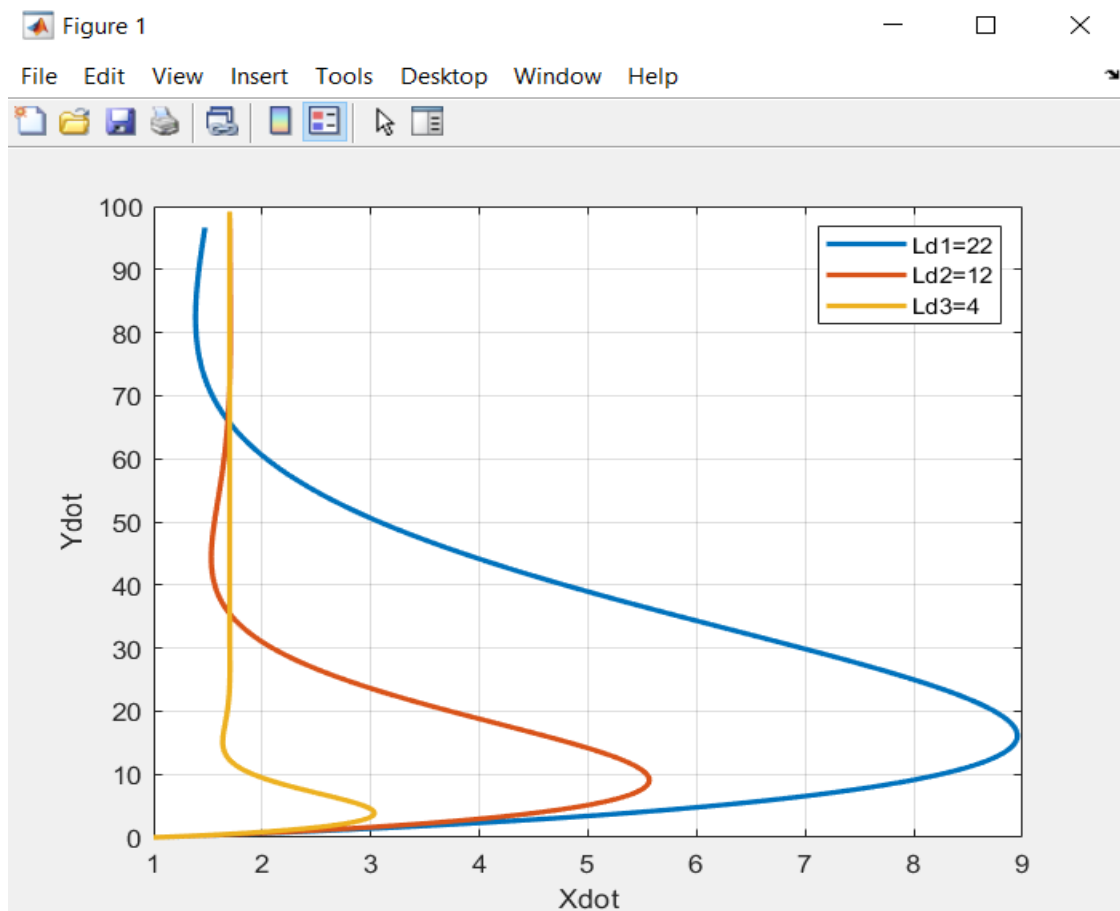


Figure 4.1: The Vehicle Follows the Y Axis with A Change In the Value of $L=3.4$ m and the ($U=10\text{m/s}$), $L_d=(4, 12, 22 \text{ m})$

L=3.4 m and the (U=15m/s), $L_d=(4,12,22)$:

%The parameters of the vehicle

```
clear all
```

```
mass=1250;
```

```
Iz=1550;
```

```
ad=1.7;
```

```
bd=1.7;
```

```
Cfront=50000;
```

```
Crear=50000;
```

```
U=15;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
```

```
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

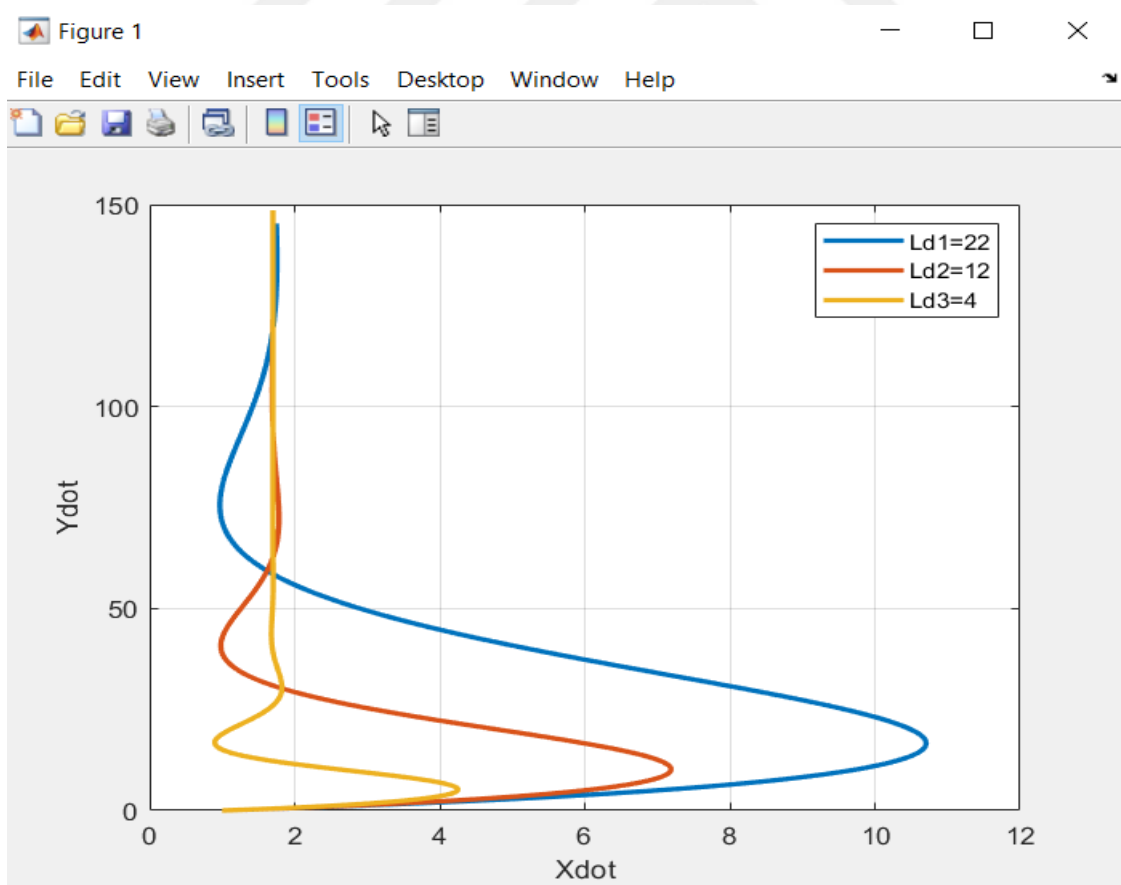


Figure 4.2: The Vehicle Follows the Y Axis with A Change In the Value of L=3.4m and the (U=15m/s), $L_d=(4, 12, 22 \text{ m})$

$L=3.4$ m and the $(U=19\text{m/s})$, $L_d=(4,12,22)$ and the controller will crash at 20m/s :

%The parameters of the vehicle

```
clear all
mass=1250;
Iz=1550;
ad=1.7;
bd=1.7;
Cfront=50000;
Crear=50000;
U=19;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

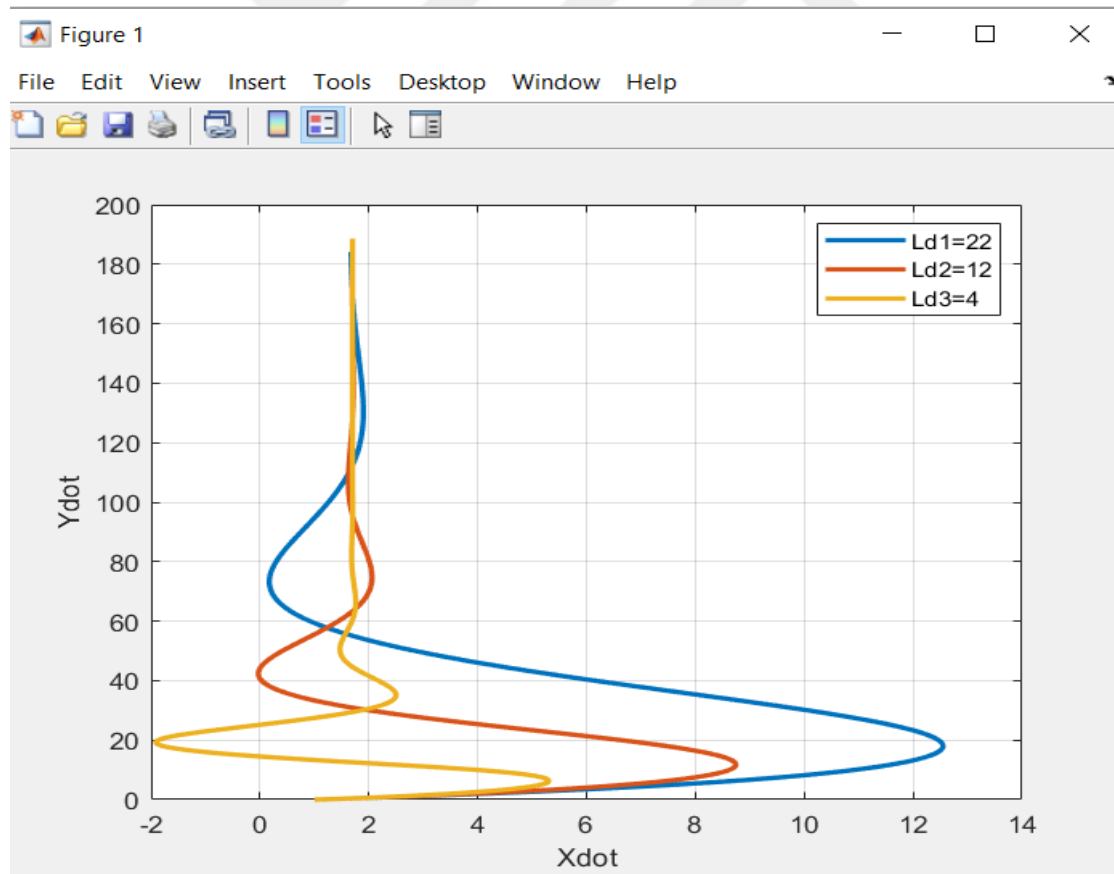


Figure 4.3: The Vehicle Follows the Y Axis with A Change In the Value of $L=3.4\text{m}$ and the $(U=19\text{m/s})$, $L_d=(4, 12, 22 \text{ m})$

The figures below show that the crash of the controller when it reaches ($U=20$ m/s)

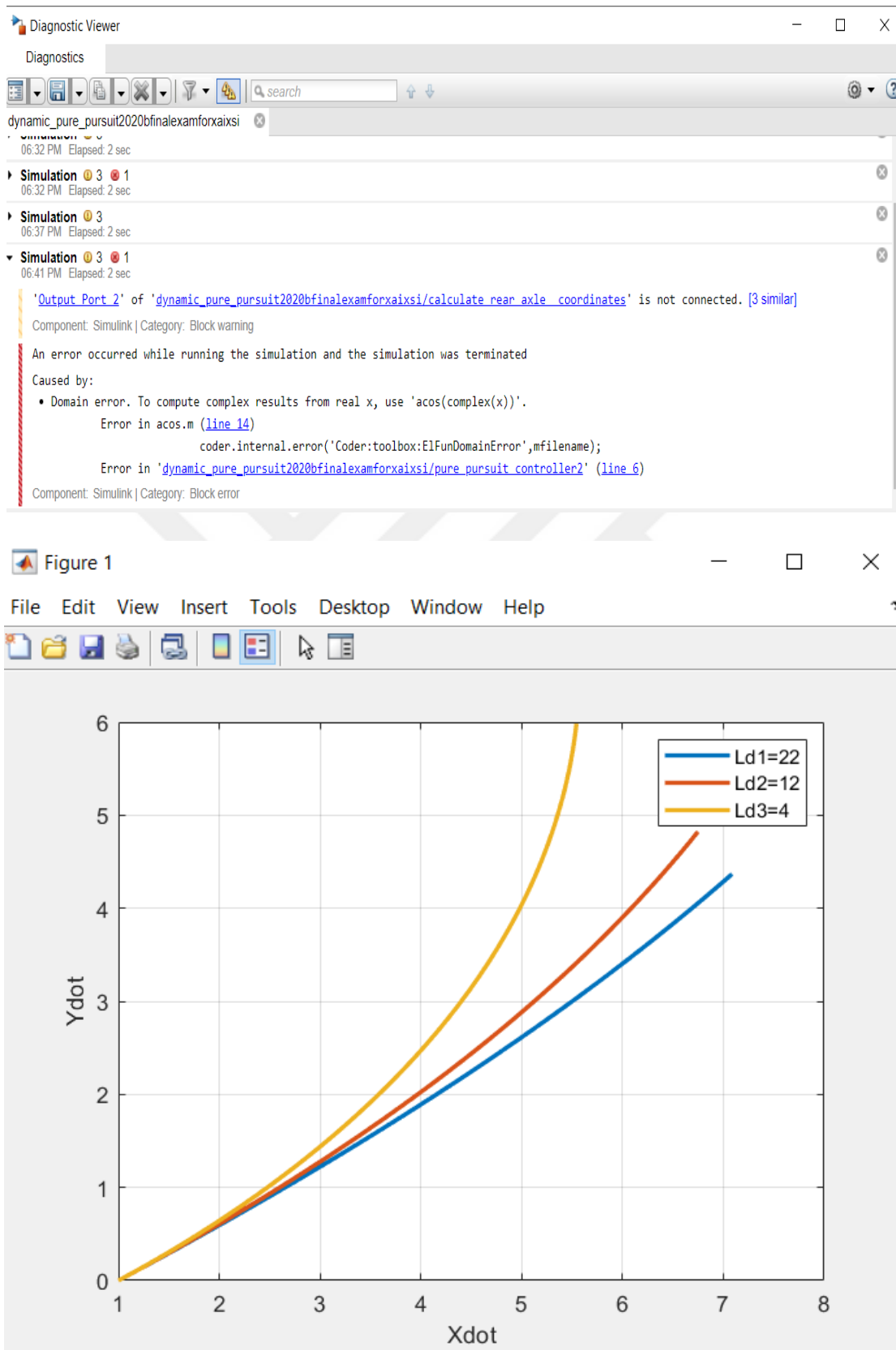


Figure 4.4: The Vehicle Follows the Y Axis with A Change In the Value of $L=3.4$ m and the ($U=20$ m/s), $L_d=$ (4, 12,22m)

4.1.2. Changing the Wheelbase When the Vehicle Following the Y-Axis (Low L)

The second case is if the car follows the Y axis with a change in the value of $L=0.8\text{m}$ and the ($U=10\text{m/s}$), $L_d=(4,12,22)$ with lower mass :

%The parameters of the vehicle

```
clear all
mass=25;
Iz=175;
ad=0.4;
bd=0.4;
Cfront=20000;
Crear=20000;
U=10;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

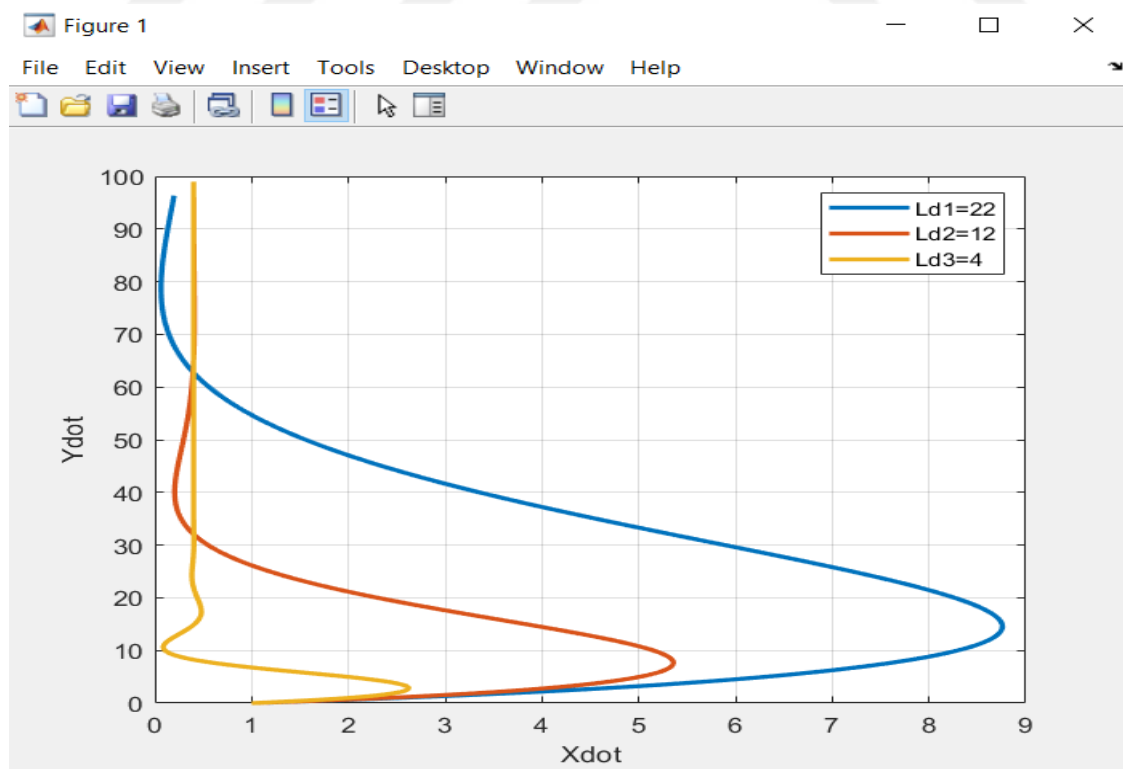


Figure 4.5: The Vehicle Follows the Y Axis with A Change In the Value of $L=0.8\text{m}$ and the ($U=10\text{m/s}$), $L_d=(4, 12, 22\text{m})$

$L=0.8\text{m}$ and the ($U=15\text{m/s}$), $L_d=(4,12,22)$ with lower mass :

%The parameters of the vehicle

```
clear all
mass=25;
Iz=175;
ad=0.4;
bd=0.4;
Cfront=20000;
Crear=20000;
U=15;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

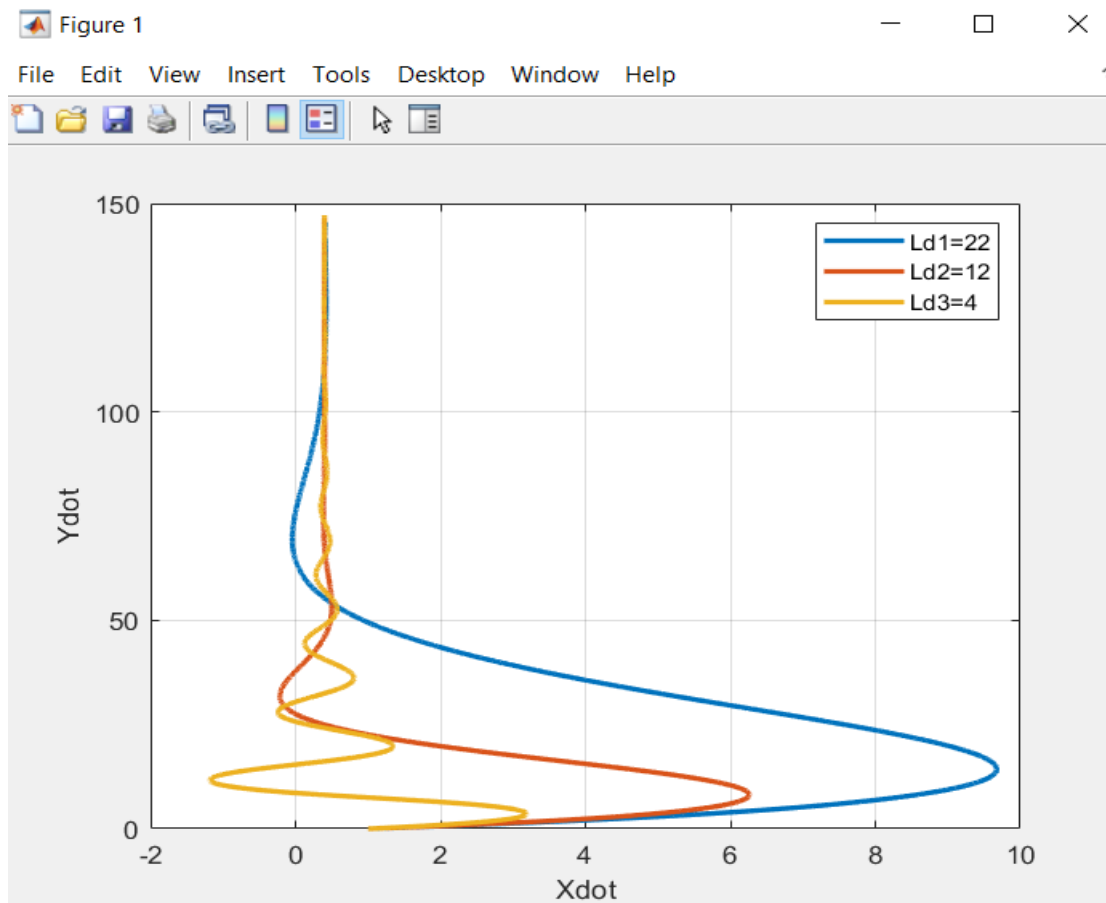


Figure 4.6: The Vehicle Follows the Y Axis with A Change In the Value of $L=0.8\text{m}$ and the ($U=15\text{m/s}$), $L_d=(4, 12,22\text{m})$

$L=0.8$ m and the ($U=19\text{m/s}$), $L_d=(4,12,22)$ and the controller will crash at 20m/s :

