Autonomous Unmanned Ground Vehicle (UGV) Follower Design

A thesis presented to

the faculty of

the Russ College of Engineering and Technology of Ohio University

In partial fulfillment

of the requirements for the degree

Master of Science

Yuanyan Chen

August 2016

This thesis titled

Autonomous Unmanned Ground Vehicle (UGV) Follower Design

by

YUANYAN CHEN

has been approved for

the School of Electrical Engineering and Computer Science

and the Russ College of Engineering and Technology by

Michael S. Braasch

Neil D. Thomas Professor of Electrical Engineering and Computer Science

Jim J. Zhu

Professor of Electrical Engineering and Computer Science

Dennis Irwin

Dean, Russ College of Engineering and Technology

Abstract

CHEN, YUANYAN, M.S., August 2016, Electrical Engineering

Autonomous Unmanned Ground Vehicle (UGV) Follower Design (74 pp.)

Directors of Thesis: Michael S. Braasch and Jim J. Zhu

A vehicle-to-vehicle follower design based on RC Unmanned Ground Vehicle (UGV) is presented in this thesis. To achieve the desired performance for two-vehicle leader-follower, a 3DOF path trajectory tracking controller with a close-loop guidance controller are used, which consider both the kinematics and the dynamics characteristics of the vehicle model. In our research, we use a Trajectory Linearization Control (TLC) to achieve the path tracking, and a PID controller to guide the preceding vehicle. To this end, the following objectives have been achieved. First, a 3DOF kinematics and dynamics vehicle model has been built. Second, an Adaptive Cruise Control (ACC) Trajectory Linearizaton Control (ACCTLC) scheme is presented. Third, a Vehicle-to-Vehicle following Trajectory Linearizaiton Control is proposed. MATLAB/SIMULINK simulation testing of 3DOF control algorithm is presented, which verifies the algorithm. Future work include implementing the current controller design by installing those algorithms to the real RC car; as well as adding lane constraint to the current work; and adding obstacle avoidance to develop fully autonomous ground vehicle.

To my husband

Zhongen Li

ACKNOWLEDGMENTS

Throughout my entire Masters study, I have been inspired, supported, and motivated by many people. Without their kindness, advice, and encouragement, a successful outcome would have been unimaginable.

First, I would like to sincerely thank my adviser Dr. Michael Braasch. Dr. Braasch brought me into the Ohio University Avionics Engineering Center, and at the first time, I came into contact with avionics and vehicle control. Since then he has been guiding me throughout my research. I would also like to extend my appreciation for the support of my co-adviser, Dr. J. Jim Zhu, his insight feedback and guidance control theory have kept me moving to the right direction.

I would like to acknowledge my thesis committee members Dr Douglas A. Lawrence and Dr. Maarten Uijt de Haag for both giving me insightful comments on my research work and lecturing extraordinary avionics and control courses. I would also like to acknowledge research engineer Tony Adami, our technical support, he always give me a hand when I need help.

My colleagues and friends in the Avionics Engineering Center and Controls Laboratory at Ohio University, Pengfei Duan, Shih-Wei Yen, Yang Liu, Letian Lin , Yue Zhao and Zhewei Wang have always been on my side over the years. Our many discussions have been very beneficial to my research.

My husband Zhongen Li and his family have been very supportive through many difficult times, offering many kind helps and understanding during my overseas study.

I am very grateful to Ohio University, The Stocker and Avionics Engineering Center for their financial support during my study, including Teaching Assistantship and Avionics Fund. Finally, I sincerely thank my parents and my husband, Zhongen Li, who have always believed in me and been constant inspirations to me, and have given me a great amount of support to reach this point.

Table of Contents

List of Tables

List of Figures

List of Acronyms

ACC Adaptive Cruise Control

AGV Autonomous Ground Vehicle

AVN Autonomous Vehicle Navigation

DOF Degrees of Freedom

EOM Equations of Motion

GCS Ground Command Station

GNC Guidance Navigation & Control

LKAS Lane Keeping Assist Systems

LTI Linear Time Invariant

LoS Line of Sight

MPC Model Predictive Controller

NED North-East-Down

PPG Pure Pursuit Guidance

PWM Pulse-Width-Modulation

SISO Single-input & Single-output

SLAM Simultaneous Localization and Mapping

TLC Trajectory Linearization Controller

UAV Unmanned Aerial Vehicles

UGV Unmanned Ground Vehicles

NOMENCLATURE

α_{LoS} = Azimuth angle of the LoS

β = Side slip angle under b-frame

$\boldsymbol{L}$ = Line of Sight

$\boldsymbol{R}$ = Range (relative distance) vector between vehicles

χ_g = Guidance heading angle command

χ_{err} = $\alpha_{LoS} - \chi_{sen}$

χ_{sen} = Heading angle of the vehicle

$\dot{P}_{sen}, \boldsymbol{V}$ = $\begin{bmatrix} \dot{P}_{x_{sen}} & \dot{P}_{y_{tgt}} \end{bmatrix}$ Sensed velocity vector in inertial frame

Γ = $\begin{bmatrix} \phi & \theta & \psi \end{bmatrix}$ Body attitude Euler angle

γ_{seeker} = Servo rotation angle

Ω = $\begin{bmatrix} p & q & r \end{bmatrix}$ Body frame angular velocity

ω_m = Motor angular speed

ω_n = Nature frequency

ρ = Mass density of air

σ_x = Slip ratio

θ_V = Tire velocity angle

ζ = Damping coefficient

A_F = Front area of the vehicle

C = Safety distance between vehicles

C_D = Aerodynamic drag coefficient

C_α = Cornering stiffness

C_σ = Tire stiffness

C_{rr} = Rolling resistance

E_m = Armature voltage

F = $\begin{bmatrix} F_x & F_y & F_z \end{bmatrix}$ Body frame force vector

g = Gravity acceleration

i = Armature current

I^*_{**} = Inertial coefficients

J_m = Motor moment of inertia

K_m = Motor electro-mechanical constant

L = Armature inductor

m = Mass of the vehicle

N = Gear ratio

P = $\begin{bmatrix} x_n & y_n & z_n \end{bmatrix}$ Inertial frame position

P_{sen} = $\begin{bmatrix} P_{x_{sen}} & P_{y_{sen}} \end{bmatrix}$ Sensed position in inertial frame