%The parameters of the vehicle

```
clear all
```

```
mass=25;
```

```
Iz=175;
```

```
ad=0.4;
```

```
bd=0.4;
```

```
Cfront=20000;
```

```
Crear=20000;
```

```
U=19;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

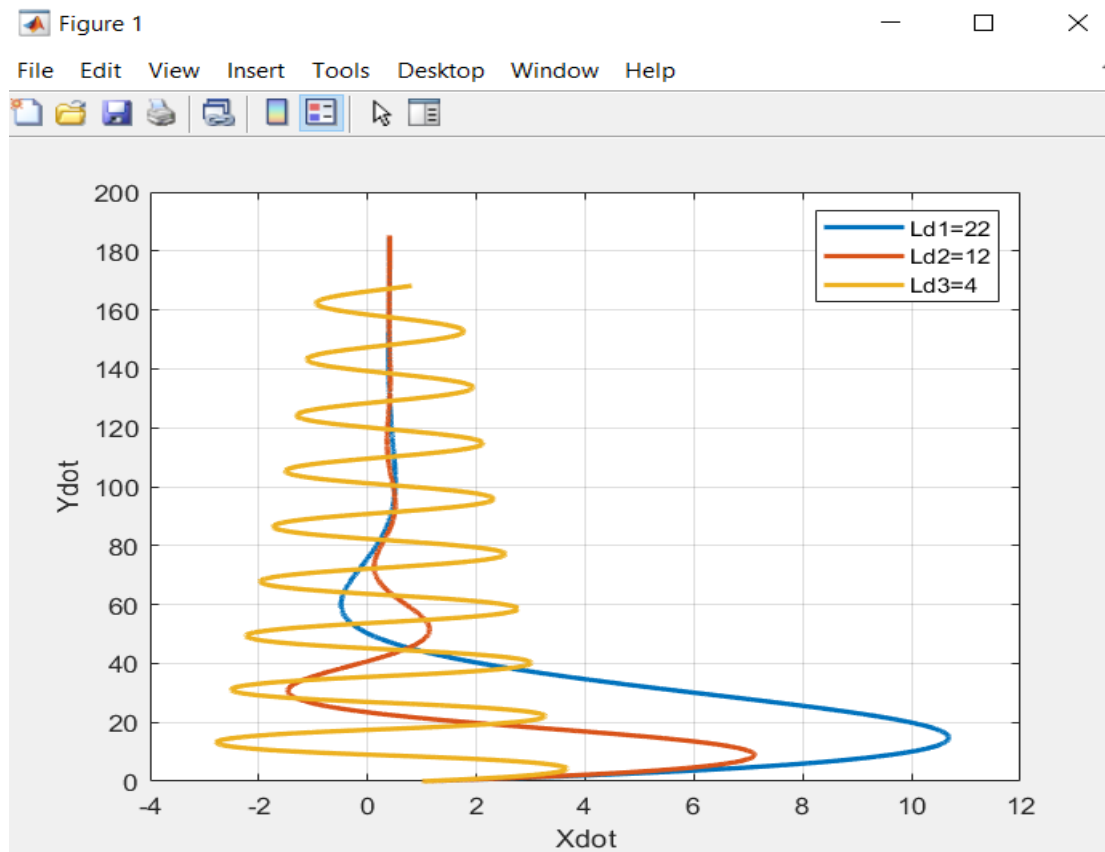


Figure 4.7: The Vehicle Follows the Y Axis with A Change In the Value of $L=0.8\text{m}$ and the ($U=19\text{m/s}$), $L_d=(4, 12, 22 \text{ m})$

The figures below show that the crash of the controller when it reaches ($U=20$ m/s)

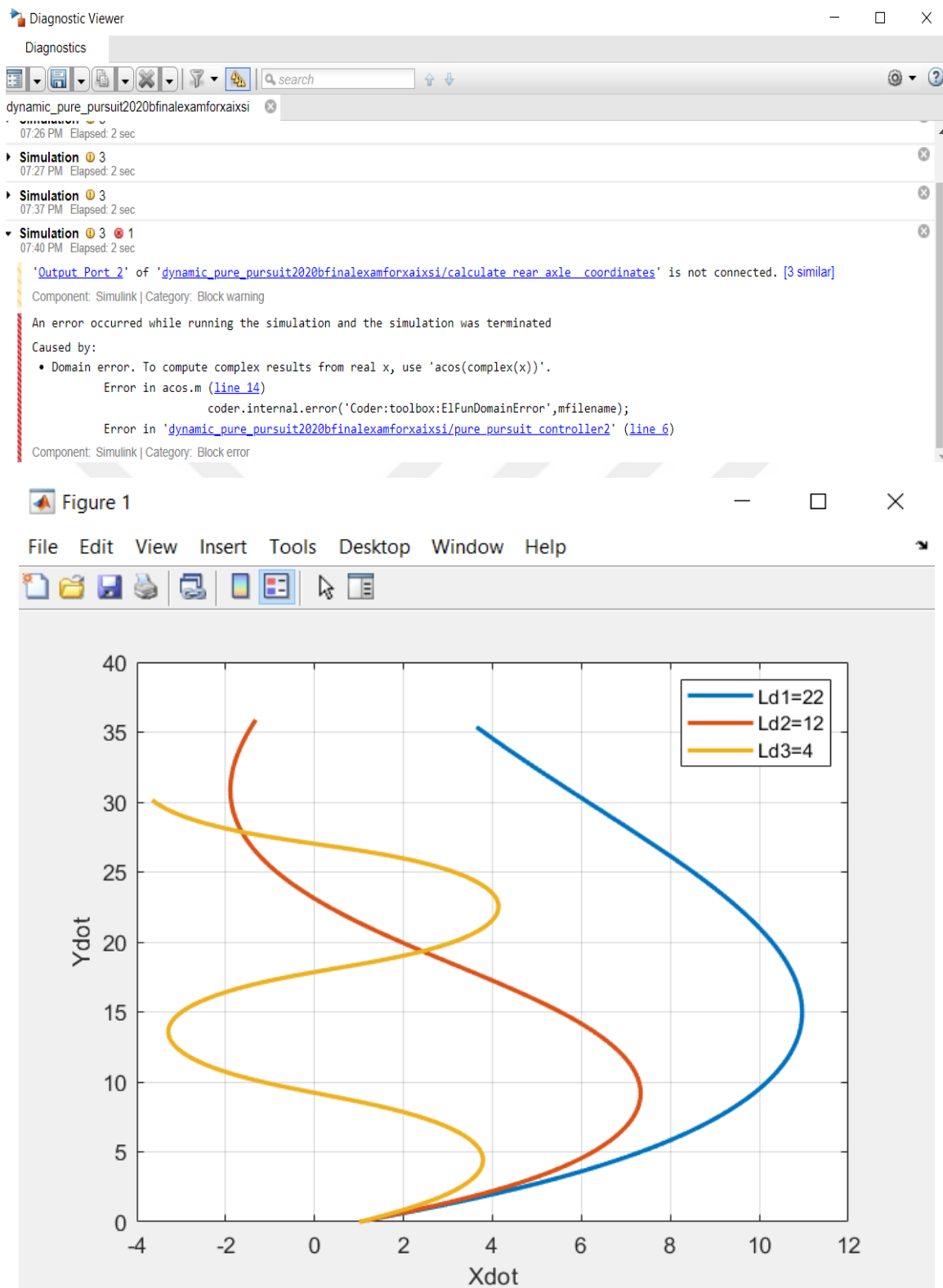


Figure 4.8: The Vehicle Follows the Y Axis with A Change In the Value of $L=0.8$ m and the ($U=20$ m/s), $L_d=(4, 12, 22$ m)

In this test, the distance between the front and rear wheels of the car was changed to three different values. The relationship was linked with the speed of the vehicle. We have noted through the graphics that when the distance between the front and rear wheels is large, the instability and the stubbornness of the car, when increasing the speed at the same time increases gradually until it reaches a point where the vehicle cannot control, the control unit for tracking the point, and leads to a malfunction In control, but when the distance between the wheels decreases, as was mentioned previously in the second case, the instability and instability appear even at low speeds. It gradually increases with the increase in speed so that if we put a comparison between the two cases at a speed of 19 m / s in both cases when the vehicle follows the point in Y axis .As it is at this speed that it appears from the graphics that the instability increases a lot and increases the instability and the response of the car becomes much worse when the wheelbase is small like a scooter, while it looks a little better when compared when the wheelbase is large, but in both cases the vehicle remains in a state of instability, I would also like to note that the vehicle's mass plays an important role in the results obtained, as well as L_d . The relationship at each speed with the difference in L_d also affects the stability of the vehicle. The greater the vehicle's mass or the greater the distance head point leads to the instability of the vehicle in all cases.

4.1.3. Changing the Wheelbase When the Vehicle Following the X-Axis (High L)

Now the study revolved around changing the wheelbase of the vehicle wheels, noting the difference when changing the looking head distance, as well as with changing the speeds of the car, when the car follows three different points to see the difference between them with the change in the distance between the wheels of the car. The subject was studied in the event that the car follows the X axis with the same initial conditions that was used in Y axis following.

The first case is if the car follows the X axis with a change in the value of $L=3.4\text{m}$ and the $(U=10\text{m/s})$, $L_d=(4,12,22)$:

```
%The parameters of the vehicle
```

```
clear all
```

```
mass=1250;
```

```
Iz=1550;
```

```
ad=1.7;
```

```
bd=1.7;
```

```
Cfront=50000;
```

```
Crear=50000;
```

```
U=10;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

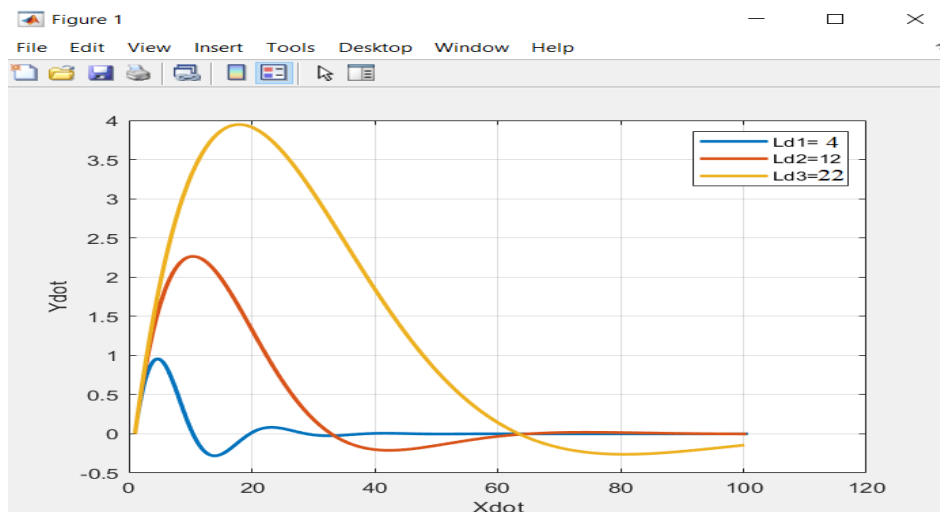


Figure 4.9: The Vehicle Follows the X Axis with A Change In the Value of $L=3.4\text{m}$ and the $(U=10\text{m/s})$, $L_d=(4, 12, 22\text{m})$

If the car follows the X axis with a change in the value of $L=3.4\text{m}$ and the ($U=15\text{m/s}$), $L_d = (4, 12, 22\text{m})$:

%The parameters of the vehicle

```
clear all
mass=1250;
Iz=1550;
ad=1.7;
bd=1.7;
Cfront=50000;
Crear=50000;
U=15;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

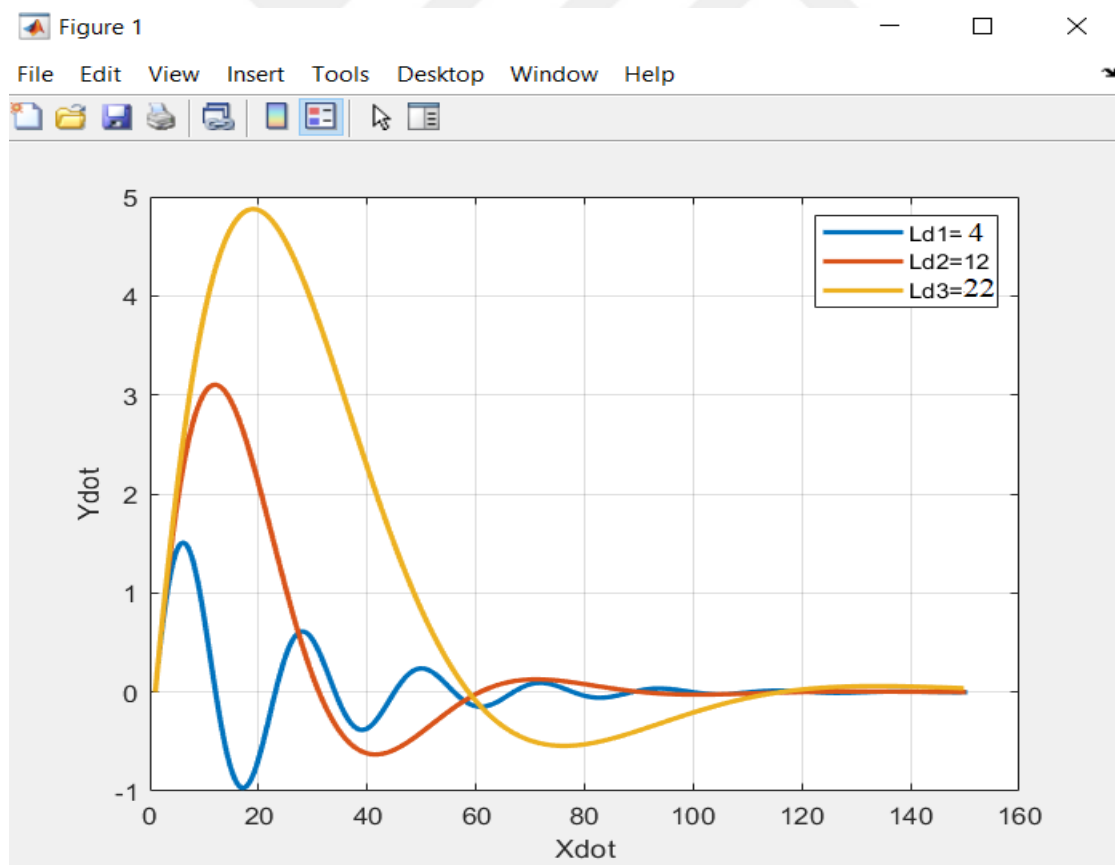


Figure 4.10: The Vehicle Follows the X Axis with A Change In the Value of $L=3.4\text{m}$ and the ($U=15\text{m/s}$), $L_d = (4, 12, 22\text{m})$

If the car follows the X axis with a change in the value of $L=3.4\text{m}$ and the ($U=19\text{m/s}$), $L_d=(4,12,22\text{m})$:

%The parameters of the vehicle

```
clear all
mass=1250;
Iz=1550;
ad=1.7;
bd=1.7;
Cfront=50000;
Crear=50000;
U=19;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

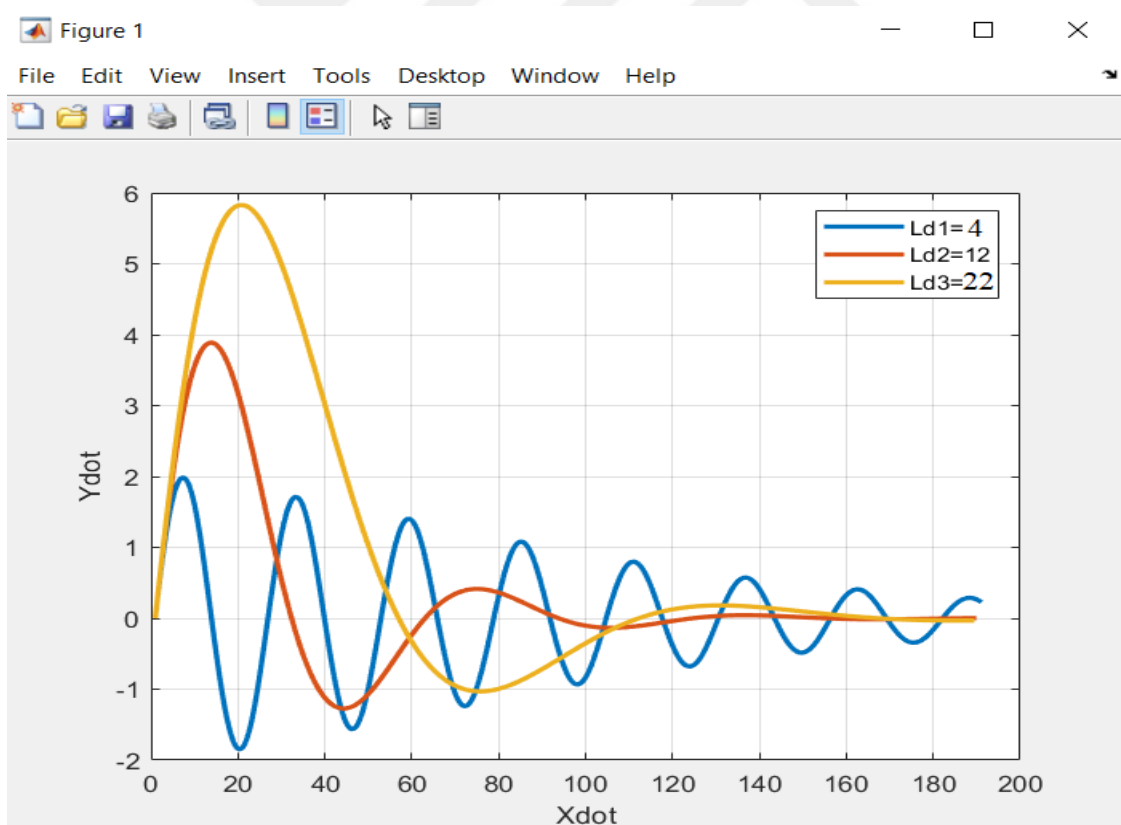


Figure 4.11: The Vehicle Follows the X Axis with A Change In the Value of $L=3.4\text{m}$ and the ($U=19\text{m/s}$), $L_d= (4, 12, 22 \text{ m})$

The figures below show that the crash of the controller when it reaches ($U=20$ m/s)

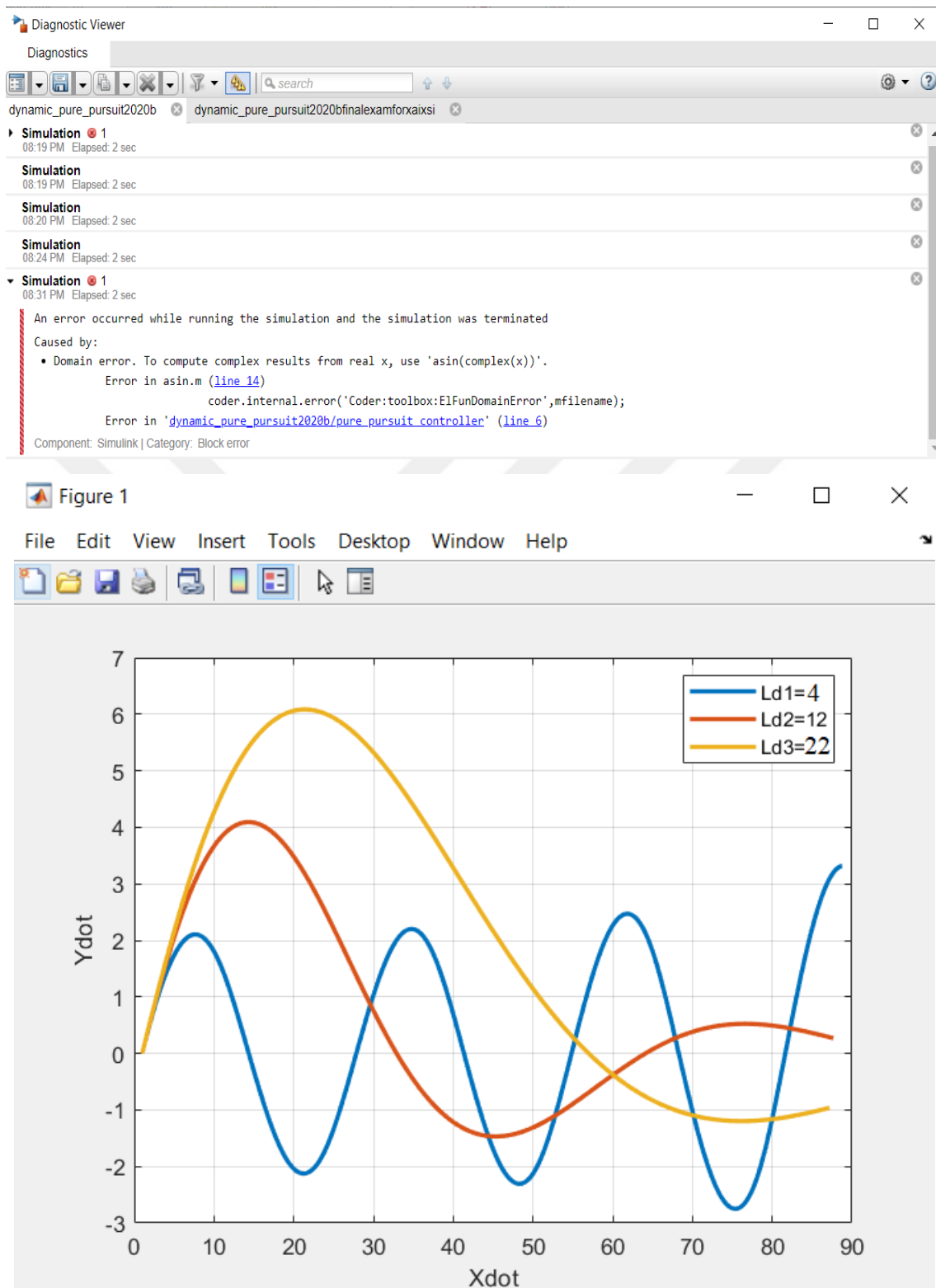


Figure 4.12: The Vehicle Follows the X Axis with A Change In the Value of $L=3.4$ m and the ($U=20$ m/s), $L_d = (4, 12, 22$ m)

4.1.4. Changing the Wheelbase When the Vehicle Following the X-Axis (Low L)

The second case is if the car follows the X axis with a change in the value of $L=0.8\text{m}$ and the ($U=10\text{m/s}$), $L_d=(4,12,22\text{m})$ with lower mass :

%The parameters of the scooter

```
clear all
mass=25;
Iz=175;
ad=0.4;
bd=0.4;
Cfront=20000;
Crear=20000;
U=10;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

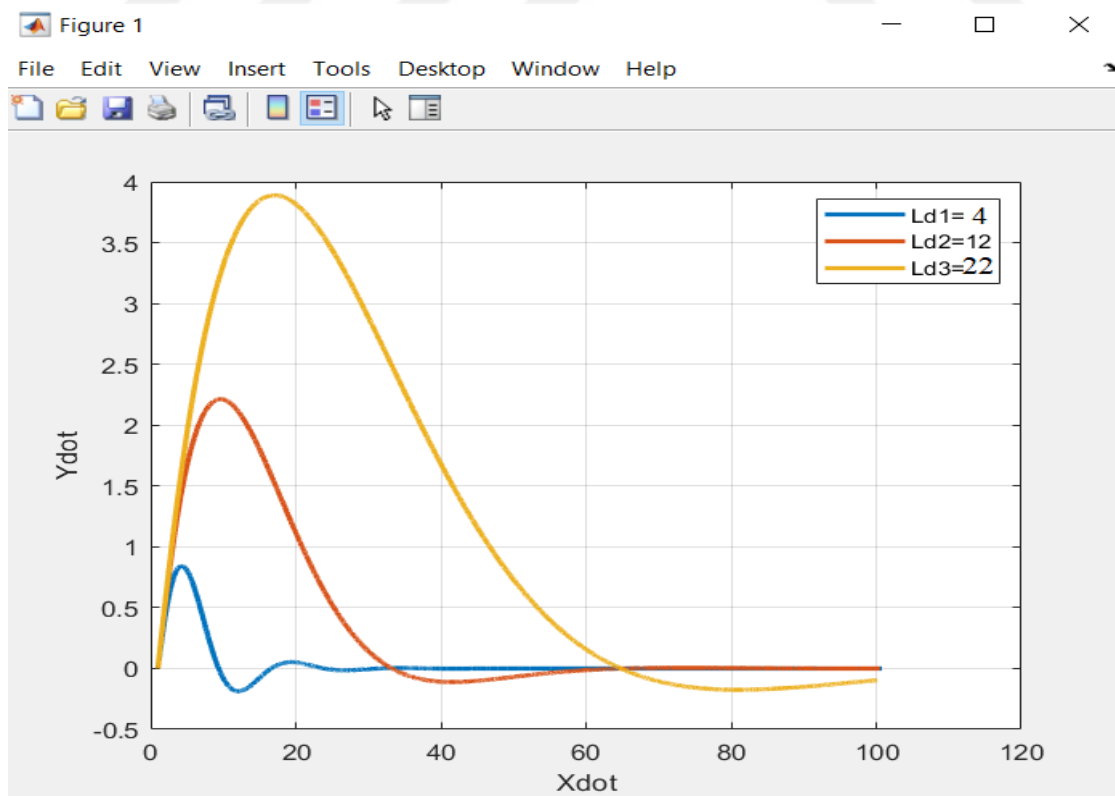


Figure 4.13: The Vehicle Follows the X Axis with A Change In the Value of $L=0.8\text{m}$ and the ($U=10\text{m/s}$), $L_d = (4, 12, 22\text{m})$

If the car follows the X axis with a change in the value of $L=0.8\text{m}$ and the ($U=15\text{m/s}$), $L_d=(4,12,22\text{m})$:

%The parameters of the scooter

```
clear all
mass=25;
Iz=175;
ad=0.4;
bd=0.4;
Cfront=20000;
Crear=20000;
U=15;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

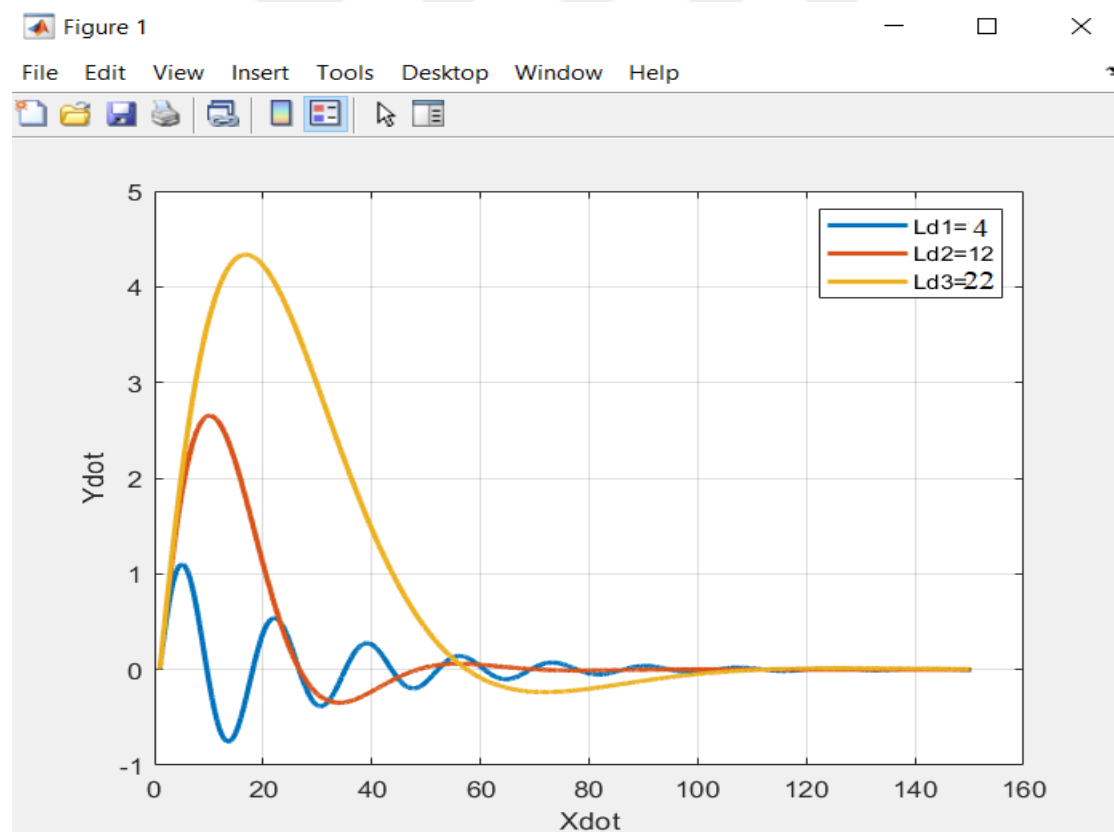


Figure 4.14: The Vehicle Follows the X Axis with A Change In the Value of $L=0.8\text{m}$ and the ($U=15\text{m/s}$), $L_d=(4,12,22\text{m})$

If the car follows the X axis with a change in the value of $L=0.8\text{m}$ and the ($U=19\text{m/s}$), $L_d=(4,12,22)$:

%The parameters of the scooter

```
clear all
mass=25;
Iz=175;
ad=0.4;
bd=0.4;
Cfront=20000;
Crear=20000;
U=19;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

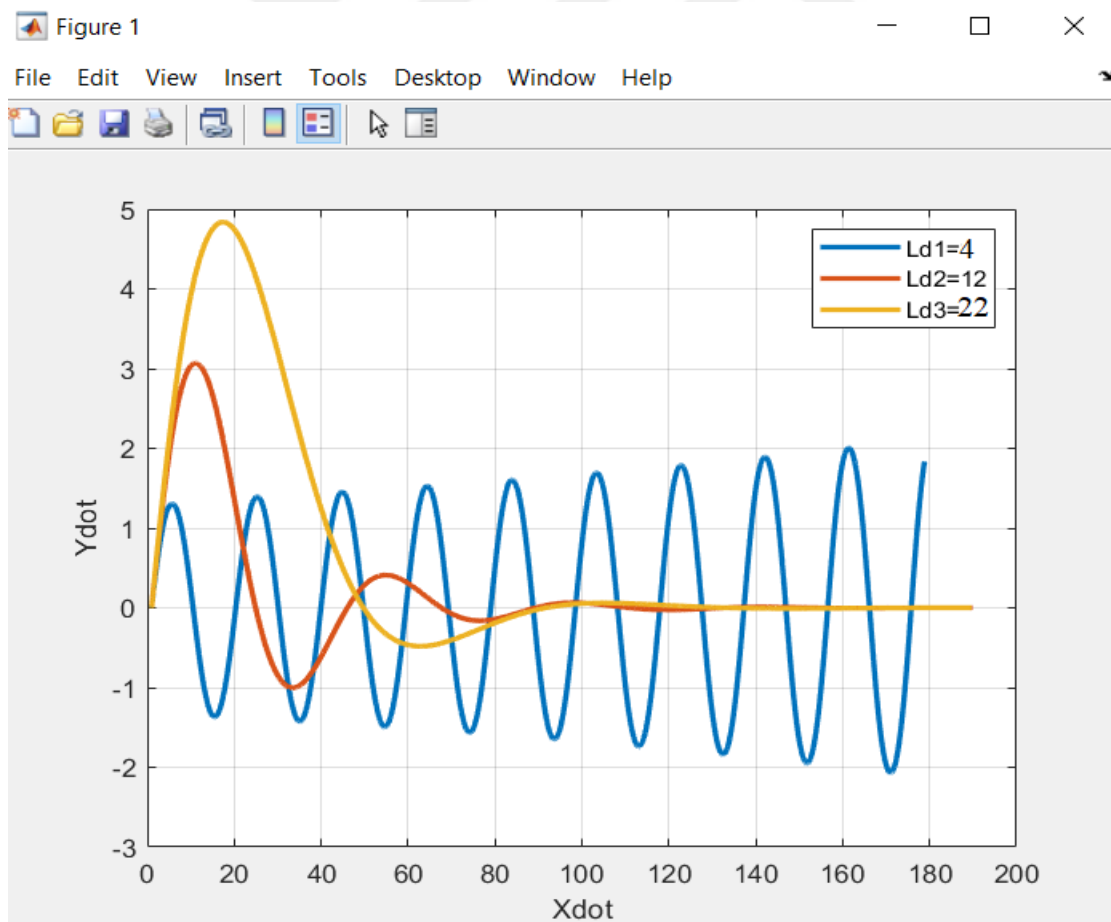


Figure 4.15: The Vehicle Follows the X Axis with A Change In the Value of $L=0.8\text{m}$ and the ($U=19\text{m/s}$), $L_d=(4,12,22\text{m})$

The figures below show that the crash of the controller when it reaches ($U=20$ m/s)

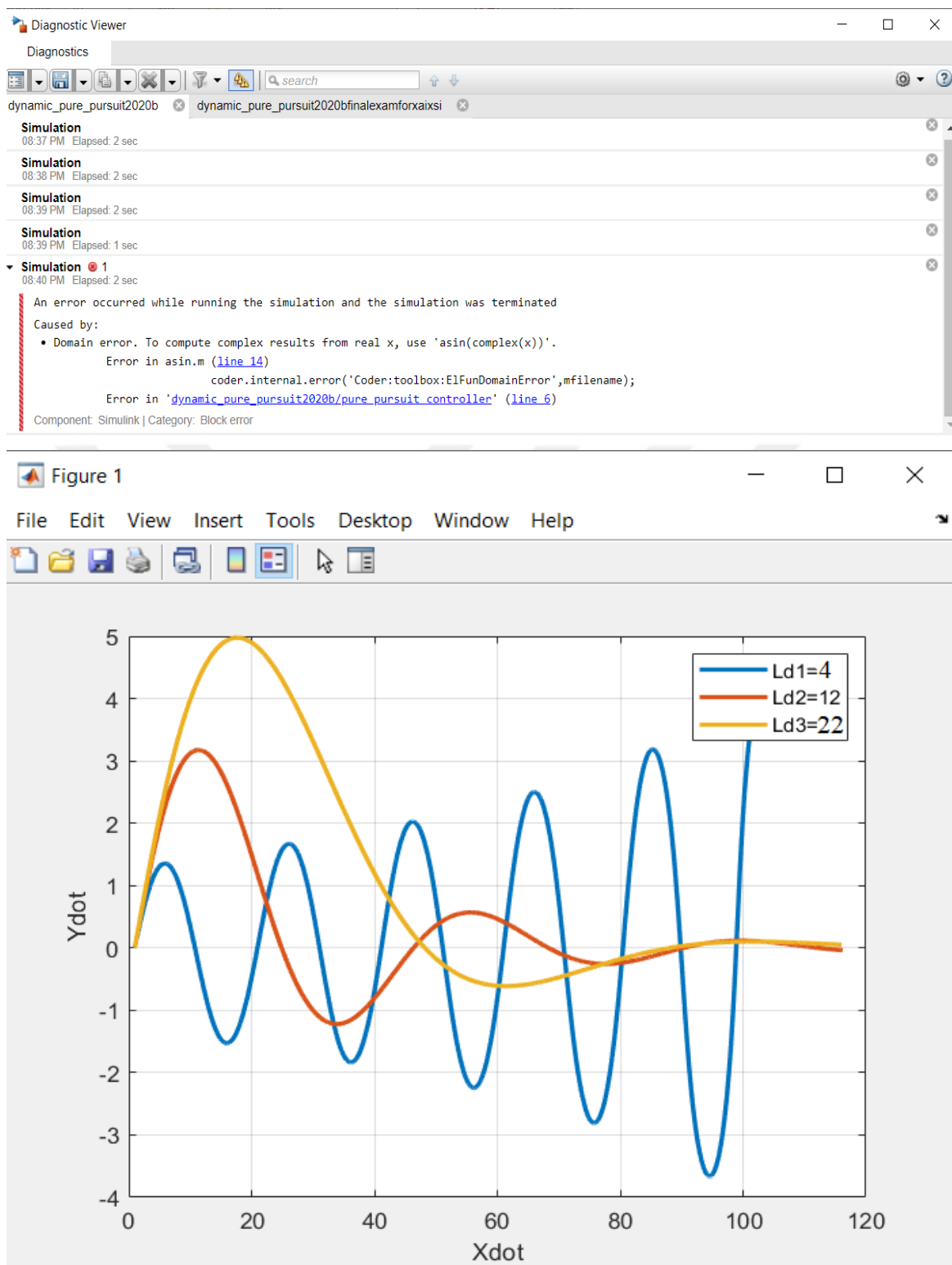


Figure 4.16: The Vehicle Follows the X Axis with A Change In the Value of $L=0.8$ m and the ($U=20$ m/s), $L_d= (4, 12, 22)$ m

4.2. STUDY THE EFFECT OF CHANGING THE MASS:

The simulation was applied and the result was seen for both the bicycle and the truck, with the difference in the distance between the wheels and the mass, where the result was seen when the bicycle and truck tracked to the point on the X-axis and Y-axis where all parameters were taken into account.

4.2.1. Bicycle Features:

The bicycle has a mass of 24 kg, and the distance between the front and rear wheels is 1.17 meters. It was assumed that the weight of the driver equals 80 kg, so the total mass of both of them becomes 104 kg. All parameters have been placed in the code as well as in the simulation, and as shown below, there are the dimensions used for the bicycle. A real speed was used to move the bicycle (10, 20 and 30 Km/h), and it was also noted when a crash would occur for the controller.[8]

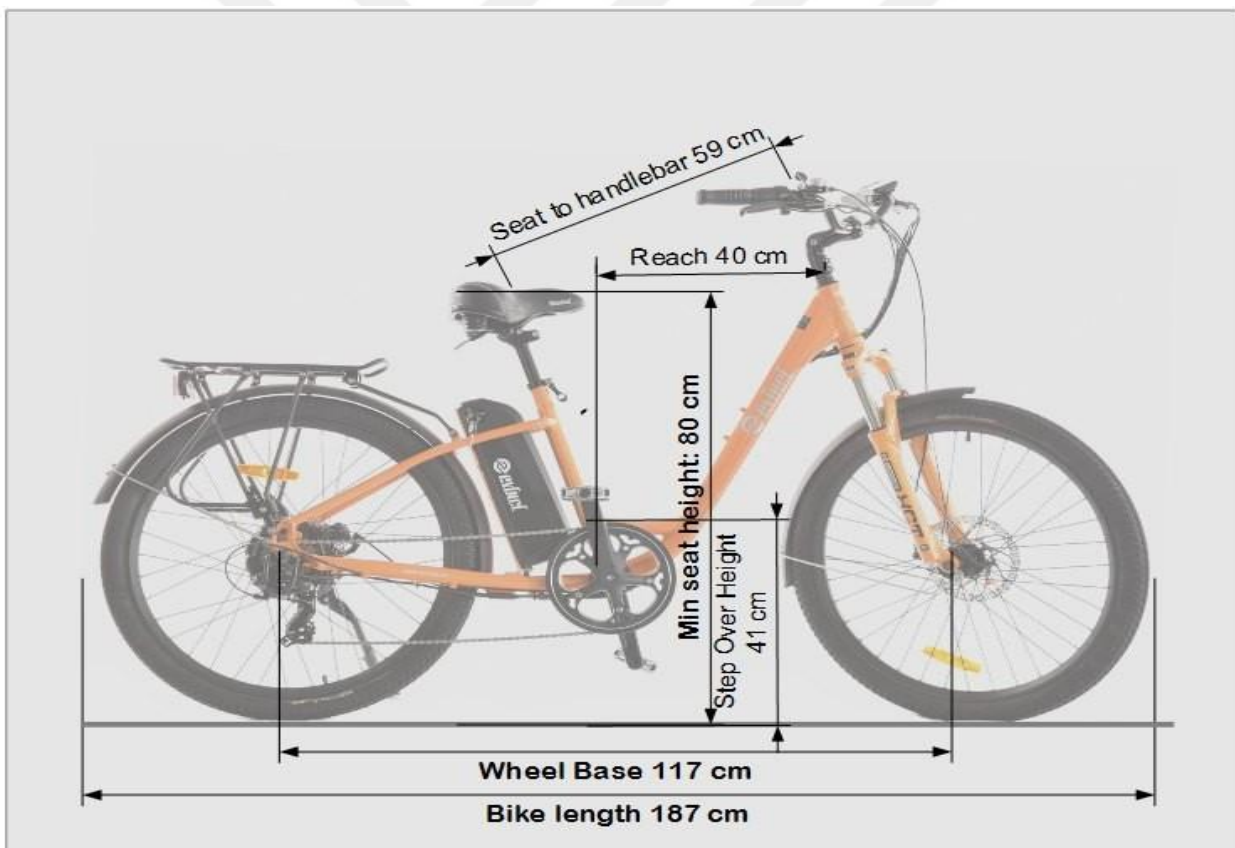


Figure 4.17: Dimensions of the Bicycle That Was Used

4.2.2. When the bicycle following the Y-axis

The bicycle tracks the Y-axis, where the results were applied to three different values of speed and three values of l_d and the result was seen at each of the speeds and knowing the date of the damage for controller. ($U=2.78\text{m/s}$), $L_d=(4,12,22)$

```
%Bicycle parameters
```

```
mass=104;
```

```
Iz=250;
```

```
ad=0.585;
```

```
bd=0.585;
```

```
Cfront=30000;
```

```
Crear=30000;
```

```
U=2.78;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

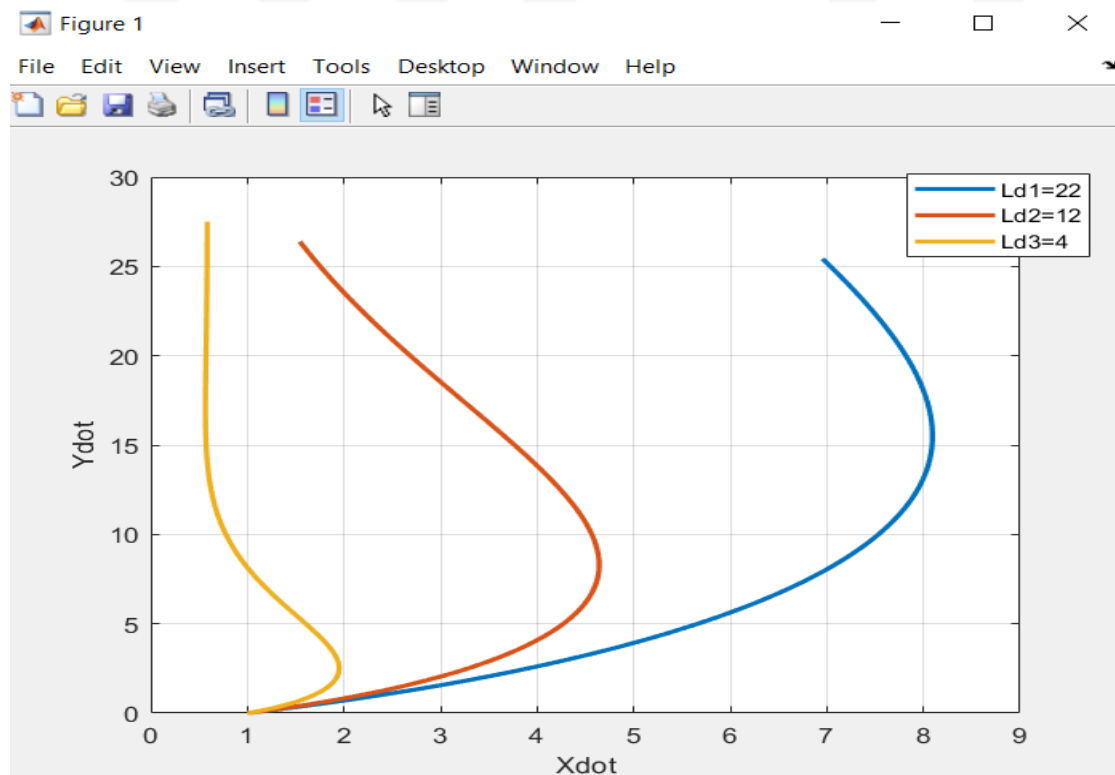


Figure 4.18: The Bicycle ($U=2.78\text{m/s}$) and $L_d = (4, 12, 22\text{m})$ When It's Following the Y-Axis

When the bicycle speed is 20 Km/h ($U=5.556\text{m/s}$), $L_d=(4,12,22)$

%Bicycle parameters

```
mass=104;
Iz=250;
ad=0.585;
bd=0.585;
Cfront=30000;
Crear=30000;
U=5.556;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

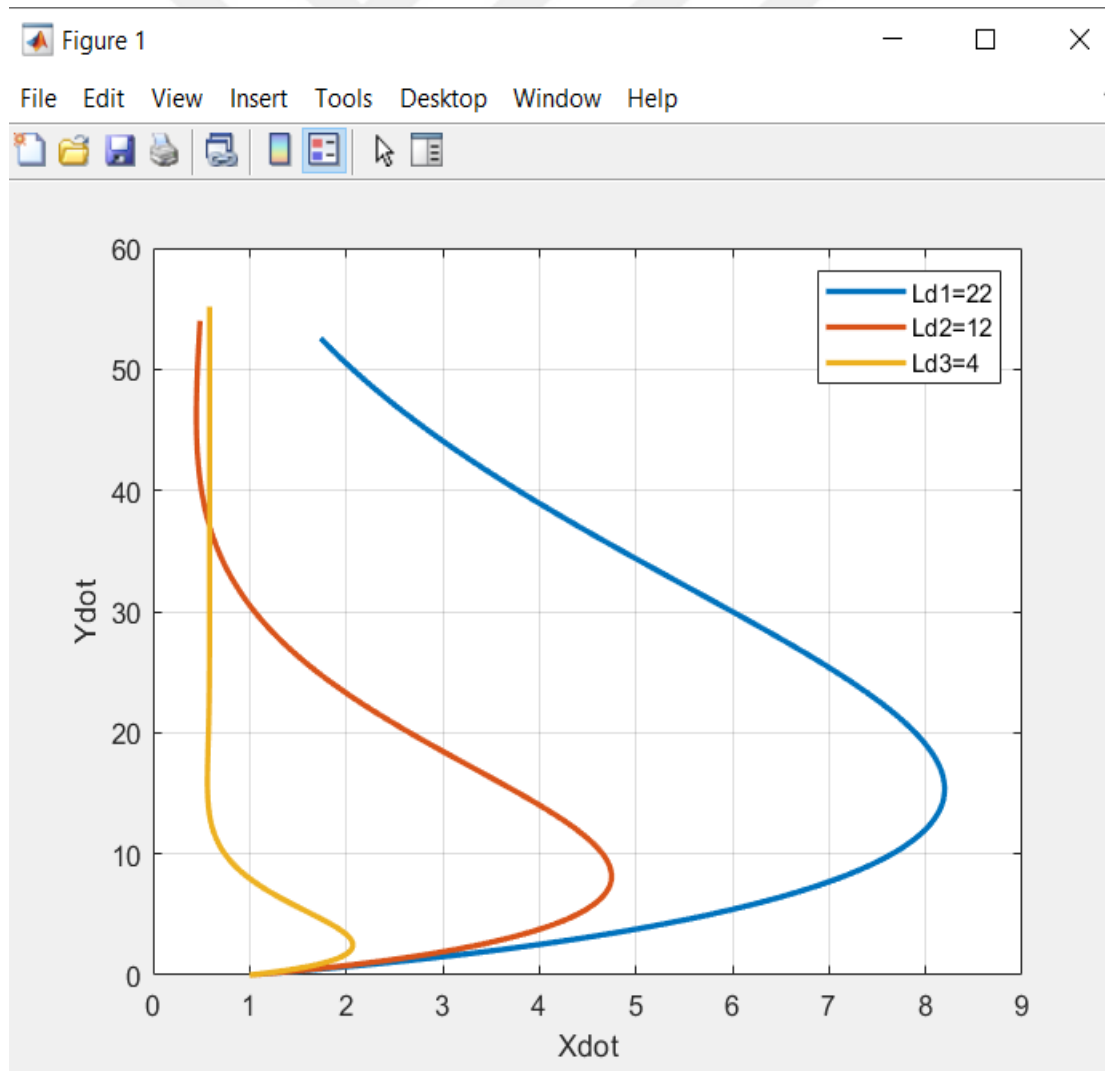


Figure 4.19: The Bicycle ($U=5.556\text{m/s}$), $L_d=(4, 12, 22\text{m})$ When It's Following the Y-Axis

When the bicycle speed is 30 Km/h ($U=8.334\text{m/s}$), $L_d=(4,12,22\text{m})$

%Bicycle parameters