P_{tgt} = $\begin{bmatrix} P_{x_{tgt}} & P_{y_{tgt}} \end{bmatrix}$ Target position in inertial frame

R_{eff} = Effective radius of the tire

R_{laser} = Laser range data

T_m = $\begin{bmatrix} L_m & M_m & N_m \end{bmatrix}$ Body frame moment vector

V = $\begin{bmatrix} u & v & w \end{bmatrix}$ Body frame velocity

V_g = Guidance speed command

V_{wind} = Wind velocity

1 INTRODUCTION

Technological advances have spread the use of robotic unmanned vehicles in several fields, such as Unmanned Aerial Vehicles (UAV), Unmanned Under Water Vehicle (UUV) and Unmanned Ground Vehicles (UGV). Each of those three types of vehicles play crucial roles in increasing efficiency, performance and safety in both civilian and military applications [3] [4].

By definition, a UGV is a vehicle that operates in contact with the ground without an on-board human presence. It is a land-based robot that is the counterpart to UAV and UUV. From the operation point of view, there are two classes of UGV: remotely-operated UGV and autonomous UGV. An autonomous vehicle requires the integration of many technologies which can make it self-acting and self-regulating without human intervention. In the past decade, due to advances in computing and sensing technology, autonomous vehicles, are proliferating. The research topic in this thesis focuses on autonomous UGV control.

1.1 Background

The idea of an autonomous ground vehicle can be traced back to the late 1920 [4]. From that time on, more and more work has been done. The first truly autonomous car appeared in 1984, with Carnegie Mellon University's Navlab and ALV projects [4]. Since then, numerous research organizations and commercial companies have developed prototype autonomous vehicles. Here are some examples: in 1994, UniBw Munich and Daimler-Benz presented autonomous driving in three-lane traffic with speeds up to 130km/h which included lane marking and other vehicle tracking [1]. By the 2000s, lane detection was used to facilitate lane departure warnings for drivers, and a lane keeping assist system (LKAS) was used for heading control [1]. More recently, adaptive cruise control (ACC) systems have been introduced for tracking and detection of preceding

vehicles and the pre-crash system triggers full braking power to lessen damage for driver's slow reaction [1].

After almost 30 years of research, autonomous vehicles have achieved the following functionalities:

• Vehicle state estimation

The estimation of the states of the vehicle, like position, velocity and acceleration of the vehicle, is the basic level of the control task. State estimation is crucial in practical applications, such as off-road navigation. Vehicle state estimation compensates vehicle's ego-motion under poor GPS conditions [2].

• Obstacle detection

Obstacle detection is one of the most prominent topics in autonomous vehicle research. In general, obstacle detection is divided into two branches, static obstacles and moving obstacles.

In order to handle static obstacles, occupancy mapping is usually used. Data from real-time camera or LIDAR point clouds (large LIDAR data sets) is condensed into a regular matrix grid. Each matrix grid cell stores the probability for the existence of the obstacle. One of the most popular methods in static obstacle detection is Simultaneous Localization and Mapping (SLAM) [12]. The SLAM robot shown in Figure 1.1 can update the map of an unknown environment while simultaneously keep track of its own location, Figure 1.2 is an example of the range and positions captured by the robot [12].

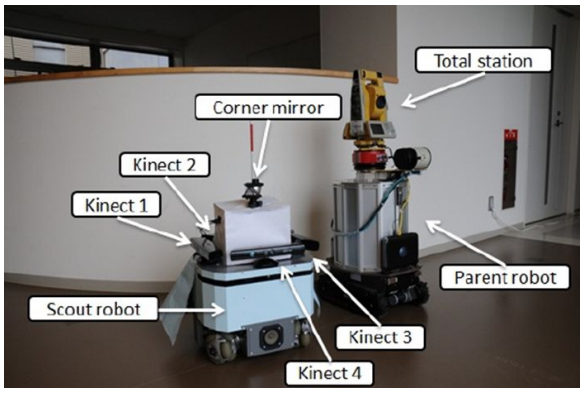


Figure 1.1: A SLAM Robot with Four RGB-D Cameras [12]

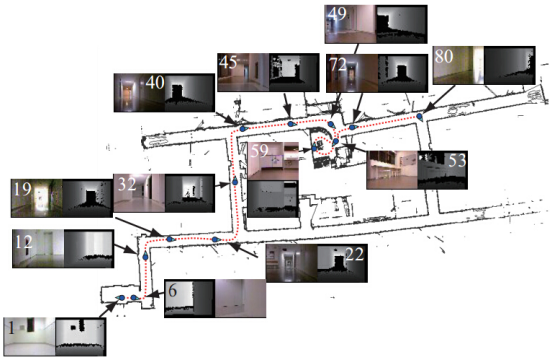


Figure 1.2: SLAM System Result [12]

Moving obstacle detection is widely used for traffic safety. Current research articles usually describe different object classes in different motion models. Several methods detect moving objects in the image plane. Usually, an image plane is segmented into different parts according to the optical flow, and group pixels have similar flow [5]. Optical flow can also be used for static object detection.

Google's self-driving car in Figure 1.3 can "see" moving objects like other cars in traffic in real time. So it can react to avoid those objects.

Figure 1.3: Google Car [7]

• Road shape estimation

Detecting static and moving objects is helpful for safety driving, but it is not sufficient for full autonomous driving. There are many other tasks required for autonomous driving. In recent decades, much research has been done on tracking lanes in 2-D image and actual system use clothoidals [9] or B-spline models for 3-D lane representation [10]. By projecting a 3-D estimation of the road, an efficient measurement of direction can be performed.

For the past 30 years, UGVs have been used in different situations, therefore, guidance and navigation is an active area in UGV research. Global navigation dominates the autonomous guidance in global metric maps of the environment[1]. Global navigation provides the planning and mapping algorithms. The corresponding method is reactive navigation [1]. A reactive navigation mechanism does not provide any mapping information, but deals with planning paths in real time with respect to perceived obstacles. Object recognition is one of the goals for reactive navigation. Between global navigation and reactive navigation is guided navigation [1]. This type of navigation uses environment structures whose perception allows the direct estimation for feedback control [1]. A good

example for guided navigation is a UGV that follows a pre-designed path or an off-road track.

Although AGV developing is a challenging work in the vehicle control field, it brings incredible benefits. According to [11], about 8 years ago, the U.S. Army has set a goal to have one-third of all operational ground vehicles converted to AGVs instead. If achieved, it will save countless lives on the battlefields. Beside military use, AGVs are also beneficial in civil use. Unlike the human drivers with limited situational awareness, AGVs can monitor the environment by using sensors with 360° field of view and they can quickly make a safe decision to avoid the potential hazards.

In the next few years, autonomous driving on public roads will be promoted in the market. Google and other major companies, like Honda and Toyota, have already tested AGVs on the roads in California and Texas. In the next 5-10 years, robust autonomous vehicle systems will not only rely on sensors for perception, but will also utilize self reflection to be aware of the reliability of the sensors and perception modules[1]. For the long-term future, more cognitive systems and a higher level of autonomy will be the main work for researchers.

1.2 Problem Statement and Objectives

Autonomous UGV comprise three basic subsystems, Guidance, Navigation and Control, which are usually called the GN&C system. Guidance determines the desired path from the vehicle's current location to a designated target. Navigation determines the vehicles position, velocity and attitude at the given time. Control system tracks guidance commands while maintaining vehicle stability.

Avionic Engineering Center in Ohio University combines navigation issues encountered in air and ground transportation. We will build an integrated GN&C system for autonomous UGV platform. This system can help students understand the kinematic

and dynamic property of the ground vehicle and also can be used for future research use. The basic functionality of this system is that, a leading vehicle is driving from any visible position, and the following vehicle can follow the leading vehicle at a safety distance by tracking the guidance trajectory generated by guidance sub-system.

This thesis aims to develop a complete set of GN&C systems for a unified Autonomous Vehicle Leader-Follower System. The whole system includes two parts: guidance trajectory generation and trajectory tracking. The trajectory guidance subsystem uses a pure pursuit guidance (PPG) system to generate guidance commands in relation to the Line of Sight (LoS) to its intended target. The trajectory tracking subsystem provides a reliable control which can ensure any suitable trajectory tracked accurately. Figure 1.4 shows the diagram of both leading and following vehicle's trajectory.

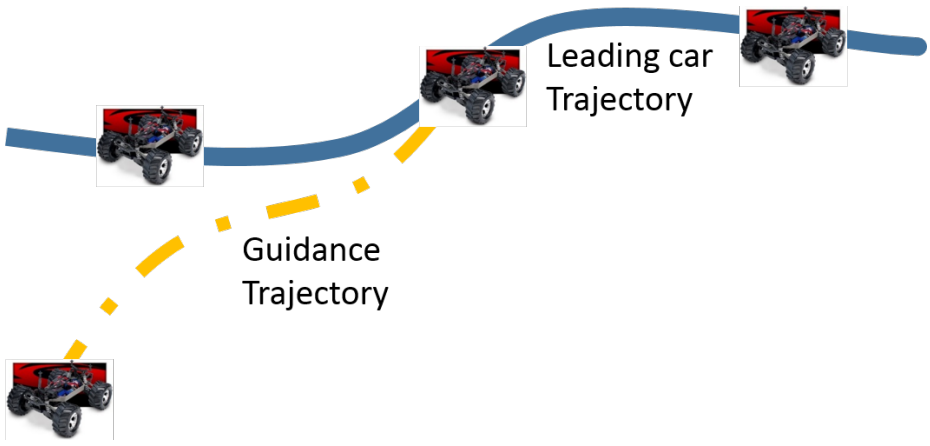


Figure 1.4: Vehicle Tracking

1.3 Approach

Due to the highly nonlinear and time-varying nature of the ground vehicle dynamics, vehicle tracking problem is, in fact, a very complex task. Based on the control task and vehicle constraints, significant research has been performed by separating the vehicle into

different models: 1) Only consider the kinematic model of the vehicle [13][14]; 2) only consider the dynamics of the vehicle [15][16]; 3) the integrated problem of the kinematics and dynamics[17]. On the other hand, many classical control methods have been developed for vehicle control such as Model Predictive Controller (MPC) [14][17] and State Feedback Controller [18].

In this thesis, a new method, Trajectory Linearization Controller (TLC) will be introduced to deal with the vehicle control for the following vehicle. TLC has been widely used in flight control [19][20][21]. It can be viewed as an ideal gain-scheduled controller designed at every point on the trajectory [22]. The trajectory command is embedded in the controller gain. It provides robust stability without the slowly varying constraint. Figure 1.5 [19] illustrates the TLC concept.

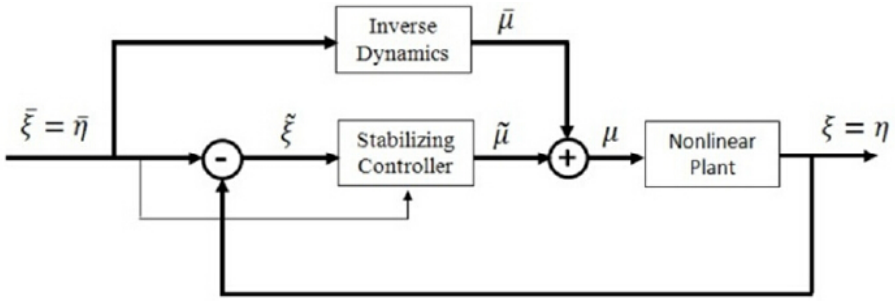


Figure 1.5: Trajectory Linearization Concept [19]

TLC combines an open-loop dynamic inverse of the nonlinear plant with a closed-loop tracking error regulator that accounts for disturbances, model error and internal dynamics. So it approximately cancels the plant nonlinearity with an open-loop controller and reduces the tracking error by Proportional-integral (PI) feedback controller.