```
mass=104;
Iz=250;
ad=0.585;
bd=0.585;
Cfront=30000;
Crear=30000;
U=8.334;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

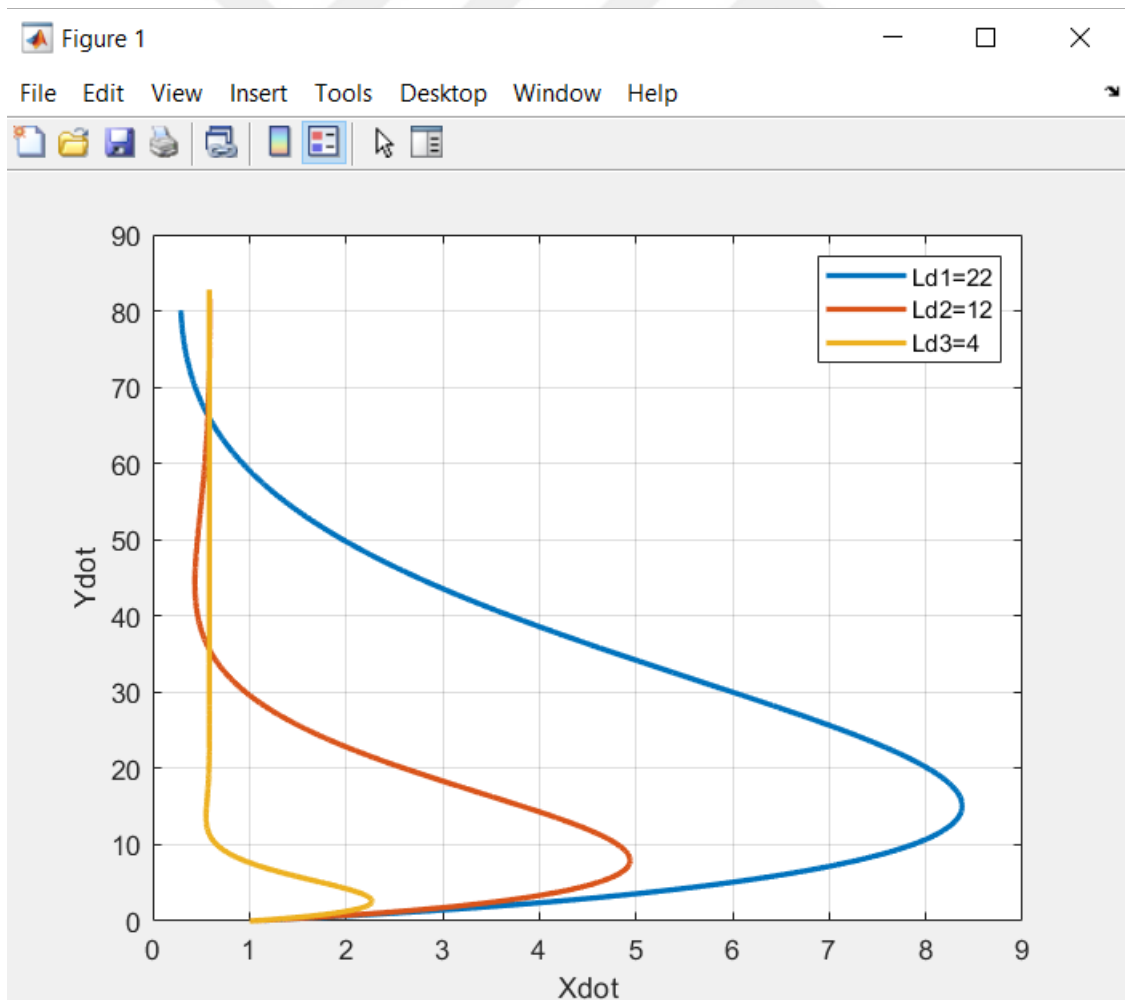


Figure 4.20: The Bicycle ($U=8.334\text{m/s}$), $L_d= (4, 12, 22\text{m})$ When It's Following the Y-Axis

It turns out that the maximum speed that a bicycle can reach is 24 meters/sec ($U=24\text{m/s}$) (86.4km/h), $L_d = (4, 12, 22\text{m})$ and after that the controller crashed

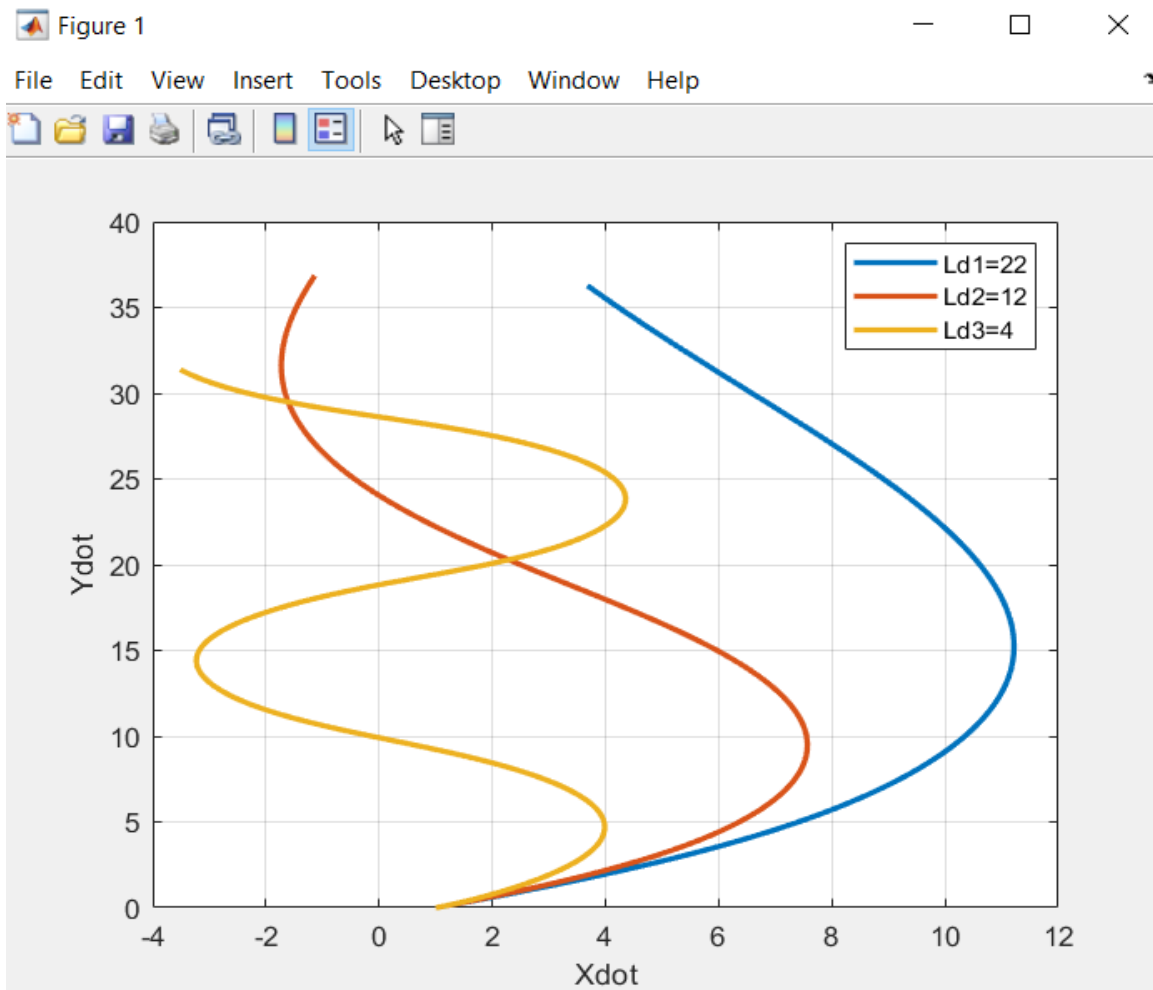
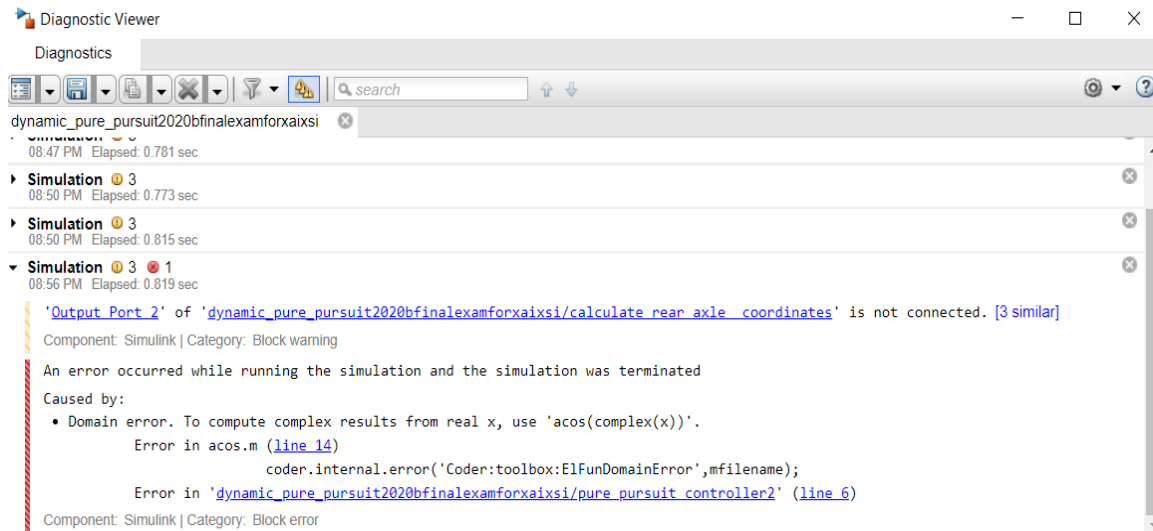


Figure 4.21: The Bicycle ($U=24\text{m/s}$), $L_d = (4, 12, 22\text{m})$ When It's Following the Y-Axis

4.2.3. When the bicycle following the X-axis

Three different tracking points and three speed values are also selected. When the bicycle speed is 10 Km/h ($U=2.78\text{m/s}$), $L_d=(4,12,22\text{m})$

```
%Bicycle parameters
```

```
mass=104;
```

```
Iz=250;
```

```
ad=0.585;
```

```
bd=0.585;
```

```
Cfront=30000;
```

```
Crear=30000;
```

```
U=2.78;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

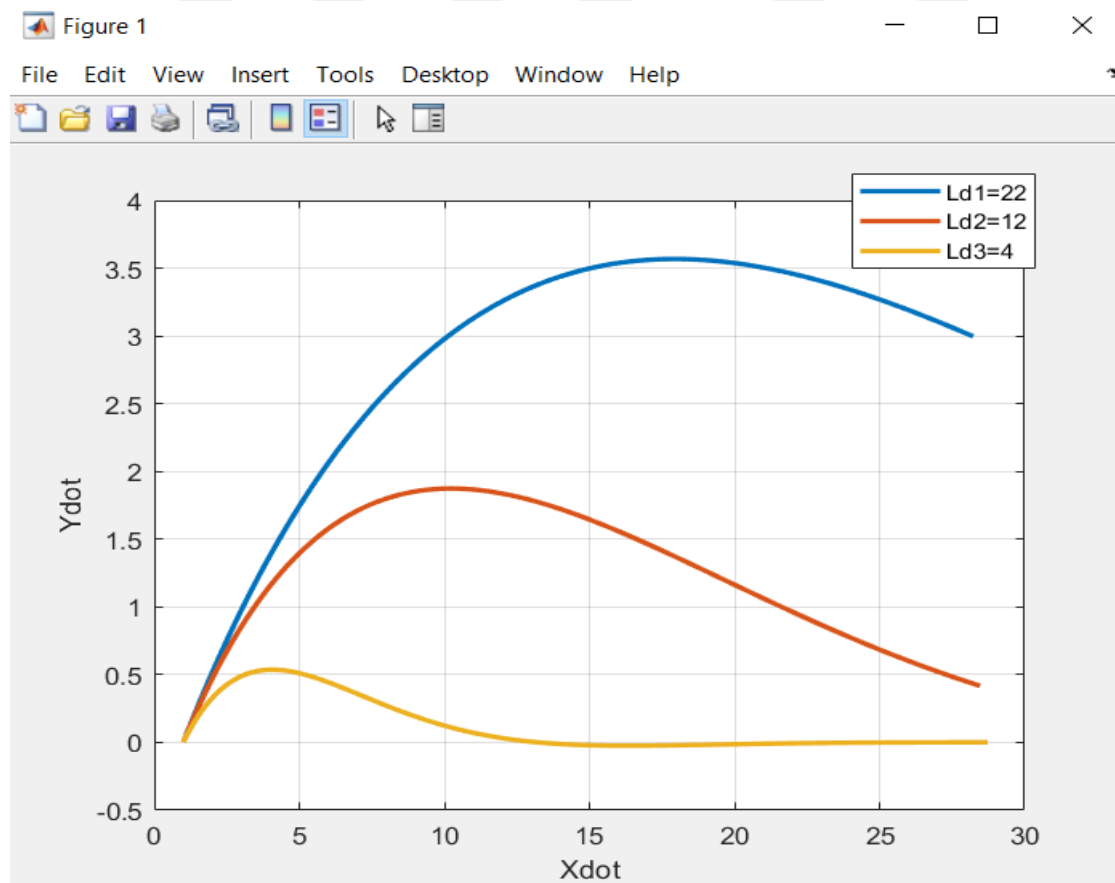


Figure 4.22: The Bicycle ($U=2.78\text{m/s}$) and $L_d = (4, 12, 22\text{m})$ When It's Following the X-Axis

When the bicycle speed is 20 Km/h ($U=5.556\text{m/s}$), $L_d=(4,12,22\text{m})$

```
%Bicycle parameters
```

```
mass=104;
```

```
Iz=250;
```

```
ad=0.585;
```

```
bd=0.585;
```

```
Cfront=30000;
```

```
Crear=30000;
```

```
U=5.556;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

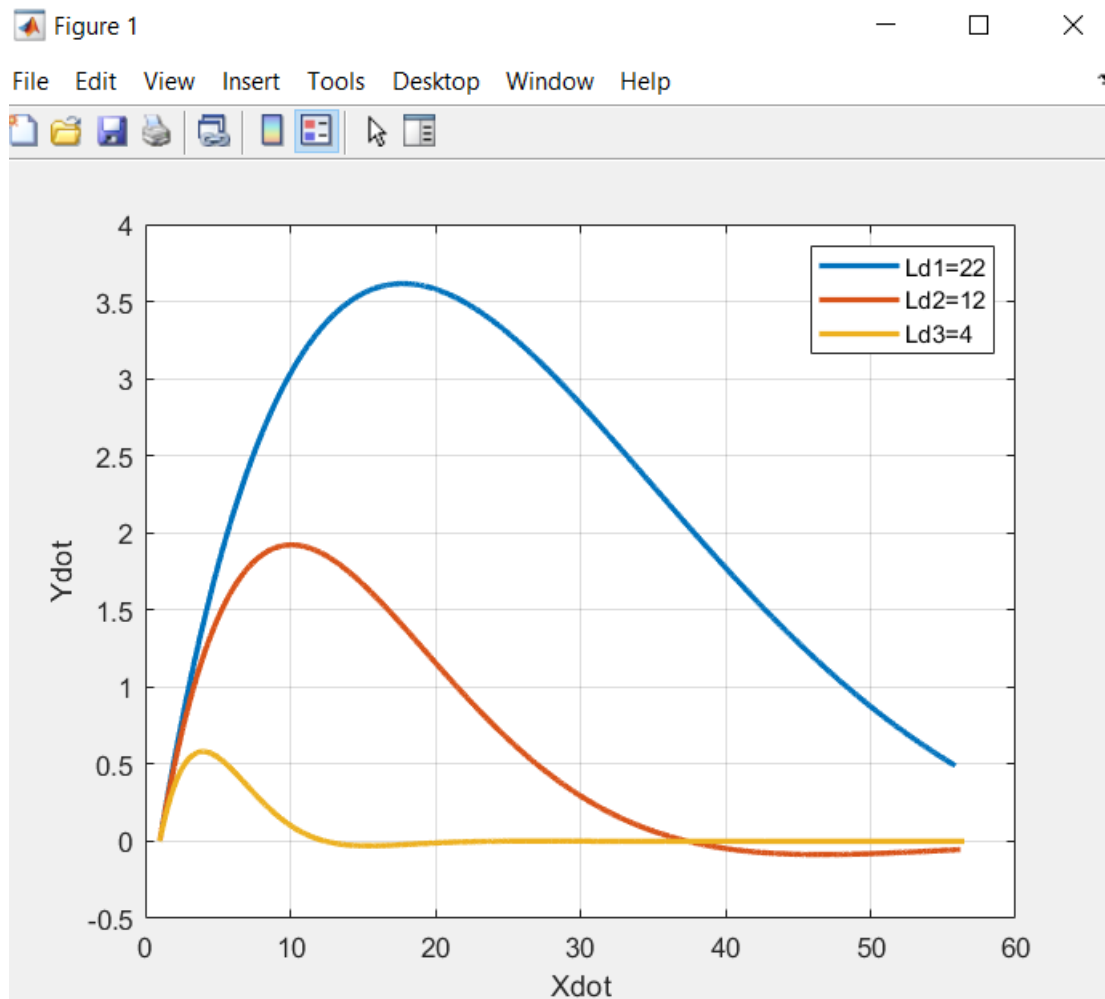


Figure 4.23: The Bicycle ($U=5.556\text{m/s}$), $L_d=(4,12,22\text{m})$ When It's Following the X-Axis