In this thesis, the TLC controller is designed for a 3DOF, double-motor RC vehicle in Figure 1.6. The vehicle's force/torque model and actuator model are given in the following chapters. Simulation results are given for longitudinal motion and turning maneuvers.

Figure 1.6: Trannas 3DOF RC Vehicle

1.4 Summary and Significance

The UGV used in this thesis is modeled with 6DOF rigid body equations which consider both the kinematics and dynamics of the vehicle and are later simplified for controller design. An autonomous 3DOF guidance, navigation and control system is proposed based on a nonlinear dynamic model of the RC vehicle in Figure 1.6. The control structure employs a multiple-loop trajectory linearization controller (TLC). The designed controller is able to follow the guidance trajectory. This design will be the first step for future Autonomous Vehicle Control research.

1.5 Organization

This thesis is organized as follows. In chapter 2, vehicle modeling will be presented in detail. Chapter 3 presents the TLC design for 1DOF longitudinal motion. Chapter 4 presents the controller design for 3DOF vehicle directional motion. Chapter 5 presents the PPG guidance trajectory design. In Chapter 6, simulation results are presented to illustrate the advantages of the controller. Chapter 7 is the conclusion for this thesis.

2 Vehicle Modeling

This chapter provides the 3DOF equations of motion for the longitudinal, lateral and directional motion of the UGV. All of the equations are based on state-space equations. Those equations are simplified directly from 6DOF rigid body equations of motion. Vehicle motion is affected by forces. All forces are discussed in detail and mathematical equations are given. The actuator model is given which consists of the vehicle's steering and drive motor dynamics for the specific RC vehicle. At the end, Traxxas RC car parameters are given.

2.1 State-space Introduction

In the classical control method, a dynamic system is usually described by a transfer function. This method models the plant as a linear-time-invariant input-output mapping. As such, the nonlinear time-varying dynamics and internal dynamics of the plant are lost. Modern control theory solves these limitations by using state-space equations.

State-space equations provide a richer description of the plant dynamics. As stated in [8], a state-determined model of a dynamics system always has the following properties:

The system in terms of a minimum set of variables $x_i(t), i = 1, \cdots, n$*, together with those variables at an initial time* t_0 *and the system inputs for* $t \geq t_0$*, are sufficient to predict the future system state and outputs for all time* $t \geq t_0$ [8].

A standard form of state equations contains n coupled first-order differential equations and each equation is expressed in terms of $x_1(t), \cdots, x_n(t)$, called state variables.

The standard form of state equation is given as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} f_1(\mathbf{x}, \mathbf{u}, t) \\ f_2(\mathbf{x}, \mathbf{u}, t) \\ \vdots \\ f_n(\mathbf{x}, \mathbf{u}, t) \end{bmatrix} \tag{2.1}$$

The vector notation is written as $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t)$.

For linear time invariant (LTI) system, the state equations can be summarized as:

$$\begin{aligned} \dot{\mathbf{x}} &= A\mathbf{x} + B\mathbf{u} \\ \mathbf{y} &= C\mathbf{x} + D\mathbf{u} \end{aligned} \tag{2.2}$$

where A is an $n \times n$ square matrix of the system and B is an $n \times r$ matrix, C is an $l \times n$ matrix and D is an $l \times r$ matrix.

2.2 Coordinate Frames and 6DOF Rigid Body Equation of Motion

In this section, we will define the coordinate frames and the corresponding equations of motion for this project. There are two frames which are needed, one is a North-East-Down (NED) frame or the navigation frame (n-frame) and another is a body-fixed frame (b-frame). In order to distinguish these two coordinate systems, we define the n-frame as x_n points to the north, y_n points to the east, and z_n points down to the center of the Earth. Because of the low speed motion of the UGV, the Earth-frame will be assumed as inertial frame. For the body fixed frame, set x_b as the longitude of the vehicle which pointing forward, y_b as the driver's right side of the vehicle and z_b points down, as shown in Figure 2.1.

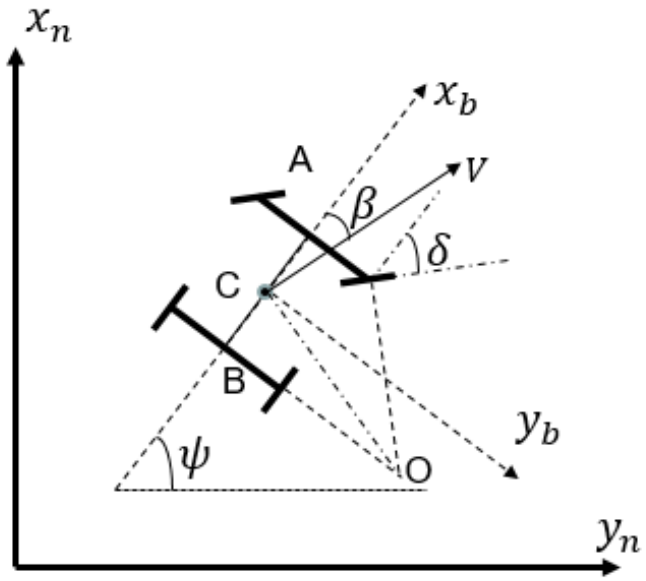


Figure 2.1: Coordinate Frame

6DOF rigid body motion can be decomposed into translational kinematics, translational dynamics, rotational kinematics and rotational dynamics [21].

The translational kinematics are given as:

$$\begin{bmatrix} \dot{x}_n \\ \dot{y}_n \\ \dot{z}_n \end{bmatrix} = \begin{bmatrix} C_\theta C_\psi & -C_\phi S_\psi + S_\phi S_\theta C_\psi & S_\phi S_\psi + C_\phi S_\theta C_\psi \\ C_\theta S_\psi & C_\phi C_\psi + S_\phi S_\theta S_\psi & -S_\phi C_\psi + C_\phi S_\theta S_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{2.3}$$

where $\boldsymbol{P} = \begin{bmatrix} x_n & y_n & z_n \end{bmatrix}^T$ is the inertial frame position; $\boldsymbol{V} = \begin{bmatrix} u & v & w \end{bmatrix}^T$ is body-frame velocity; $\boldsymbol{\Gamma} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$ is body attitude Euler angle and

$$\begin{aligned} C_\phi &= \cos\phi \quad S_\phi = \sin\phi \\ C_\theta &= \cos\theta \quad S_\theta = \sin\theta \\ C_\psi &= \cos\psi \quad S_\psi = \sin\psi \end{aligned} \tag{2.4}$$

The translational dynamics is given as:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 0 & r & -q \\ -r & 0 & p \\ q & -p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \frac{1}{m} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{2.5}$$

where $\boldsymbol{\Omega} = \begin{bmatrix} p & q & r \end{bmatrix}^T$ is body-frame angular velocity and $\boldsymbol{F} = \begin{bmatrix} F_x & F_y & F_z \end{bmatrix}^T$ is body-frame force vector.

The rotational kinematics are given as:

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \frac{S_\phi S_\theta}{C_\theta} & \frac{S_\theta S_\phi}{C_\theta} \\ 0 & C_\phi & -S_\phi \\ 0 & \frac{S_\phi}{C_\theta} & \frac{C_\phi}{C_\theta} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.6}$$

The rotational dynamics are given as:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} I^p_{pq} pq + I^p_{pr} pr \\ I^q_{pp} p^2 + I^q_{rr} r^2 + I^q_{pr} pr \\ I^r_{pq} pq + I^r_{qr} qr \end{bmatrix} + \begin{bmatrix} g^p_l & 0 & g^p_n \\ 0 & g^q_m & 0 \\ g^r_l & 0 & g^r_n \end{bmatrix} \begin{bmatrix} L_m \\ M_m \\ N_m \end{bmatrix} \tag{2.7}$$

where $\boldsymbol{T_m} = \begin{bmatrix} L_m & M_m & N_m \end{bmatrix}^T$ is the body-frame moment vector and the inertial coefficients I^*_{**} are calculated from the vector rotational EOM

$$\dot{\boldsymbol{\Omega}} = \mathbf{I}^{-1} \left[-\boldsymbol{\Omega} \times \mathbf{I}\boldsymbol{\Omega} + \mathbf{M} \right] \tag{2.8}$$

where $\boldsymbol{M} = \begin{bmatrix} M_x & M_y & M_z \end{bmatrix}^T$, and $\boldsymbol{I}$ is the moment of inertia matrix [21].

2.3 Vehicle EOM: 6DOF to 3DOF Simplification

In this section, the UGV's 3DOF dynamics and kinematics model for the longitudinal, lateral and directional motions will be simplified from previous 6DOF rigid body EOM.

2.3.1 Longitudinal Motion

The UGV's longitudinal motion is a 1DOF motion. The vehicle is assumed to be moving on a flat road without lateral velocity; yaw angle ψ is a constant and angular velocity r is 0. Based on above assumptions, the state equations can be simplified as:

$$\begin{aligned} \dot{x}_n &= \cos(\psi)u \\ \dot{u} &= \frac{F_x}{m} \\ \dot{v} &= \dot{\psi} = \dot{r} = 0 \\ y_n &= \tan(\psi)x_n \end{aligned} \tag{2.9}$$

2.3.2 3DOF Motion

For 3DOF motion, assuming the vehicle is on a flat road implies $\phi = \theta = 0$ and $z_n = 0$, and the equations of motion can be written as:

Translational Kinematics:

$$\begin{bmatrix} \dot{x}_n \\ \dot{y}_n \end{bmatrix} = \begin{bmatrix} C_\psi & -S_\psi \\ S_\psi & C_\psi \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = B_1(\psi)\, V \tag{2.10}$$

Translational Dynamics:

$$\begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} 0 & r \\ -r & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \frac{1}{m} \begin{bmatrix} F_x \\ F_y \end{bmatrix} = A_2(r)\, V + \frac{1}{m} F \tag{2.11}$$

Rotational Kinematics:

$$\dot{\psi} = r \tag{2.12}$$

Rotational Dynamics:

$$\dot{r} = \frac{1}{I_{zz}} N_m \tag{2.13}$$

2.4 Force Model

This section states the UGV's force model, including tire traction force, aerodynamic drag force, rolling resistance force and gravitational force, which affect the vehicle's

motion. Models for both longitudinal and 3DOF motion are discussed separately in two sub-sections. The tire force modeling comes from [25].

2.4.1 Longitudinal Force

There are three types of forces that comprise the longitudinal force [25]: longitudinal tire traction force F_{xt}, rolling resistance force R_x and aerodynamic drag force F_{aero}

$$F_x = F_{xt} - F_{aero} - R_x \tag{2.14}$$

The longitudinal tire traction force can be modeled as

$$F_{xt} = C_\sigma \sigma_x \tag{2.15}$$

where C_σ is tire stiffness coefficient and σ_x is the longitudinal slip ratio which can be defined as

$$\sigma_x = \begin{cases} \dfrac{R_{\text{eff}}\omega_w - u}{u}, & \text{during braking} \\ \dfrac{R_{\text{eff}}\omega_w - u}{R_{\text{eff}}\omega_w}, & \text{during acceleration} \end{cases} \tag{2.16}$$

where u is vehicle's longitudinal velocity, R_{eff} is effective wheel radius and ω_m is motor's rotational speed. Figure 2.2 shows the relationship between longitudinal tire force and the slip ration. From the figure, when $\sigma_x \in \begin{bmatrix} -0.1 & 0.1 \end{bmatrix}$, the tire force is a linear function of the slip ratio. It means the vehicle is not slippery at this small range; otherwise, the vehicle is slippery. So Eq (2.14) is obtained under the assumption that the slip ratio belongs to the small range in which the vehicle is not slippery.