When the bicycle speed is 30 Km/h ($U=8.334\text{m/s}$), $L_d=(4,12,22\text{m})$

`%Bicycle parameters`

```
mass=104;
Iz=250;
ad=0.585;
bd=0.585;
Cfront=30000;
Crear=30000;
U=8.334;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

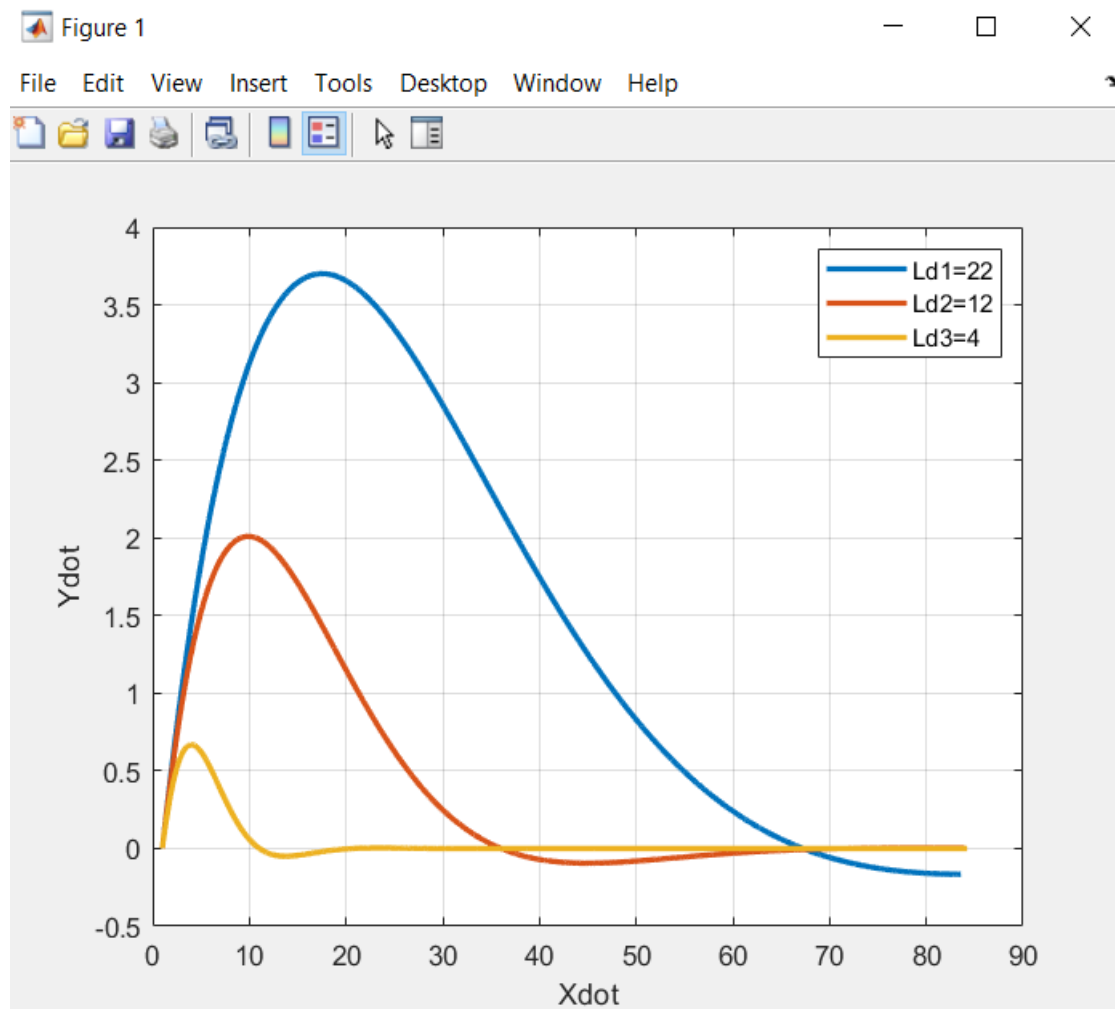


Figure 4.24: The Bicycle ($U=8.334\text{m/s}$), $L_d=(4, 12, 22\text{m})$ When It's Following the X-Axis

It turns out that the maximum speed that a bicycle can reach is 23 meters/sec ($U=23\text{m/s}$) (82.8Km/h), $L_d=(4,12,22\text{m})$ and after that the controller crashed

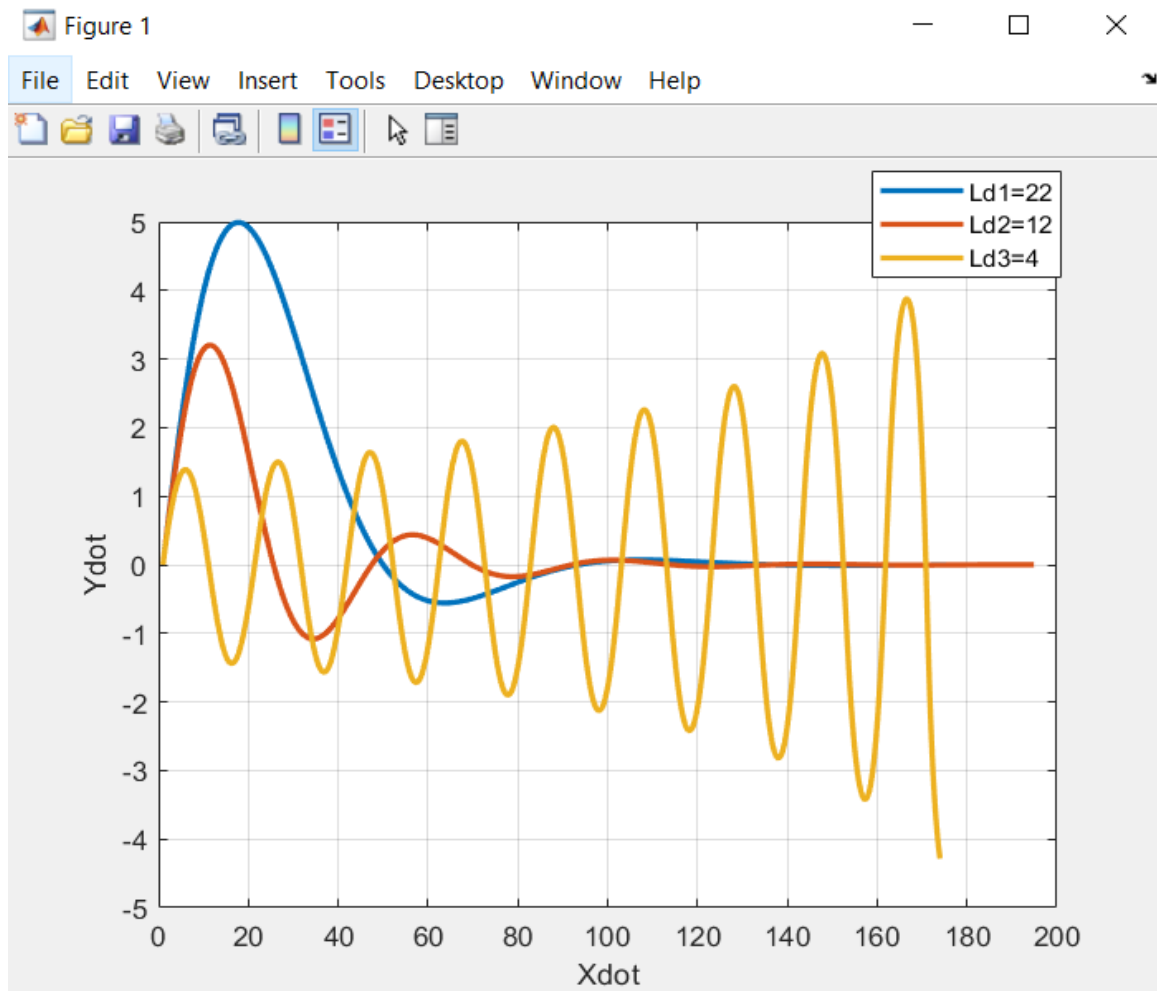
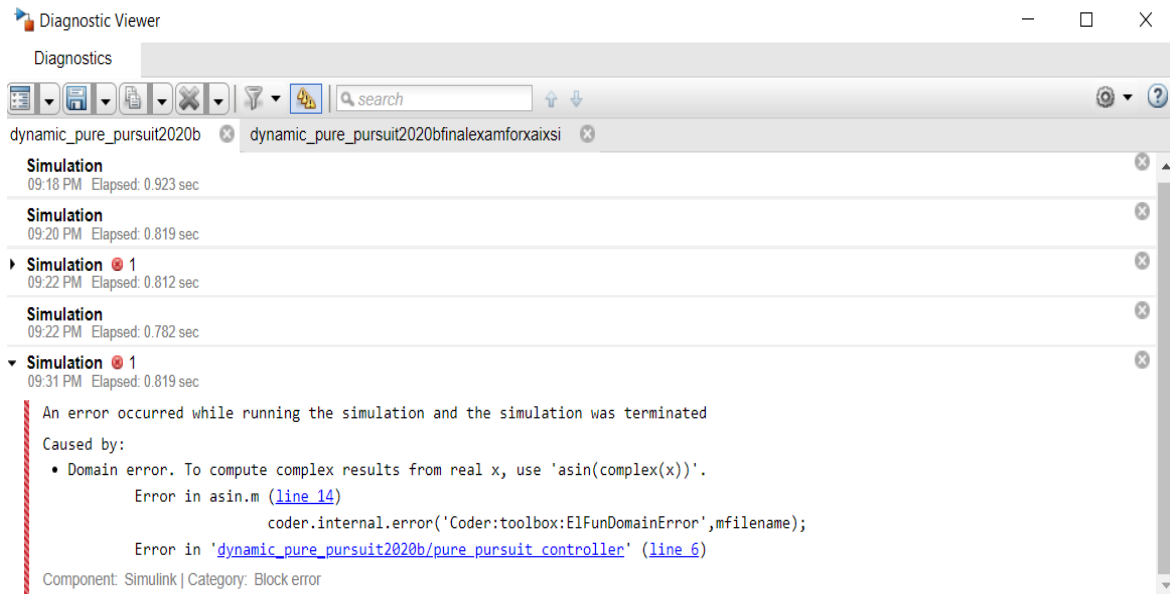


Figure 4.25: The Bicycle ($U=23\text{m/s}$), $L_d=(4, 12, 22\text{m})$ When It's Following the X-Axis

4.2.4. Truck Features:

The results of the bicycle were compared when tracking the X, Y axis with the truck and under the same conditions to notice the differences, and differences when changing the speed and mass and in this way the whole topic was covered in these points and the figure below shows the dimensions that were used for the truck.[9]

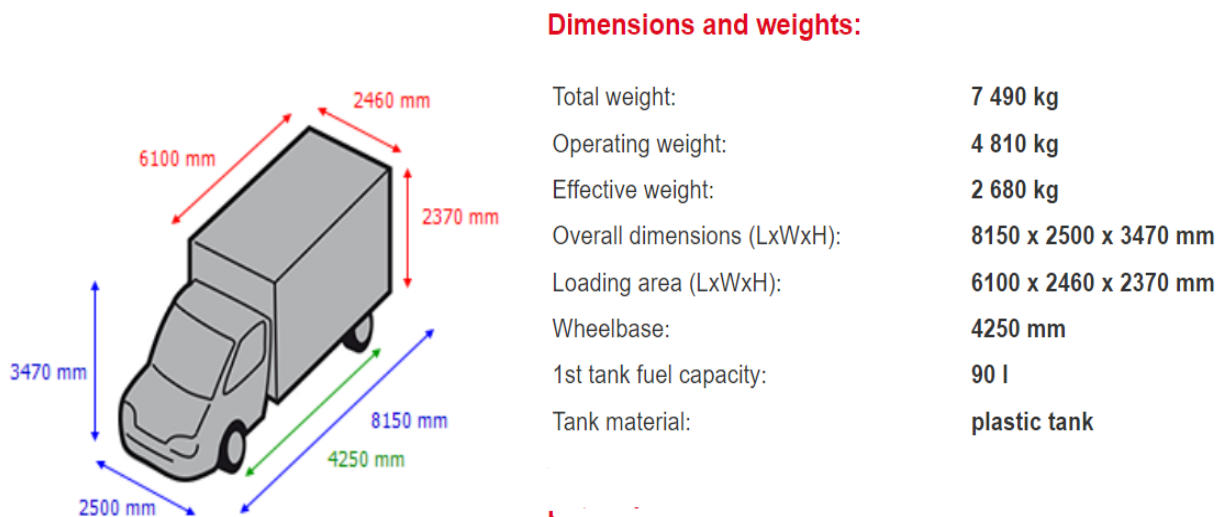


Figure 4.26: Dimensions of the truck that was used

4.2.5. When the truck following the Y-axis

The truck tracks the Y-axis, where the results were applied to three different values of speed and three values of l_d and the result was seen at each of the speeds and knowing the date of the damage for controller.

When the truck speed is 10 Km/h ($U=2.78\text{m/s}$), $L_d=(4,12,22\text{m})$

```
%truck parameters
```

```
mass=7490;
```

```
Iz=4500;
```

```
ad=2.125;
```

```
bd=2.125;
```

```
Cfront=60000;
```

```
Crear=60000;
```

```
U=2.78;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

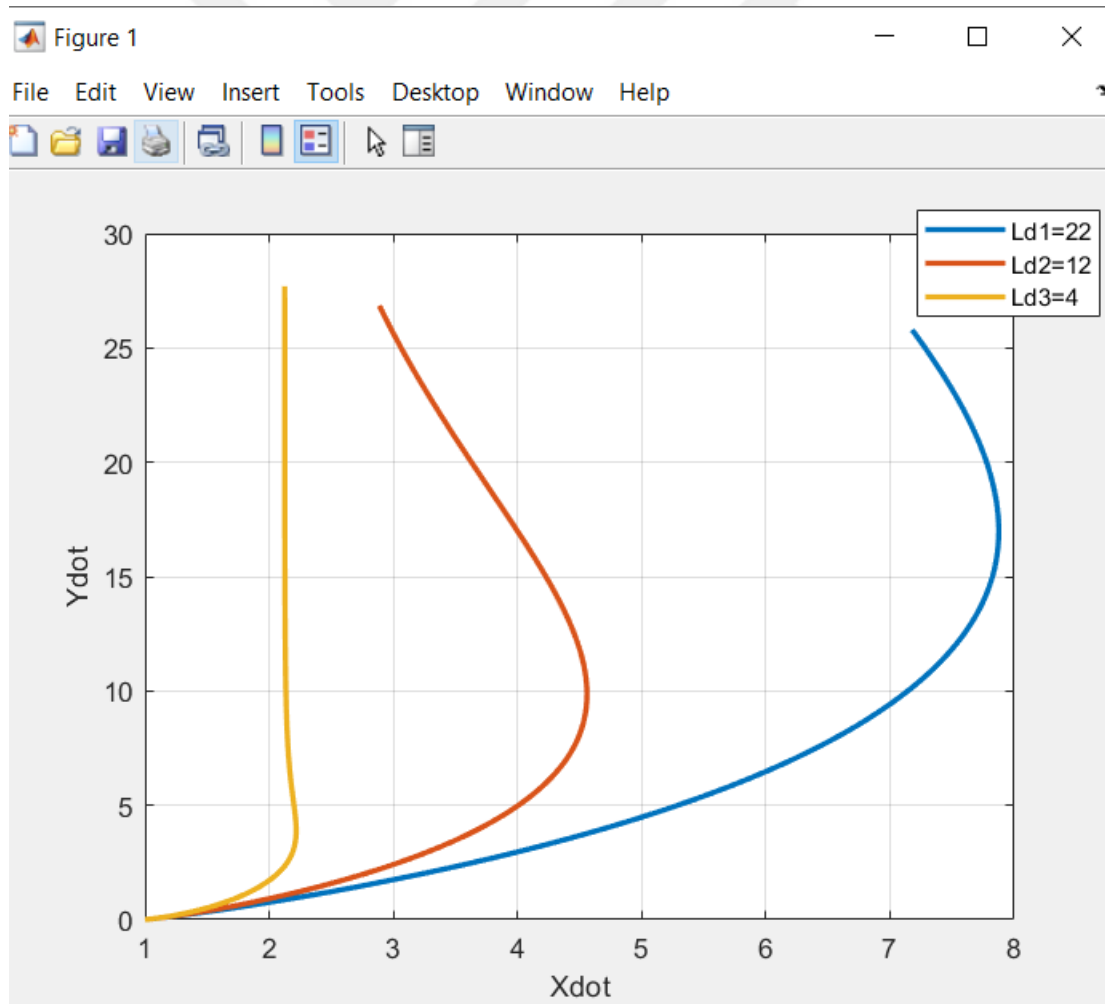


Figure 4.27: The Truck Speed Is 10 Km/h ($U=2.78\text{m/s}$), $L_d= (4, 12, 22\text{m})$ and Following the Y-Axis

When the truck speed is 20 Km/h ($U=5.556\text{m/s}$), $L_d=(4,12,22\text{m})$

```
%truck parameters
```

```
mass=7490;
```

```
Iz=4500;
```

```
ad=2.125;
```

```
bd=2.125;
```

```
Cfront=60000;
```

```
Crear=60000;
```

```
U=5.556;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

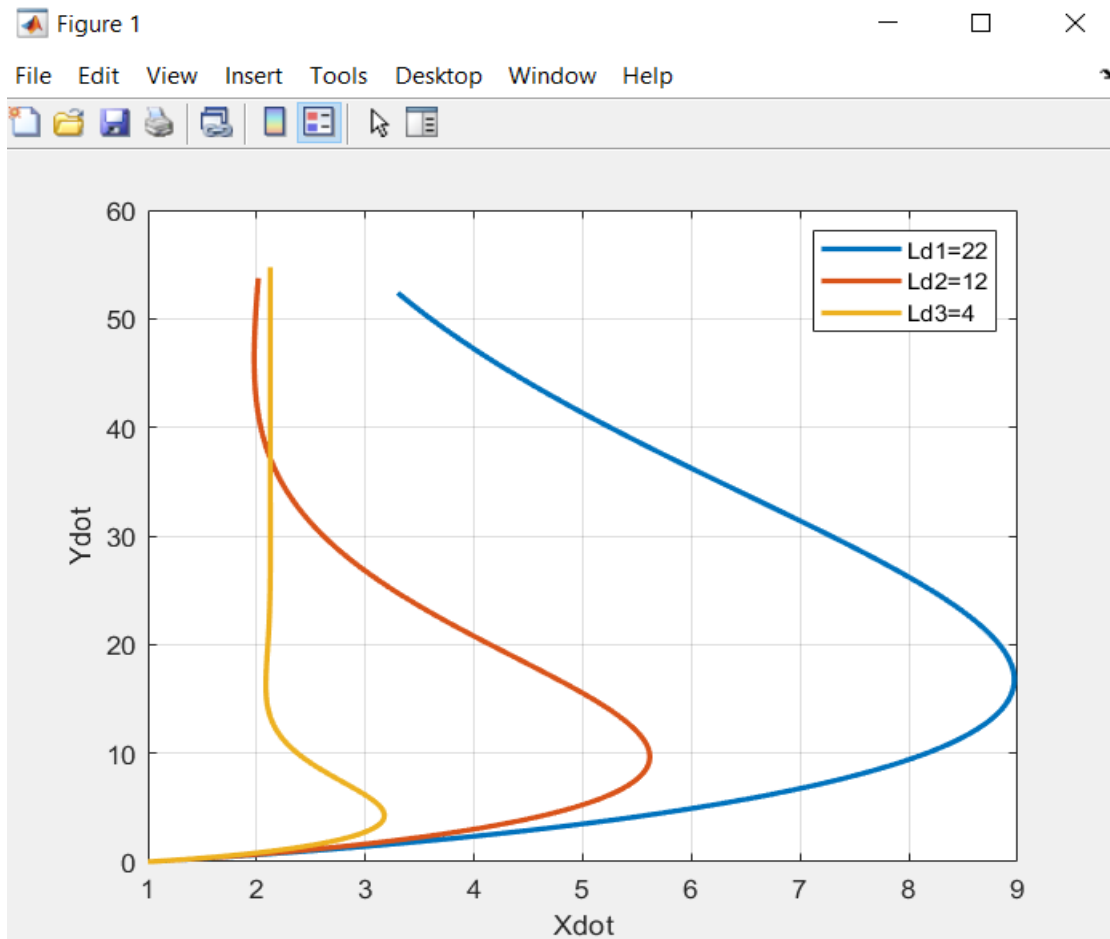


Figure 4.28: The Truck Speed Is 20 Km/h ($U=5.556\text{m/s}$), $L_d = (4, 12, 22\text{m})$ and Following the Y-Axis

When the truck speed is 30 Km/h ($U=8.334\text{m/s}$), $L_d=(4,12,22\text{m})$

```
%truck parameters
```

```
mass=7490;
```

```
Iz=4500;
```

```
ad=2.125;
```

```
bd=2.125;
```

```
Cfront=60000;
```

```
Crear=60000;
```

```
U=8.334;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

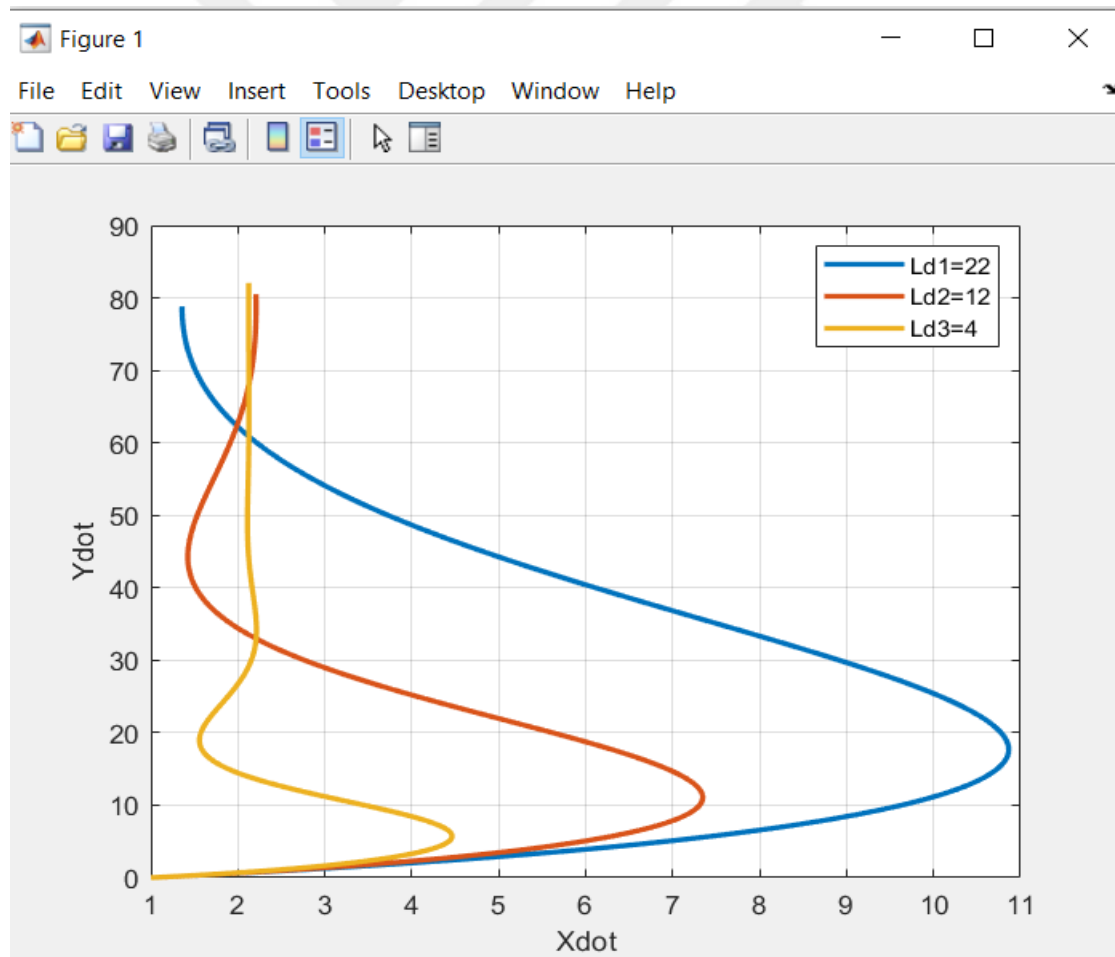


Figure 4.29: The Truck Speed is 30 Km/h ($U=8.334\text{m/s}$), $L_d= (4, 12, 22\text{m})$ And Following the Y-Axis

The controller will crash according when the ($U=11\text{m/s}$)(39.6Km/h), $L_d=(4,12,22\text{m})$

A rapid breakdown of the control occurred at low speed, as a result of the large mass negatively affecting the tracking of the cutoff, which led to the control stopping at low speed. This indicates that the controller works under conditions and conditions suitable for small blocks, not large ones, and it is effective for relatively few masses.