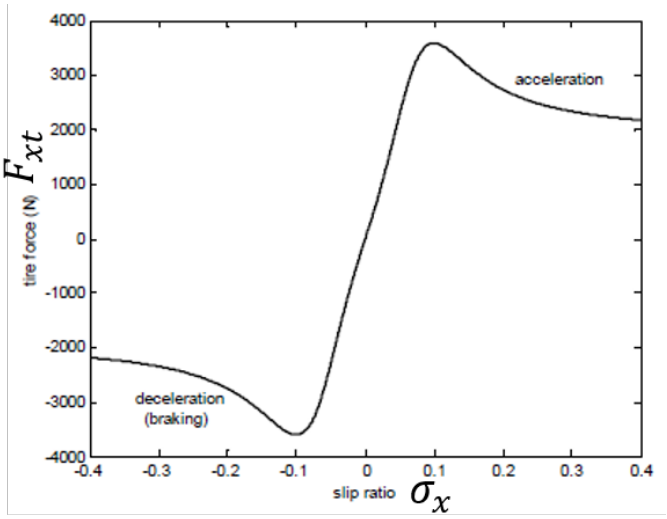


Figure 2.2: Longitudinal Tire Force as a Function of Slip Ratio [25]

The aerodynamic drag force on the vehicle can be represented as

$$F_{aero} = \frac{1}{2}\rho C_D A_F (u + V_{wind})^2 \tag{2.17}$$

where ρ is the mass density of air, C_D is the aerodynamic drag coefficient, A_F is the front area of the vehicle, and V_{wind} is the wind velocity.

Rolling resistance force is roughly proportional to the nominal force mg

$$R_x = C_{rr} mg \tag{2.18}$$

where C_{rr} is rolling resistance coefficient, typically the value is 0.015 [25].

2.4.2 Lateral Force

The tire lateral traction force is the only force that affects the lateral motion. The tire lateral traction force for the front and rear tires can be written respectively as [25].

$$F_{y_f} = 2C_{\alpha_f}\left(\delta - \theta_{V_f}\right) \tag{2.19}$$

$$F_{y_r} = 2C_{\alpha_r}\left(-\theta_{V_r}\right) \tag{2.20}$$

where θ_{V*} defines the vehicle velocity side slip angle as shown in Figure 2.3 [25], $C_{\alpha*}$ is cornering stiffness. Here and throughout the paper the indices $_{.f}$ and $_{.r}$ refer to the front and rear axle, respectively.

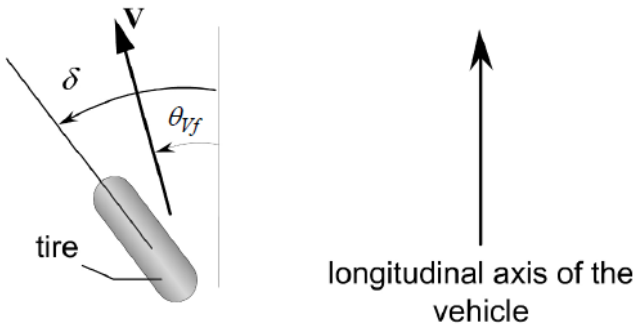


Figure 2.3: Velocity Angle

Figure 2.3 defines the velocity angle, where δ is front wheel steering angle.

The following equations can be used to calculate θ_{Vf} and θ_{Vr}

$$\begin{aligned} \tan\left(\theta_{V_f}\right) &= \frac{v + l_f\dot{\psi}}{u} \\ \tan\left(\theta_{V_r}\right) &= \frac{v - l_r\dot{\psi}}{u} \end{aligned} \tag{2.21}$$

where l_f and l_r are longitudinal distance of the front and rear axle to the c.g.

2.5 Actuator Modeling

There are two actuators that we need to model when working on the actuator modeling. One is the vehicle steering actuator and the other is drive motor dynamics. We will start with the steering actuator.

For our RC car, the steering actuator consists of two identical servo motors with $0° \sim 60°$ rotational range. The servo motors require $0 \sim 6V$ to drive it and the angular position is controlled by Pulse-Width-Modulation (PWM). Pulse width is from $1ms$ to $2ms$ corresponding to $\delta = -30°$ and $\delta = 30°$ with $1.5ms$ at the center corresponding to

$\delta = 0°$. The expression can be given as

$$\delta = \frac{V_s}{V_{s_{max}}}\delta_{max} - \frac{\delta_{max}}{2} \tag{2.22}$$

where V_s is the input voltage for steering servo, $V_{s_{max}}$ is the maximum input voltage ($6V$) and δ_{max} is the maximum value of steering angle ($60°$).

The armature-controlled DC motor can be modeled by the following second-order state equation [28]

$$\begin{aligned} L_a\frac{di}{dt} &= -K_m\omega_m - R_a i + E_m \\ J_m\frac{d\omega_m}{dt} &= K_m i - B_m\omega_m - NR_{\text{eff}}F_{xL} \end{aligned} \tag{2.23}$$

where i, E_m, R_a and L_a are the armature current, voltage, resistance, and inductance respectively, J_m is moment of inertia, and B_m is torsional damping of the rotor, gears and wheels lumped together; ω_m is motor angular speed; N is gear ratio; K_m is motor electro-mechanical constant; R_{eff} is effective radius of the tire. The first equation is for electric transient in armature and the second equation is for mechanical motion. In general,time constant of the electric circuit is significantly smaller than the mechanical system, so the second equation of Eq (2.23) can be rewritten as

$$0 = -K_m\omega_m - R_a i + E_m \tag{2.24}$$

then the current can be obtained as

$$i = \frac{E_m - K_m\omega_m}{R_a} \tag{2.25}$$

substitute Eq (2.25) into Eq (2.23)

$$J_m\dot{\omega}_m = -\left(B_m + \frac{K_m^2}{R_a}\right)\omega_m + \frac{K_m}{R_a}E_m - NR_{\text{eff}}F_{xL} \tag{2.26}$$

In this section, we introduced 3DOF vehicle EOM and the vehicle's mechanical model. All of the equations will be used in controller design in subsequent chapters.