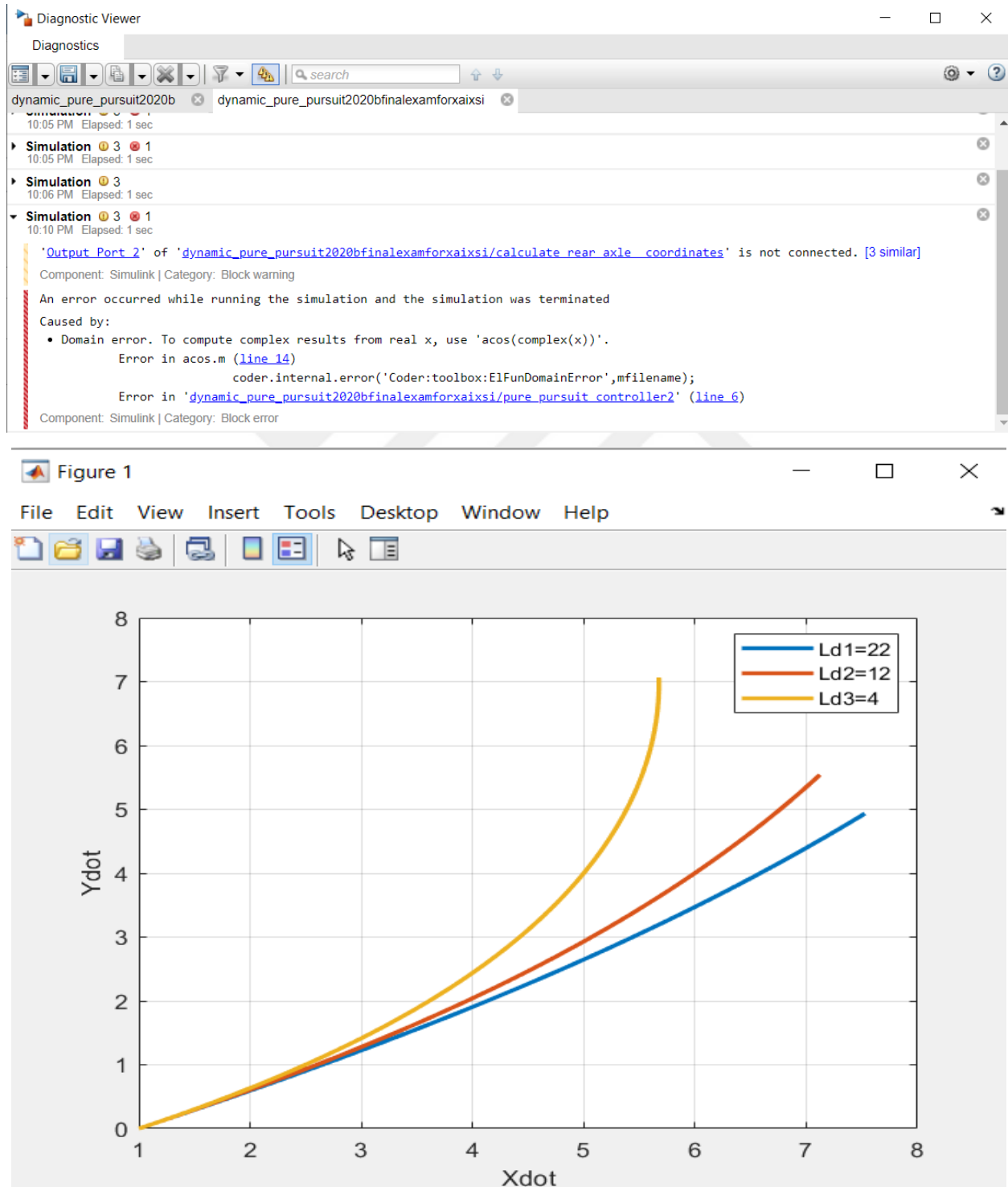


Figure 4.30: The Truck Speed Is ($U=11\text{m/s}$) (39.6Km/h), $L_d= (4, 12, 22\text{m})$ and Following the Y-Axis

4.2.6. When the truck following the X-axis

The results were also obtained for the truck when tracking the X-axis, as it was done at three different speeds and three different tracking points, but the most important factor is the mass and the set of figures below show that

When the truck speed is 10 Km/h ($U=2.78\text{m/s}$), $L_d=(4,12,22)$

```
%truck parameters
```

```
mass=7490;
```

```
Iz=4500;
```

```
ad=2.125;
```

```
bd=2.125;
```

```
Cfront=60000;
```

```
Crear=60000;
```

```
U=2.78;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
    (ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

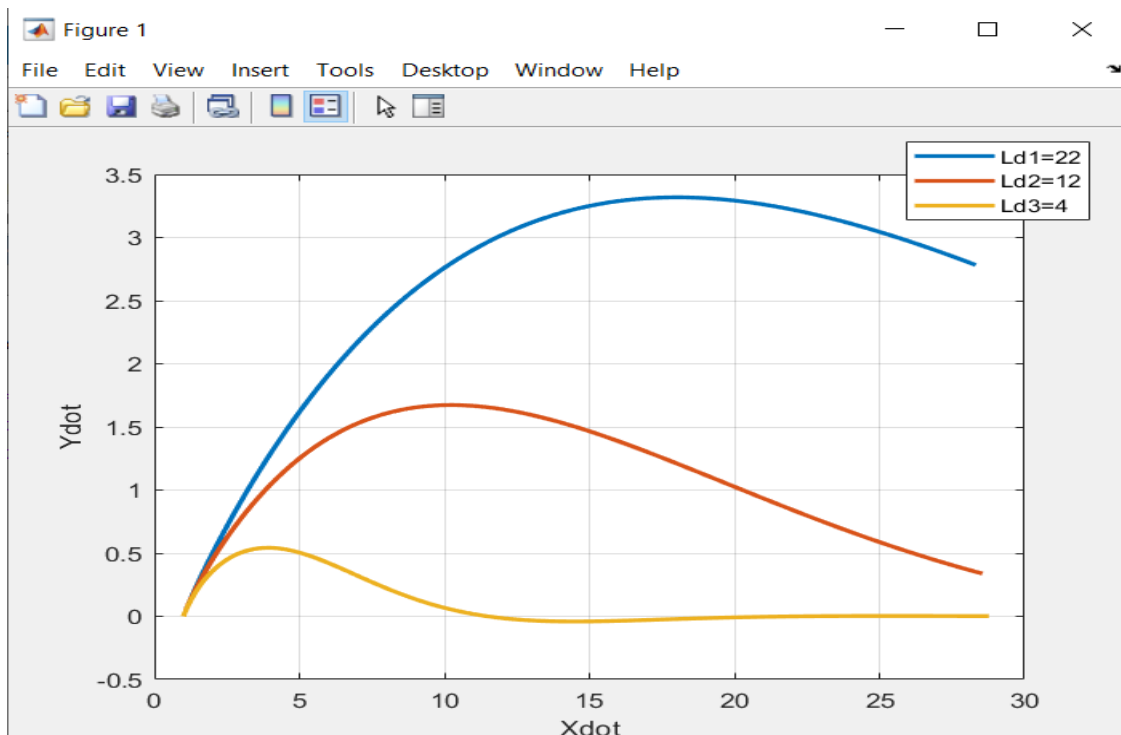


Figure 4.31: The Truck Speed Is 10 Km/h ($U=2.78\text{m/s}$), $L_d=(4, 12,22\text{m})$ and Following the X-Axis

When the truck speed is 20 Km/h ($U=5.556\text{m/s}$), $L_d=(4,12,22)$

```
%truck parameters
```

```
mass=7490;
```

```
Iz=4500;
```

```
ad=2.125;
```

```
bd=2.125;
```

```
Cfront=60000;
```

```
Crear=60000;
```

```
U=5.556;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

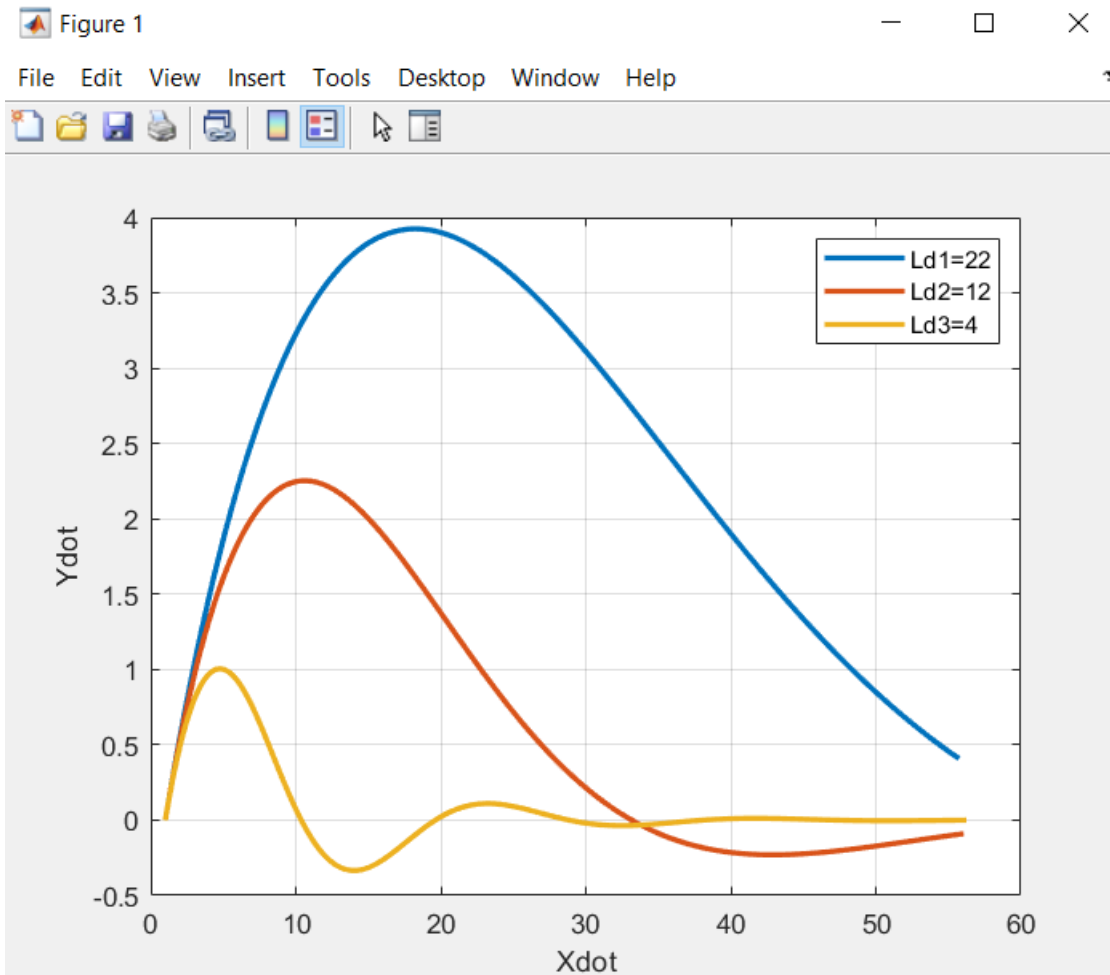


Figure 4.32: The Truck Speed is 20 Km/h ($U=5.556\text{m/s}$), $L_d= (4, 12, 22\text{m})$ And Following the X-Axis

When the truck speed is 30 Km/h ($U=8.334\text{m/s}$), $L_d=(4,12,22\text{m})$

```
%truck parameters
```

```
mass=7490;
```

```
Iz=4500;
```

```
ad=2.125;
```

```
bd=2.125;
```

```
Cfront=60000;
```

```
Crear=60000;
```

```
U=8.334;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

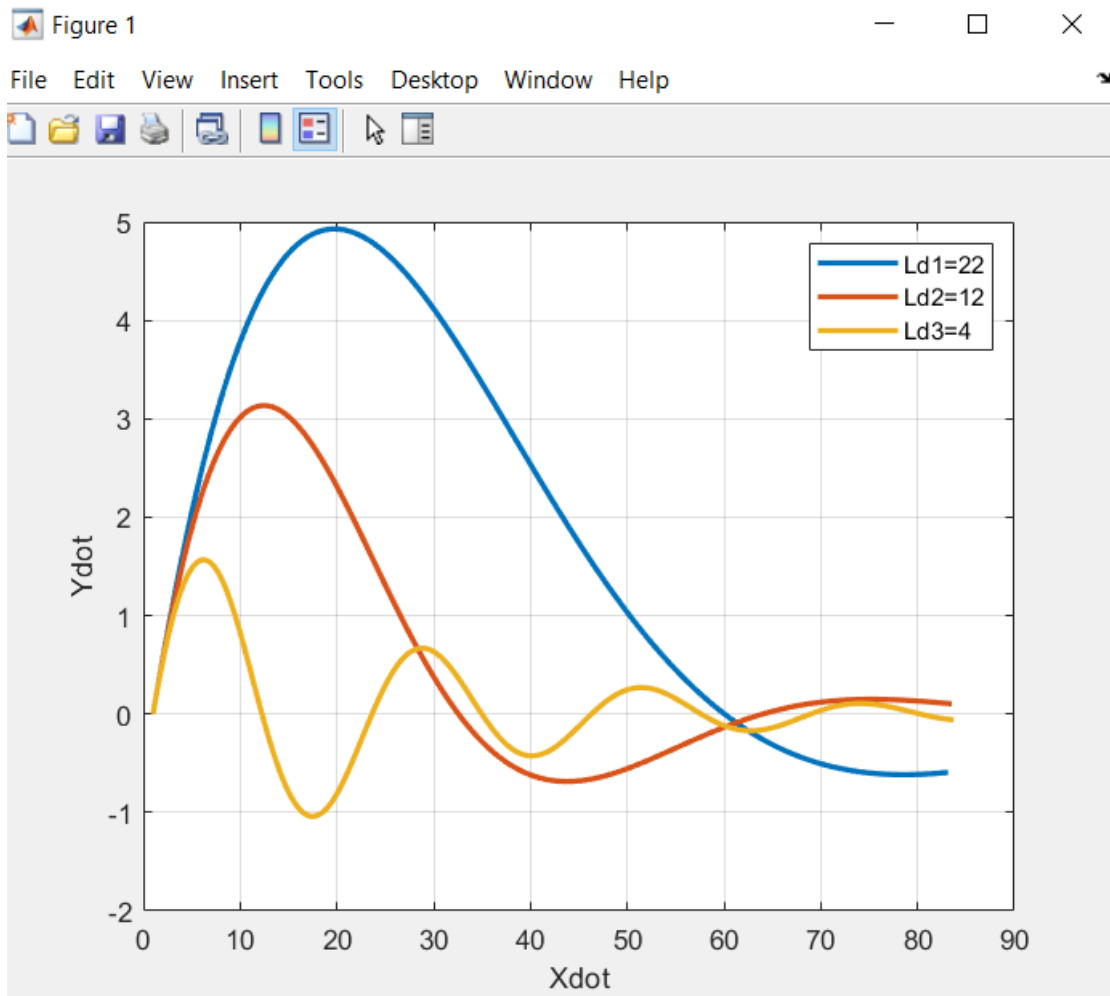


Figure 4.33: The Truck Speed is 30 Km/h ($U=8.334\text{m/s}$), $L_d= (4, 12, 22\text{m})$ and Following the X-Axis

The controller will crash according when the ($U=13\text{m/s}$)(46.8Km/h), $L_d=(4,12,22\text{m})$

A rapid breakdown of the control occurred at low speed, as a result of the large mass negatively affecting the tracking of the cutoff, which led to the control stopping at low speed. This indicates that the controller works under conditions and conditions suitable for small blocks, not large ones, and it is effective for relatively few masses. The control is more effective when the X-axis is tracked, as there is a difference in the end speed at which the control will be damaged, and this is what the figures shows below.

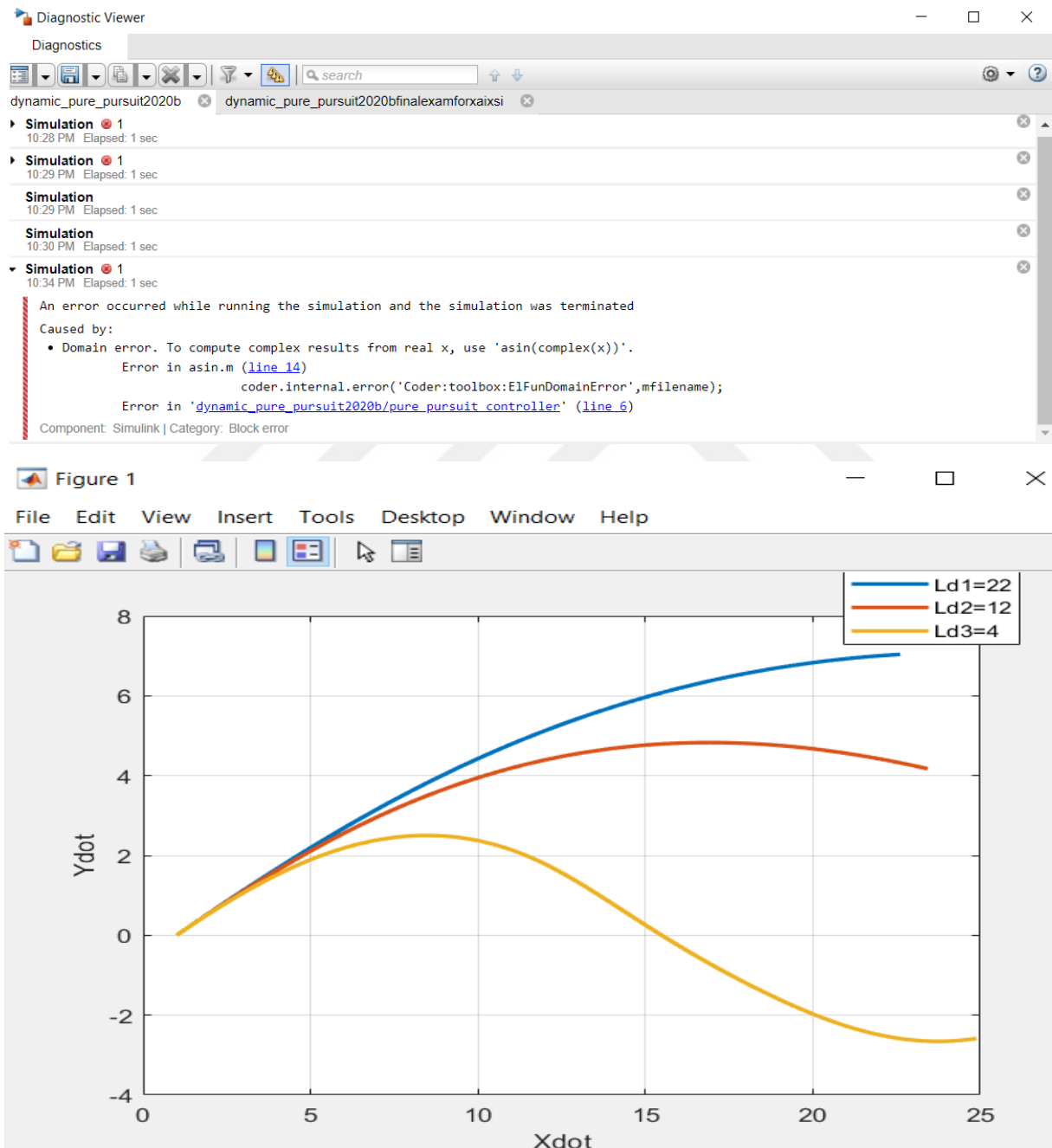


Figure 4.34: The Truck Speed Is ($U=13\text{m/s}$), (46.8Km/h), $L_d= (4, 12, 22\text{m})$ and Following the X-Axis

4.3. CHANGING THE POSITION OF CENTER OF GRAVITY FOR PURE PURSUIT

In the last section, the position of the C.G. was changed, as the study was done when the vehicle followed the Y-axis as well as the X-axis, and then it was studied how the looking ahead distance changes with the change of C.G. Knowing that a , b has been changed in order to change the C.G. By changing the code and watching the result.

4.3.1. Making the C.G. Goes forward (near the front wheel) when it's following Y-axis:

```
%The parameters of the vehicle
```

```
clear all
mass=1250;
Iz=1550;
ad=0.9;
bd=1.3;
Cfront=50000;
Crear=50000;
U=15;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

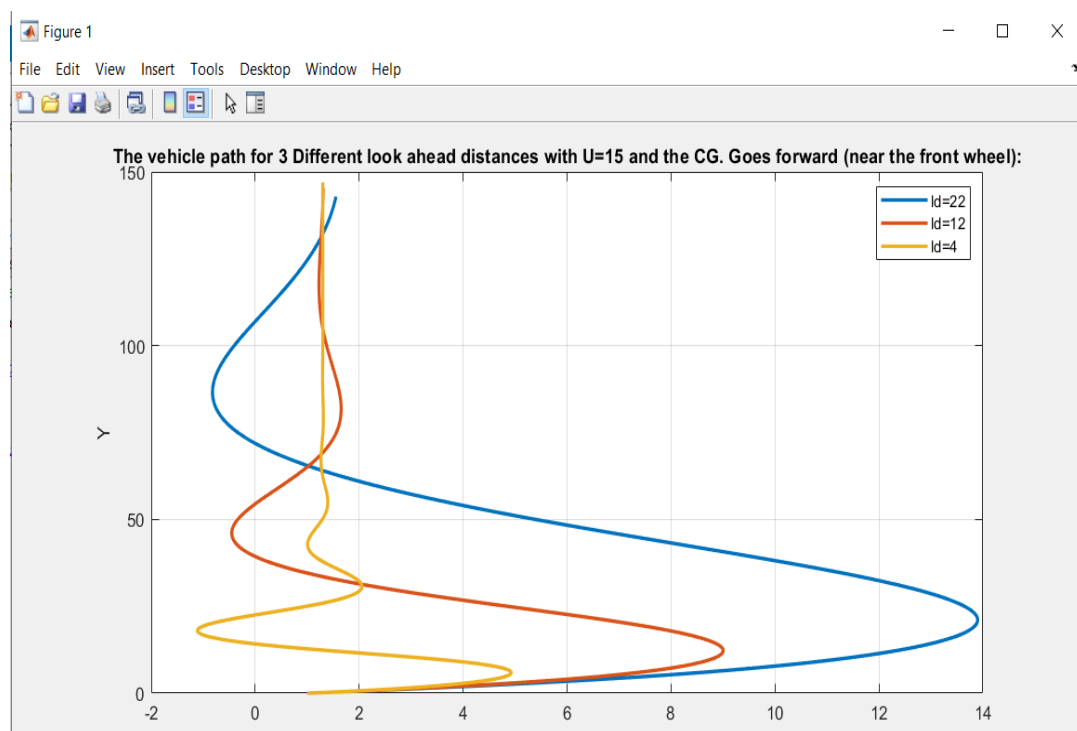


Figure 4.35: The Vehicle Path for 3 Different Look Ahead Distance with ($U=15\text{m/s}$) and the C.G Goes Forward