2.6 Vehicle Modeling Parameter Determination

The design provided in this paper is based on Traxxas RC car shown in Figure1.6. The purpose of this section is to describe how to determine the vehicle modeling parameters which will be used in simulation demonstration.

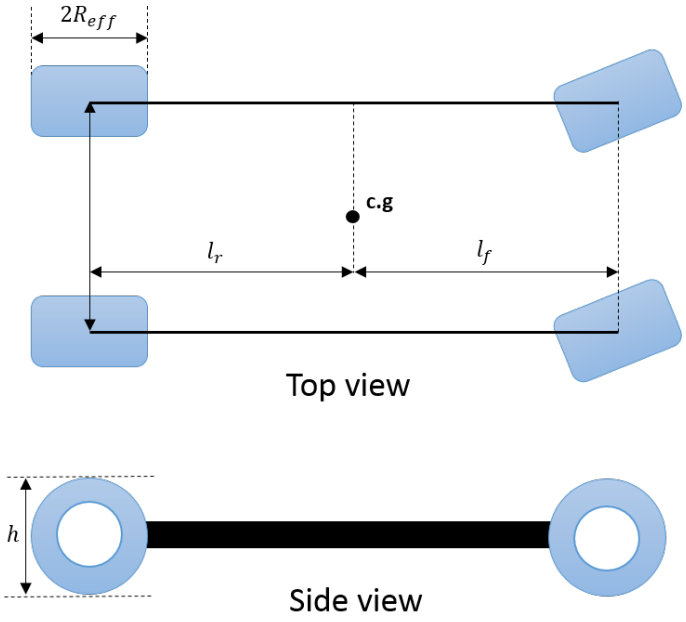


Figure 2.4: Sketch of the Vehicle Model

2.6.1 Moment of Inertia Around z-axis I_{zz}

The simplified 2D plant sketch of the RC car is given in Figure 2.4. In order to estimate the vehicle's moment of inertia around z-axis, we dissembled the vehicle into 5 parts, 4 tires and the vehicle body. We treat the tire as a thick-walled cylindrical tube with open ends, and the vehicle body as a solid cuboid; calculated the moment of inertia for each part and then added them together.

$$I_{zz} = \Sigma I_z \tag{2.27}$$

2.6.2 Longitudinal Stiffness C_σ

As for the estimation of the longitudinal stiffness, the procedure is discussed below:

1. Brake the wheels of the vehicle, and set the steering angle to 0 degree.

2. Apply a longitudinal force using standard weights through a pulley until the tires start to skid, and take this as the critical traction force F_{max} that the tires can produce at the critical slip ratio σ_{max}.

3. Assume the critical slip ratio is 0.1 based on typical tire data and assume the traction force is linear with the slip ratio under the critical slip ratio, the coefficient is then determined as $F_0/0.1$.

2.6.3 Cornering Stiffness C_α

The estimation for the cornering stiffness is described as below:

1. Same as the first step for C_σ estimation.

2. Apply a lateral force to the vehicle using standard weight through a pulley, and measure the angular lateral displacement of the tire wall at several force settings.

3. Then the cornering stiffness coefficient can be approximated by δ (Force)/ δ(Angular displacement).

2.6.4 Rolling Resistance C_{rr}

The rolling resistance C_{rr} of tires is a coulomb friction, which can be measured using standard weight w_0 to propel the vehicle at a constant low speed u_0. Then $C_{rr} \simeq \frac{w_0}{mg}$.

2.6.5 Motor Moment of Inertia

The motor moment of inertia is determined as a solid cylinder with radius r, height h and mass m by using Eq (2.28) [27]

$$I_z = \frac{mr^2}{2} \tag{2.28}$$

so based our Titan 550 electric motor specification, $r = 19$mm, $m = 300$g, motor moment of inertia $J_m = 5.4e^{-5}$.

2.6.6 K_m Constant

The dynamic equation for an armature-controlled DC motor is given in Eq (2.23) and the static equation is given in Eq (2.24). At the static state without any shaft load, the armature current is very small. So assume $i = 0$, K_m constant is then be estimated as

$$K_m = \frac{E_m}{\omega_m} \tag{2.29}$$

Here, we applied a high voltage, up to 12 V and the corresponding rotational speed is around 2200 RPM.

2.6.7 Armature Resistance R_a

The same method described in Section 2.6.6 was used to estimate the armature resistor. We applied a very low voltage to the motor such that the motor was rotating at $\omega_m \simeq 0$. Recording the voltage and current, and R_a was calculated by $R_a = \frac{E_m}{i}$

2.6.8 Motor Viscous Friction Constant B_m

The last parameter is motor viscous friction constant B_m. B_m can be determined by running the motor without shaft load at a high speed ω_m, then cut off the power and measure the time T for the speed to drop to $0.37\omega_m$, then by Eq (2.26),

$$B_m = \frac{J_m}{T} \tag{2.30}$$

Here we estimated it based on other motor's specification data sheet downloaded from the internet.

Table 2.1: Vehicle Modeling Parameters

R_{eff}	Effective wheel radius	0.0725	m
l_f	Longitudinal distance of the front axle to the c.g.	0.35	m
l_r	Longitudinal distance of the rear axle to the c.g.	0.35	m
m	Total mass	4.76	kg
I_{zz}	Moment of inertia about z-axis	0.0687	$kg \cdot m^2$
C_σ	Longitudinal tire stiffness	400	N/deg
C_{rr}	Tire rolling resistance	0.02	deg^{-1}
C_α	Cornering stiffness	150	N/deg
J_m	Motor moment of inertia	$5.4 * 10^{-5}$	$kg \cdot m^2$
B_m	Motor viscous friction constant	$0.5 * 10^{-5}$	$N \cdot m/(rad \cdot s)$
K_m	Motor KM constant	0.00515	$N \cdot m/\sqrt{W}$
R_a	Total armature resistance	0.2	Ω

Table 2.1 represents the parameters setting for Traxxas RC car described in Sections 2.6.1-2.6.8.

In this chapter, we gave the 3DOF vehicle modeling simplified from 6DOF rigid body EOM. The actuator model and vehicle parameters for the specific Traxxas RC car were given. The vehicle modeling in chapter will be used as nonlinear plant for the controller design in the subsequent chapters.