4.3.2. Making the C.G. Goes backwards (near the rear wheel) For following Y-axis:

%The parameters of the vehicle

```
clear all
```

```
mass=1250;
```

```
Iz=1550;
```

```
L=2.2;
```

```
ad=1.3;
```

```
bd=0.9;
```

```
Cfront=50000;
```

```
Crear=50000;
```

```
U=15;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

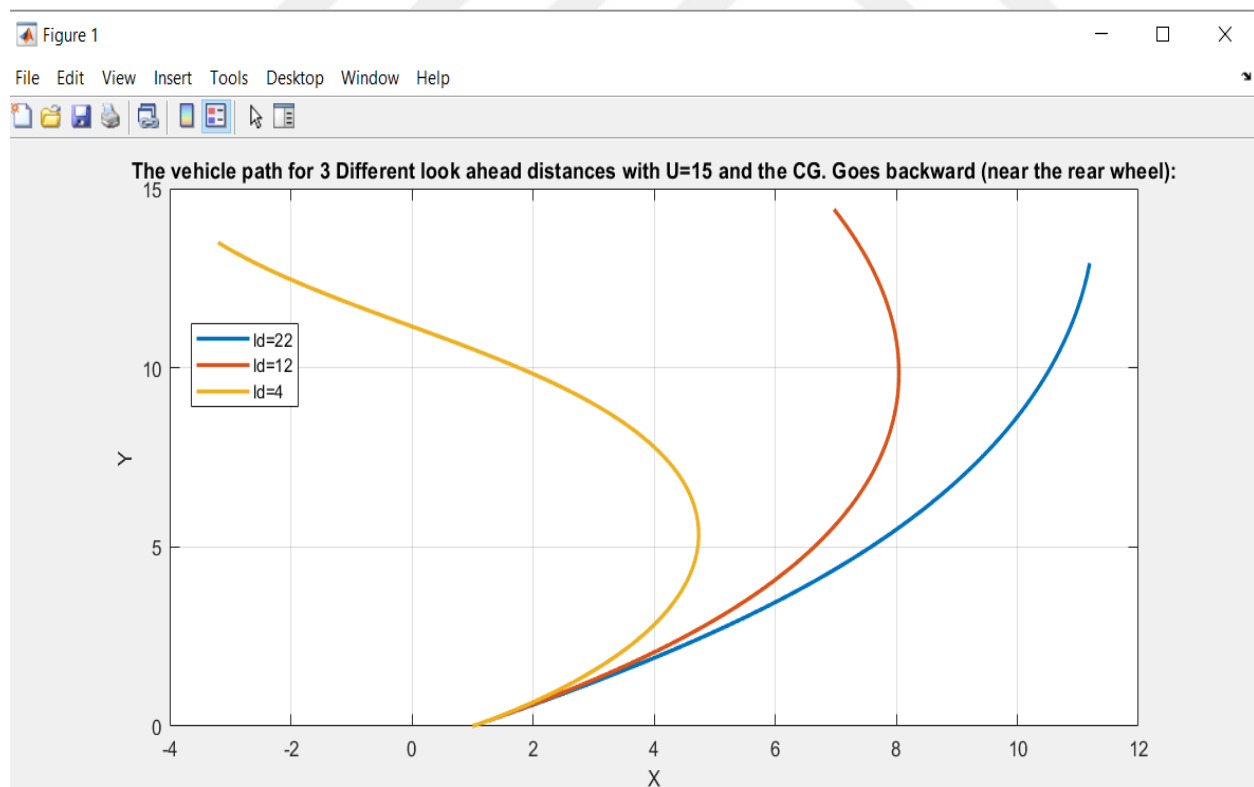


Figure 4.36: The Vehicle Path for 3 Different Look Ahead Distance with (U=15m/s) and the C.G Goes Backward

The controller was crashed according the changing the place of the center of gravity, so the speed of vehicle was changed to 10 to see the effort of changing the center of gravity.

%The parameters of the vehicle

```
clear all
mass=1250;
Iz=1550;
L=2.2;
ad=1.3;
bd=0.9;
Cfront=50000;
Crear=50000;
U=10;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

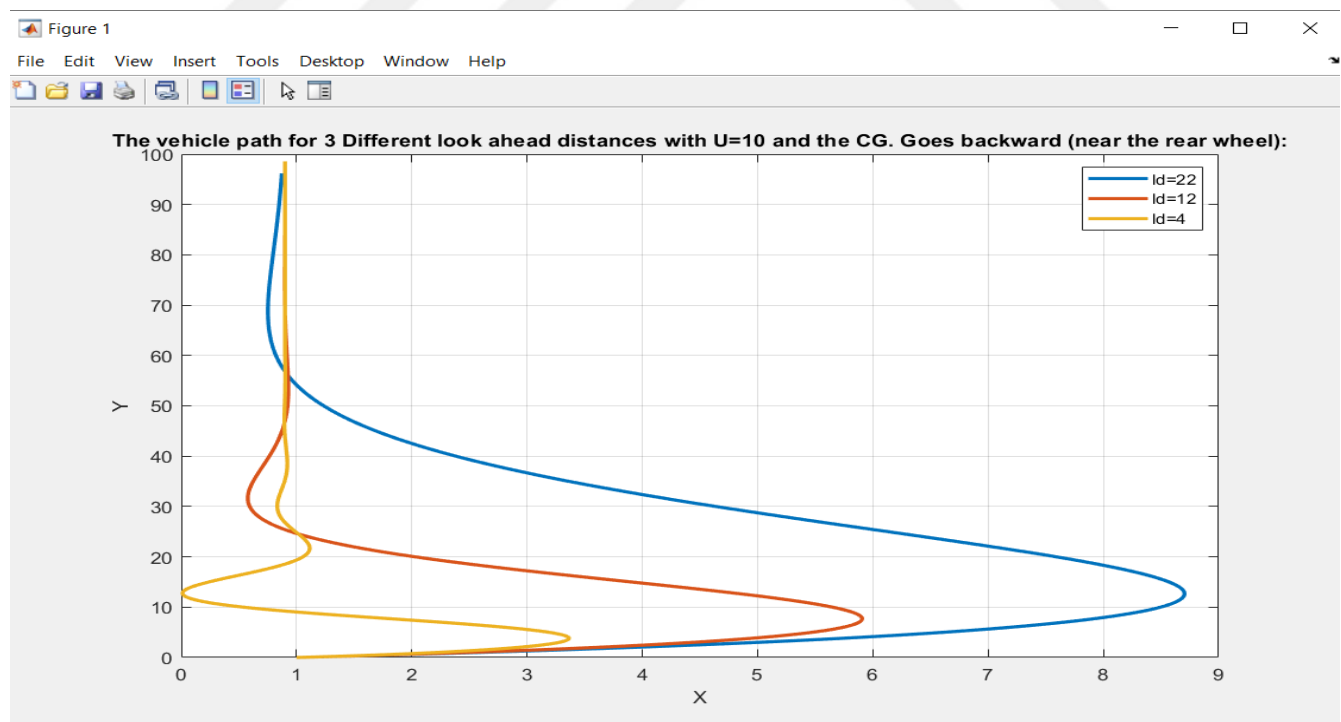


Figure 4.37: The Vehicle Path for 3 Different Look Ahead Distance with (U=10m/s) and the C.G Goes Backward

Moving the position of the center of gravity led to an increase in the overshoot in the control result with a difference in L_d but it did not change, significantly, when you moved the center of gravity to become near the front wheel, as there is no significant difference when the vehicle speed is equal to 15(m/s), but when changing the location of the center of gravity To become close to the rear wheel, this led to a crash of the controller due to the increase in the response speed and the increased instability of the wheels when they approach the rear wheel, knowing that, another method was later tried using simulation and nothing happened to the controller, but the speed was changed in order to notice the difference, it was also found that the change that occurred is very simple and was only in the overshoot when the speed was adjusted to 10 (m/s). And when the overshoot is increase when the center of gravity goes to the center of the rear wheel but it's still unstable and have fast response.



4.3.3. Making the CG. Goes forward (near the front wheel) For following X-axis

The explanation was given below, in the case of the car tracking the x-axis with the change of the center of the Gravity and it was presented to make a quick comparison, if the car was tracking the x-axis, but with the change of CG and it was only found that the response is a little faster and that the overshoot increases a little and the vehicle remains unstable.

%The parameters of the vehicle

```
clear all
mass=1250;
Iz=1550;
ad=0.9;
bd=1.3;
Cfront=50000;
Crear=50000;
U=15;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

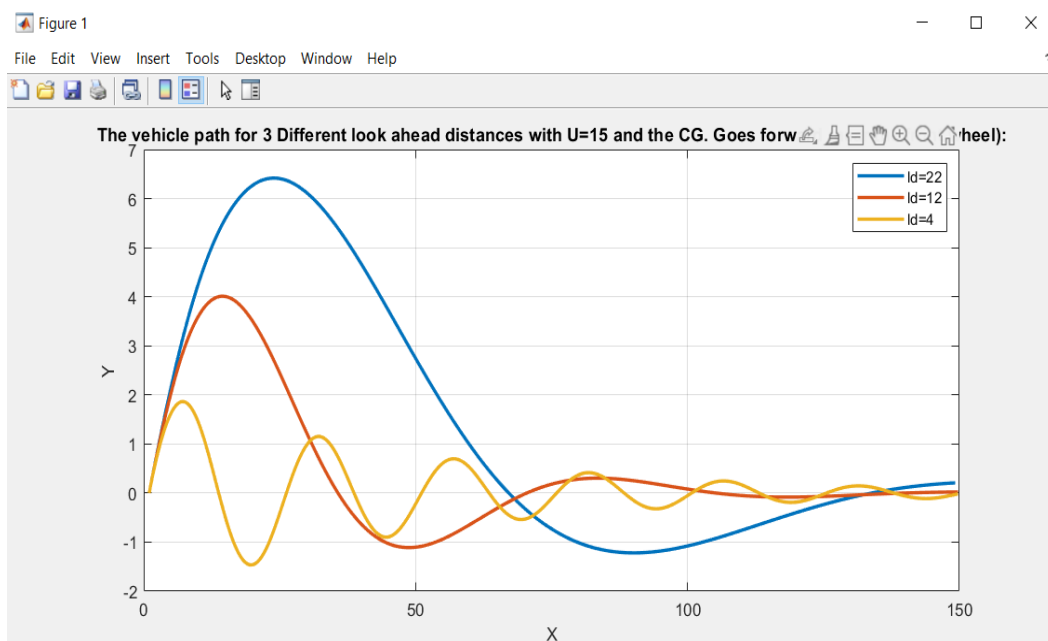


Figure 4.38: The Vehicle Path for 3 Different Look Ahead Distance with (U=15m/s) and the C.G at the Rear Wheel

4.3.4. Making the CG. Goes backwards (near the rear wheel) For following X-axis:

```
%The parameters of the vehicle
clear all
mass=1250;
Iz=1550;
L=2.2;
ad=1.3;
bd=0.9;
Cfront=50000;
Crear=50000;
U=15;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

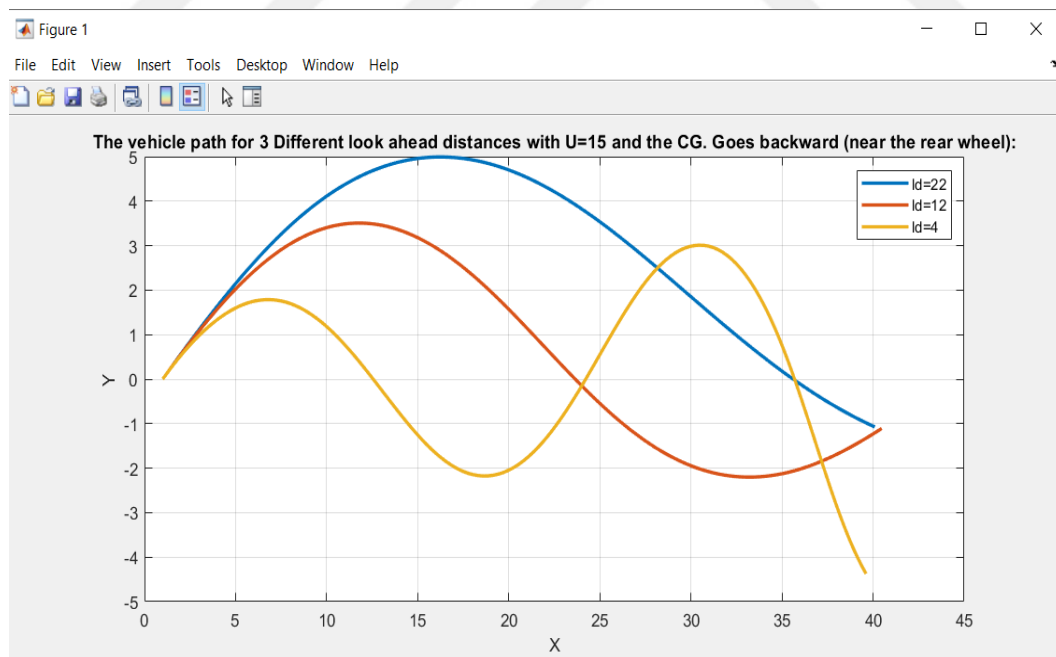


Figure 4.39: The Vehicle Path for 3 Different Look Ahead Distance with (U=15m/s) and the C.G Goes Backward

As it was previously explained, there is no significant effect, but when the center of gravity is changed to become close to the rear wheel, a crash for the controller occurs.

4.4. CHANGING THE CENTER OF GRAVITY TO THE FRONT WHEEL FOR STANLEY MODEL

The position of the center of Gravity has been changed to be fully applicable to the front wheel of the vehicle by changing the value of a and b to become the value of a very small and the results were observed at three different speeds, and the results were obtained as shown in the figures below by changing the code that was used. The speeds was used is (U=3, 8, 13 m/s)

Stanley model with changing the center of gravity when (U=3m/s):

`%The parameters of the vehicle`

```
clear all
```

```
mass=1250;
```

```
Iz=1550;
```

```
ad=0.2;
```

```
bd=2;
```

```
Cfront=50000;
```

```
Crear=50000;
```

```
U=3;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-  
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

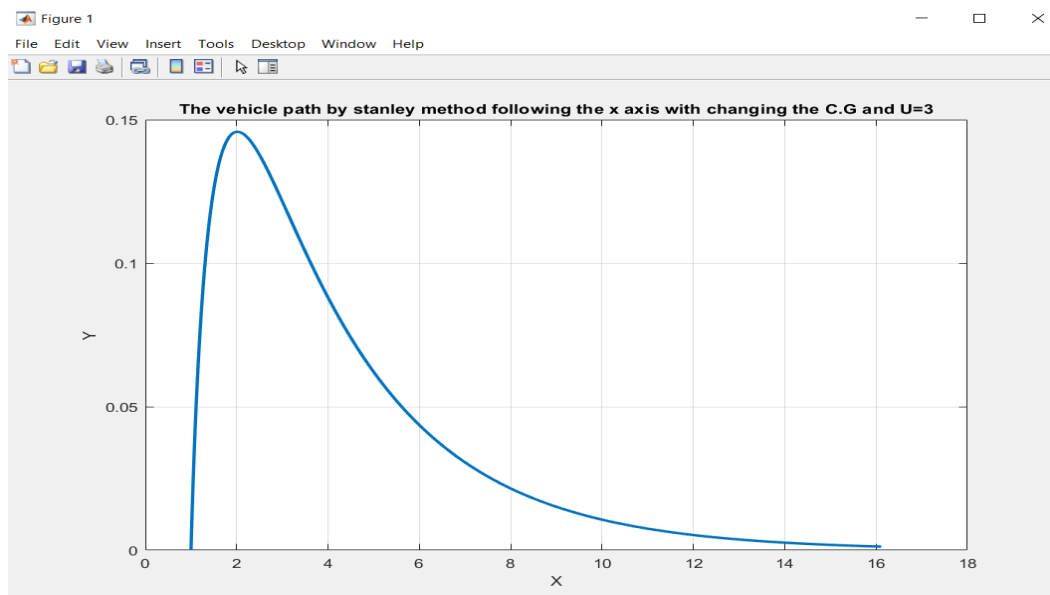


Figure 4.40: Stanley Model with Changing the Center of Gravity When (U=3m/s)

Stanley model with changing the center of gravity when (U=8m/s):

%The parameters of the vehicle

```
clear all
```

```
mass=1250;
```

```
Iz=1550;
```

```
ad=0.2;
```

```
bd=2;
```

```
Cfront=50000;
```

```
Crear=50000;
```

```
U=8;
```

```
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
```

```
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
```

```
B=[Cfront/mass;ad*Cfront/Iz];
```

```
C=eye(2);
```

```
D=[0;0];
```

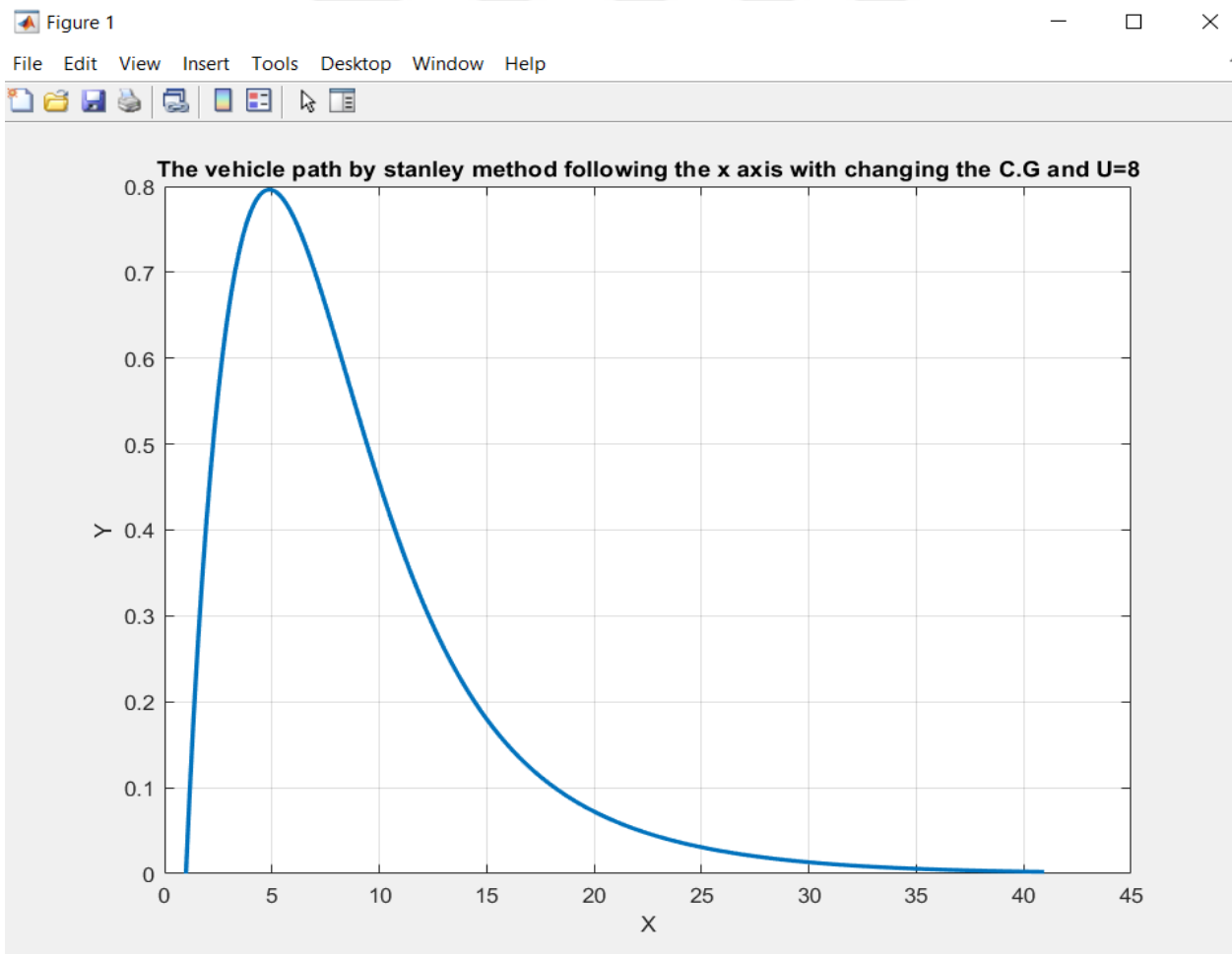


Figure 4.41: Stanley Model with Changing the Center of Gravity When (U=8m/s)

Stanley model with changing the center of gravity when (U=13m/s):

%The parameters of the vehicle

```
clear all
mass=1250;
Iz=1550;
ad=0.2;
bd=2;
Cfront=50000;
Crear=50000;
U=13;
A=[-(Cfront+Crear)/mass/U -(ad*Cfront-bd*Crear)/U/mass-U;-
(ad*Cfront-bd*Crear)/U/Iz -(ad^2*Cfront+bd^2*Crear)/U/Iz];
B=[Cfront/mass;ad*Cfront/Iz];
C=eye(2);
D=[0;0];
```

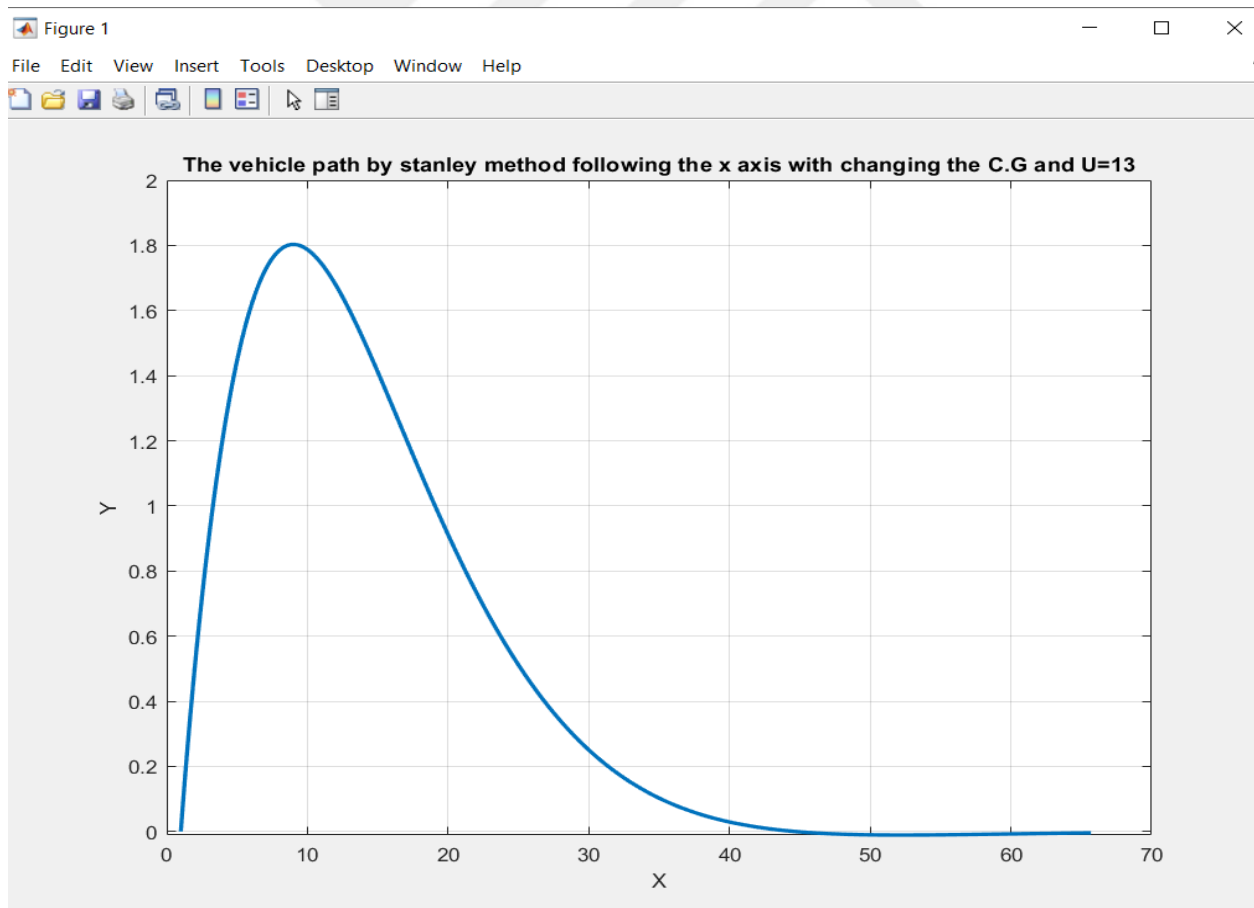


Figure 4.42: Stanley Model with Changing the Center of Gravity When (U=13m/s)

CHAPTER 5. RESULTS AND CONCLUSION

In general, reducing L_d makes cornering faster in the case of low speeds and in the case of high speeds, and when L_d is small, it leads to an increase in oscillation and becomes unstable, while when the value of L_d increases, it increases stability and makes it smooth. There is an important point that has been concluded: When the speed rises to a certain extent the controller becomes a Not Valid Anymore.

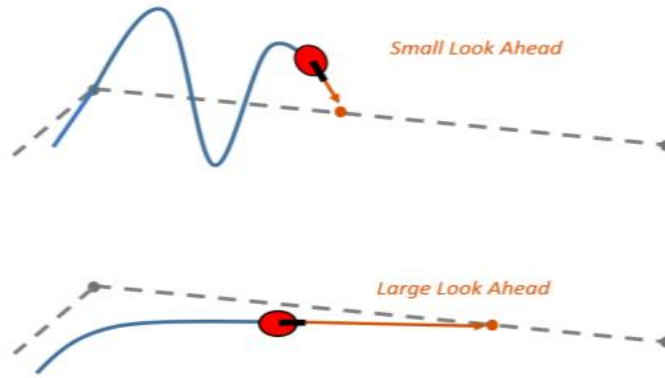


Figure 5.1: Small and Large Look Ahead Distance

For bicycle when it is following the Y-axis:

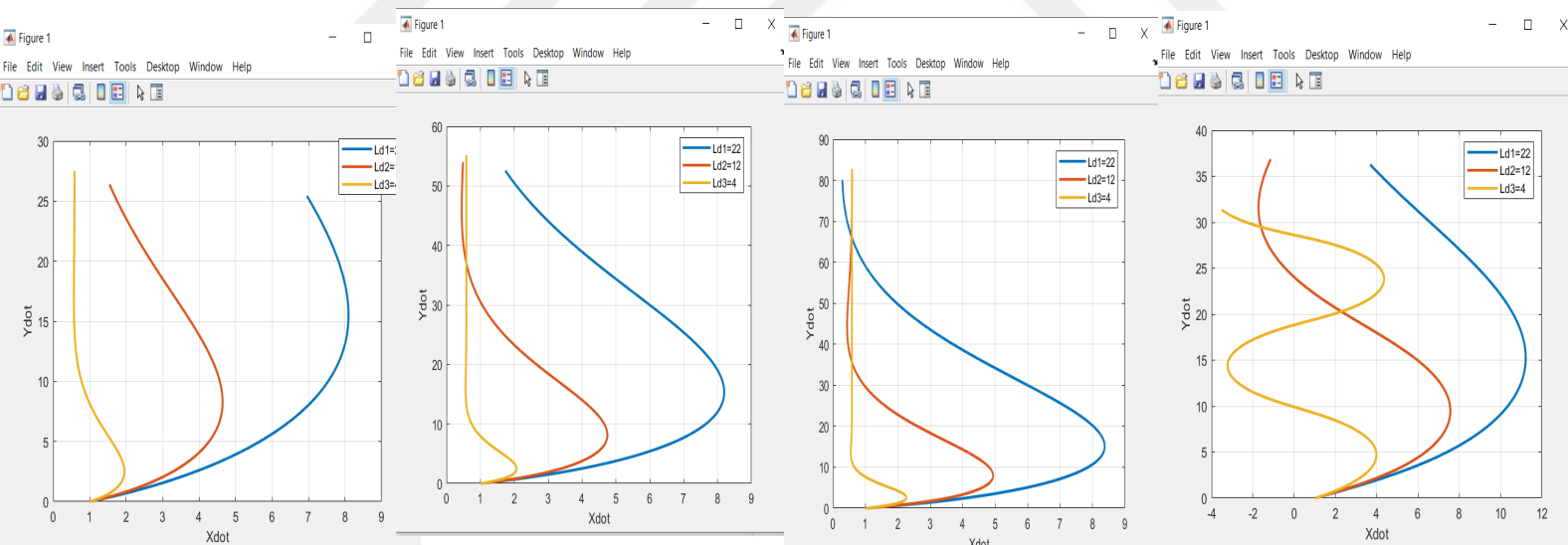


Figure 5.2: All Results for Bicycle When It's Following the Y-Axis

At the beginning of the movement when its following the Y-axis, when the bicycle begins to follow the three points, the first figure on the left shows that when $L_d = 4$ at a speed of 10 km / h, the bicycle is traveling smoothly, but will reach a stage where the bicycle begins to lose some stability at the beginning, and in comparison, when the value of L_d increases, the movement of The bike is without any problems at all, and remains stable when the values of the points being tracked are 12 and 22 In

the second two drawings, the effect of changing the speed is shown in a large way, as the more the speed increases, the oscillation of the waves increases, and therefore the stability of the bicycle will increase, but with that the bicycle does not reach the point of the cricket because, as shown in the figures, speeds are used that are not very large, and we also see clearly that the more If the mass is lower, the amplitude of the wave is small compared to if the object that follows the point has a large mass. And in the last drawing on the right, it clearly shows the total loss of control over the control, and the occurrence of oscillation in a large way, and the increase in instability at a relatively high speed. Here it appears that the control works with the highest efficiency at relatively low speeds. But it is new to note that in the last diagram on the right, the bicycle can remain in a slightly stable state when the L_d value is large.

For bicycle when it is following the X-axis:

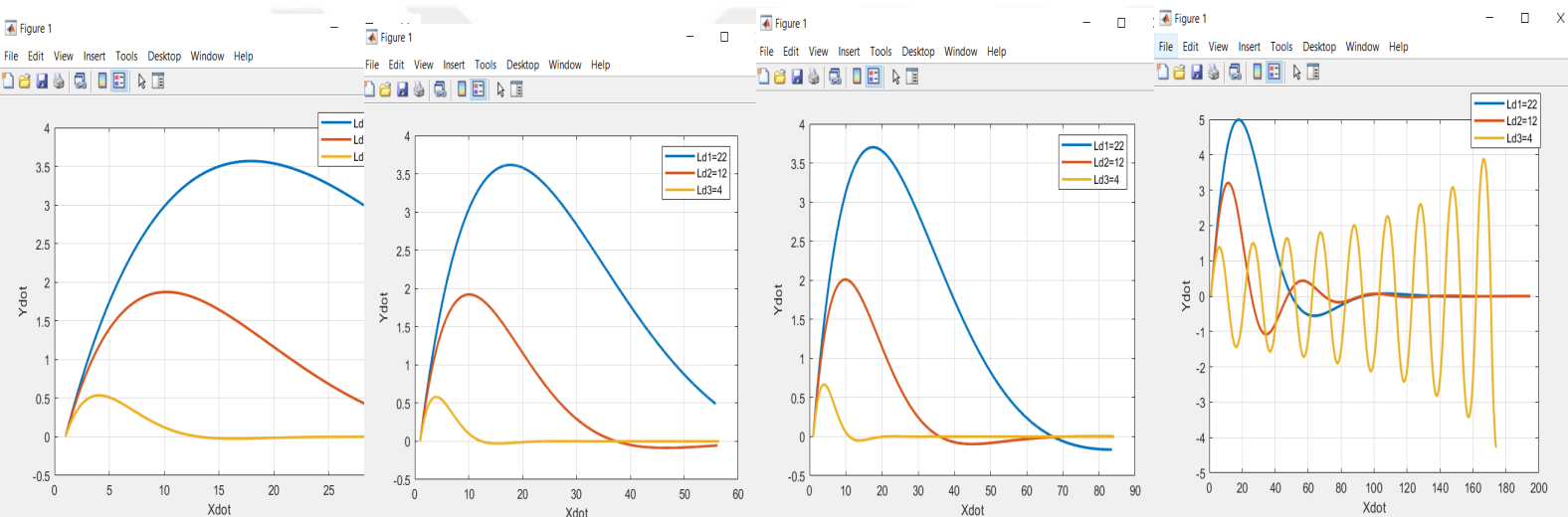


Figure 5.3: All Results for Bicycle When It's Following the X-Axis

When the bicycle follows the X-axis, there is no significant difference when the bicycle follows the Y-axis, as the first figure shows, it starts in the beginning in a smooth shape and then begins to appear when the point being tracked is few and while it takes longer for the oscillation and instability to appear in the tracking points are large, but there is an important point that the speed of the bicycle affects the shape of the waves in both cases, as was previously explained. The second and third figures show that the bicycle remains in a stable state to a certain extent and also begins to lose stability when the amplitude of the wave appears. In the last figure from the right, it shows a break in the control unit, as it was previously shown that at high speeds, the bicycle loses control and a loss of stability and stability occurs, and the oscillation appears significantly. But it is new to note that in the last diagram on the right, the bicycle can remain in a slightly stable state when the L_d value is large.

For truck when it is following the Y-axis:

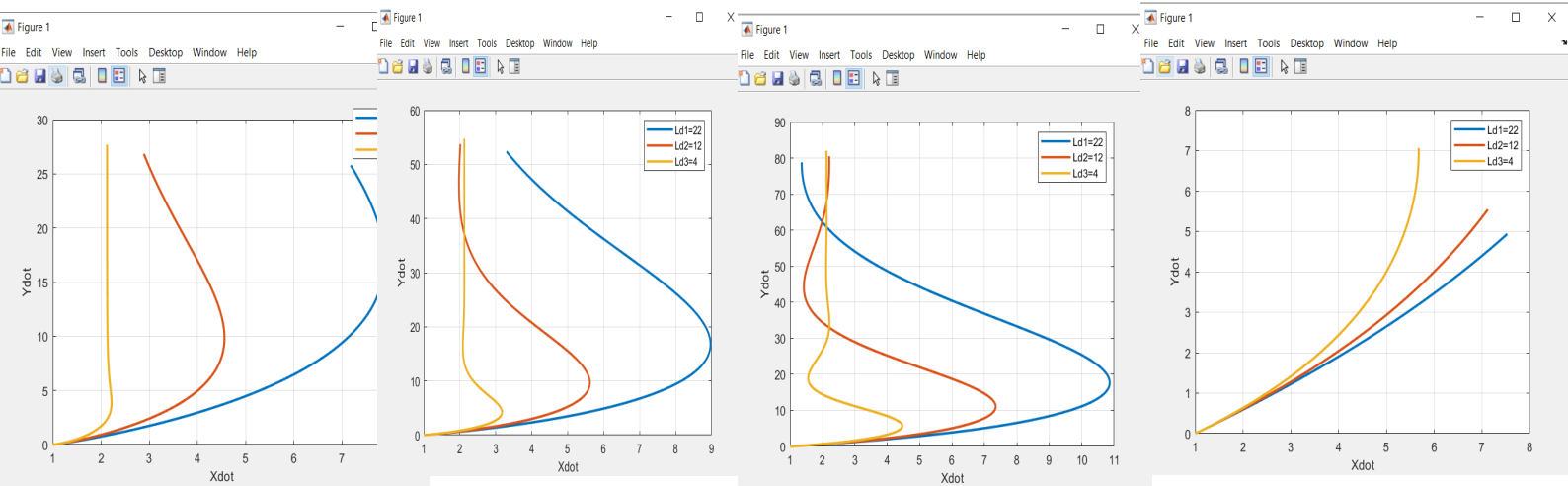


Figure 5.4: All Results for Truck When It's Following the Y-Axis

At the beginning of the truck's movement, at a speed of 10 km / h, the movement will be at all the tracking points, and according to their value, the truck is stable in all cases, but it will be greatly affected when the value of the tracked point decreases. When the L_d is equal to 4, it appears that the truck loses its stability in a very simple way, but As shown in the second figure from the left, the truck with increasing speed shows significantly that as the speed increases, the facilitation will increase, and since the truck's mass is large, the amplitude of the waves, as shown previously, will increase dramatically. It is worth noting that whenever the speed increases to 30 km/h, the oscillation of the waves will increase with the increase in speed, as shown in the diagram, and the truck begins to lose stability. The controller reaches the state of slumping quickly at a speed of less than 40 km / h. This also means that the controller remains effective within the limits of the appropriate mass and speed and is affected by all factors on it. As the last drawing shows that the truck will lose stability and the oscillation of the waves increases dramatically. But as it was previously explained, it remains after the point that is being tracked as one of the important factors as well, so it remains slightly stable, but it will lose stability while the speed increases more.

For truck when it is following the X-axis:

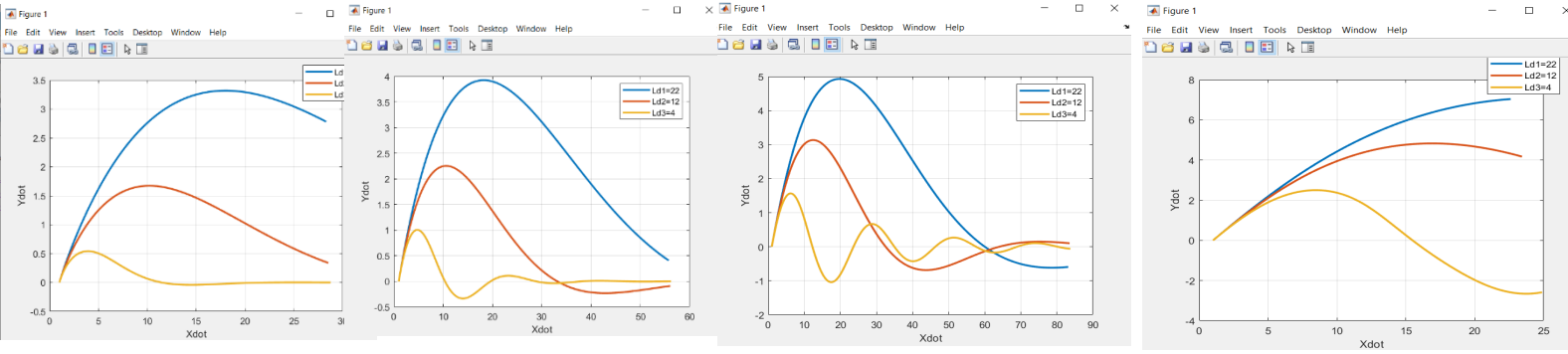


Figure 5.5: All Results for Truck When It's Following the X-Axis

When the truck follows the X-axis, there is no significant difference when the truck follows the Y-axis, as the first figure shows, it starts in the beginning in a smooth shape and then begins to appear when the point being tracked is few and while it takes longer for the oscillation and instability to appear in the tracking points are large, but there is an important point that the speed of the bicycle affects the shape of the waves in both cases, as was previously explained. The second and third figures show that the bicycle remains in a stable state to a certain extent, and also begins to lose stability when the amplitude of the wave appears. In the last figure from the right, it shows a break in the control unit, as it was previously shown that at small speed, the truck loses control and a loss of stability and the oscillation appears significantly. But note that in the last diagram on the right, the truck can remain in a slightly stable state when the L_d value is large.

For Stanley Model:

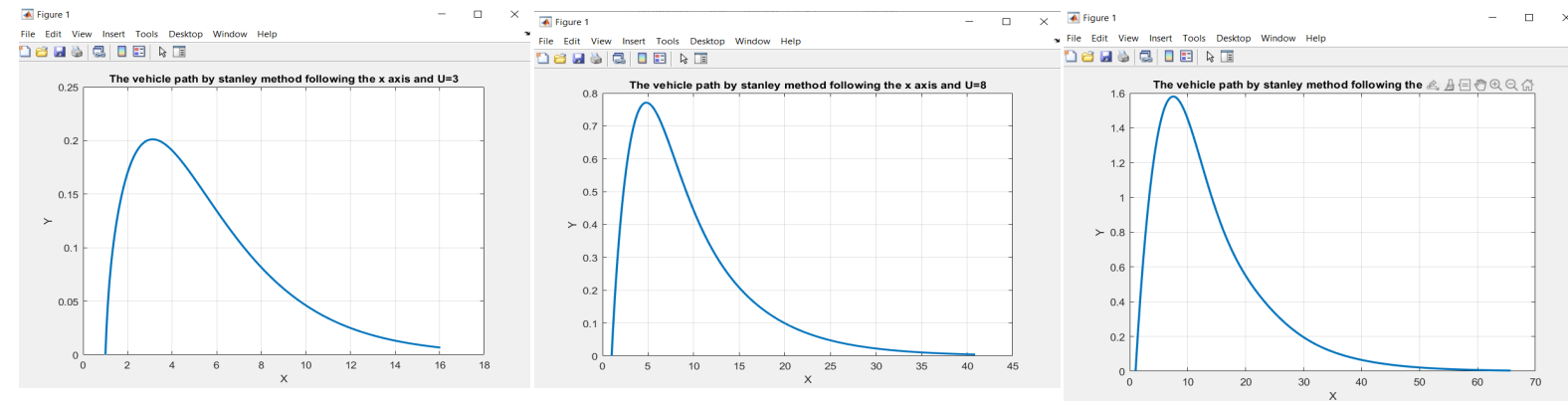


Figure 5.6: All Results for Stanly Model with Different Speeds

One of the most important things that can be concluded is that this method of tracking points is very effective for high speeds because it depends on the angle and does not depend on the distance of the point being tracked on the desired path, and this is more efficient than the pure pursuit method. We

also note by comparing the two methods that were used pure pursuit, and Stanley methods that the two methods have almost the same results, when the point being tracked is 20 meters large for the same initial conditions.

Stanley Method with changing the center of gravity to the front wheel:

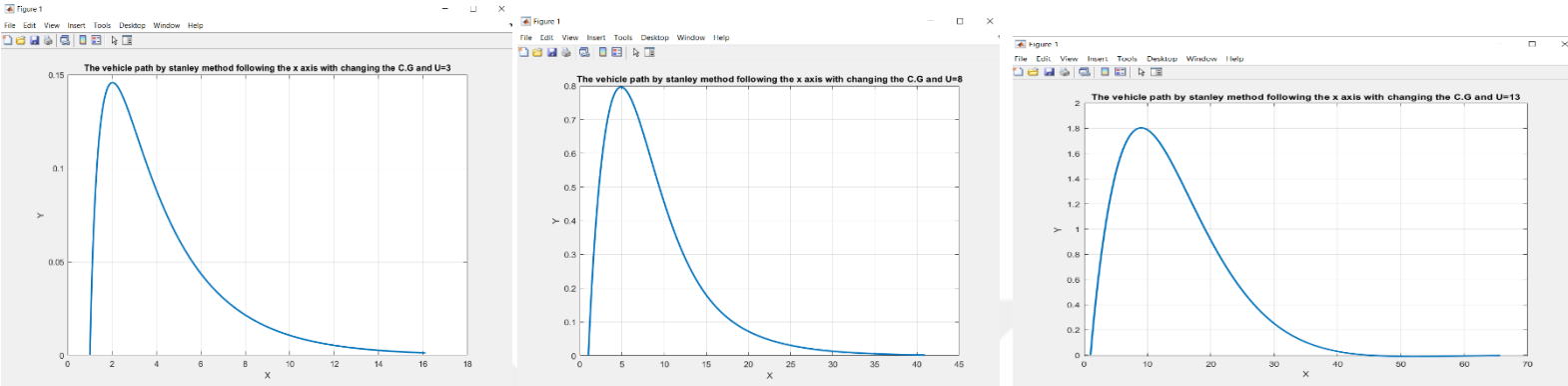


Figure 5.7: All the Results for Stanley Method with Changing the Center of Gravity Near to the Front Wheel

REFERENCES

- [1] Sun, Y., & Ling, L. (2022). An optimal tangential contact model for wheel-rail non-Hertzian contact analysis and its application in railway vehicle dynamics simulation. *Vehicle System Dynamics*, 60(9), 3240-3268.
- [2] Kong, J., Pfeiffer, M., Schildbach, G., & Borrelli, F. (2015, June). Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1094-1099). IEEE
- [3] Liu, J., Yang, Z., Huang, Z., Li, W., Dang, S., & Li, H. (2021, July). Simulation Performance Evaluation of Pure Pursuit, Stanley, LQR, MPC Controller for Autonomous Vehicles. In *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)* (pp. 1444-1449). IEEE.
- [4] Conlter, R. C. (1992). Implementation of the Pure Pursuit Path'hcking Algorithm
- [5] Shin, D.H “High Performance Tracking of Bxplicit Path by Roadworthy Mobile Robots.”, Doctoral Thesis, Dept. of Civil Engineering, CMU, May, 1990.
- [6] Cibooglu, M., Karapinar, U., & Söylemez, M. T. (2017, July). Hybrid controller approach for an autonomous ground vehicle path tracking problem. In *2017 25th Mediterranean Conference on Control and Automation (MED)* (pp. 583-588). IEEE.
- [7] Morales, J., Martínez, J. L., Martínez, M. A., & Mandow, A. (2009). Pure-pursuit reactive path tracking for nonholonomic mobile robots with a 2D laser scanner. *EURASIP Journal on Advances in Signal Processing*, 2009, 1-10
- [8] Yang, Y., Li, Y., Wen, X., Zhang, G., Ma, Q., Cheng, S., ... & Chen, L. (2022). An optimal goal point determination algorithm for automatic navigation of agricultural machinery: Improving the tracking accuracy of the Pure Pursuit algorithm. *Computers and Electronics in Agriculture*, 194, 106760.
- [9] Zhang, B., Li, Z., Ni, Y., & Li, Y. (2022). Research on Path Planning and Tracking Control of Automatic Parking System. *World Electric Vehicle Journal*, 13(1), 14.

- [10] Hoffmann, G. M., Tomlin, C. J., Montemerlo, M., & Thrun, S. (2007, July). Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. In *2007 American control conference* (pp. 2296-2301). IEEE.
- [11] Ibrahim, A. E. B. (2022). Self-tuning Look-ahead Distance of Pure-pursuit Path-following Control for Autonomous Vehicles Using an Automated Curve Information Extraction Method. *International Journal of Intelligent Transportation Systems Research*, 1-11.
- [12] Kim, D. H., Han, C. S., & Lee, J. Y. (2013). Sensor-based motion planning for path tracking and obstacle avoidance of robotic vehicles with nonholonomic constraints. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 227(1), 178-191.
- [13] Pendleton, S. D., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y. H., ... & Ang, M. H. (2017). Perception, planning, control, and coordination for autonomous vehicles. *Machines*, 5(1), 6.
- [14] <https://www.e-bike.kiwi/product/tui-step-through-e-bike/>
- [15] <https://www.automarket.cz/en/man-8-163-lc-4x2-6098>