3 Autonomous Vehicle Longitudinal Following

Due to the nonlinear and time-varying nature of the vehicle dynamics in trajectory tracking, our research uses Trajectory Linearization Control (TLC) [19][20][21] to design a 1DOF, longitudinal trajectory tracking/adaptive cruise control for the RC car. TLC combines an open-loop dynamic inverse of the plant with a closed-loop tracking-error regulator that accounts for disturbances, model errors and internal dynamics.

The controller for this 1DOF application takes position command and calculate the voltage applied on the electric motor. For this application, we only consider the longitudinal characteristic of the vehicle, so only the guidance controller is needed to meet the control task. The overall closed-loop consists of two loops, guidance outer loop and guidance inner loop which provide the vehicle velocity control signal and vehicle longitudinal force signal respectively. Each loop has a complete TLC structure, a nominal controller with a feedback PI controller. The block diagram of the closed-loop system is shown in Figure 3.1.

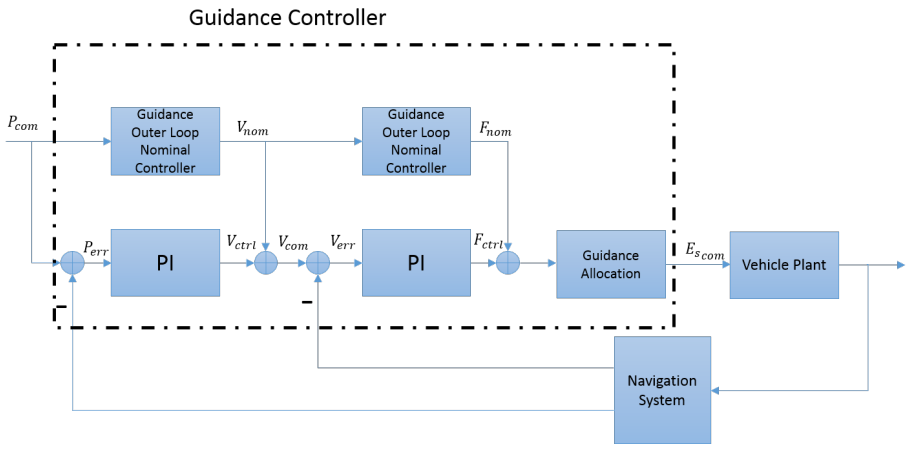


Figure 3.1: 1DOF Trajectory Linearization Controller Block Diagram

3.1 Guidance Controller

The guidance controller consists of two loops, guidance outer loop and guidance inner loop. We will start our design from the guidance outer loop. Guidance outer loop takes position measurement and position command to calculate velocity command for the guidance inner loop.

Based on the aforementioned simplification analysis in Chapter 2, the longitudinal mathematical model for Traxxas RC car is given as:

$$\begin{aligned} \dot{P} &= u \\ \dot{u} &= \frac{1}{m} F_x \\ \dot{\omega}_m &= -\left(\frac{B_m}{J_m} + \frac{K_m^2}{J_m R_a}\right)\omega_m + \frac{K_m}{J_m R_a} E_m + \frac{N R_{\text{eff}}}{J_m} F_{xL} \end{aligned} \tag{3.1}$$

with the force model in driving mode given by

$$F_x = -C_{rr} mg \ \text{sgn}(u) - C_D A_F \frac{\rho}{2} (u + V_{wind})^2 + C_\sigma \left(1 - \frac{u}{R_{\text{eff}} N \omega_m}\right) - mg \sin\theta \tag{3.2}$$

where θ is the angle of inclination of the road and is defined to be positive when the vehicle is going uphill. The following controller is designed without the effect of θ and wind velocity V_{wind}.

3.1.1 Guidance Outer Loop

The guidance outer loop takes position measurement and position command to calculate the velocity command. Here, it is a one dimensional motion, so the position is the distance traversed by the vehicle.

The guidance outer loop has a complete TLC structure, a nominal controller and a feedback controller. We will start the design from the nominal controller. The nominal body velocity is obtain from Eq (3.1) as $u_{nom} = \dot{P}_{nom}$, where $\dot{P}_{nom}$ is obtained from the

position command P_{com} using a pseudo-differentiator with a transfer function as:

$$G_{\text{diff}}(s) = \frac{\omega_{n,\text{diff}}^2 s}{s^2 + 2\zeta_{\text{diff}}\omega_{n,\text{diff}} s + \omega_{n,\text{diff}}^2} \tag{3.3}$$

Then define the position tracking error as $\tilde{P}_{err} = P_{com} - P_{nom}$. The error dynamics are $\dot{P}_{err} = \dot{P}_{sen} - \dot{P}_{com} = u_{ctrl}$. The proportional-integral (PI) control law for this loop is

$$u_{ctrl} = -K_{I_1} \int_0^t P_{err}(\tau)dt - K_{P_1} P_{err} \tag{3.4}$$

The close-loop state space error dynamic can be rewritten as

$$\begin{bmatrix} \dot{\tilde{z}} \\ \dot{P}_{err} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{m}K_{I_1} & -\frac{1}{m}K_{P_1} \end{bmatrix} \begin{bmatrix} \tilde{z} \\ P_{err} \end{bmatrix} \tag{3.5}$$

The controller coefficient $K_{I_1} = \omega_{n_1}^2$ and $K_{P_1} = 2\zeta_1\omega_{n_1}$, where ζ_i and ω_{n_i} are the closed-loop damping ratio and nature frequency which satisfy the desired dynamics. The output of this loop is

$$u_{com} = u_{nom} + u_{ctrl} \tag{3.6}$$

3.1.2 Guidance Inner Loop

Guidance inner loop takes velocity measurement and velocity command to calculate force command for the guidance allocation. The nominal force is calculated by inverting the second line of the Eq (3.1), $F_{nom} = m\dot{u}_{nom}$. Here $\dot{u}_{nom}$ is obtained by passing u_{nom} through a pseudo-differentiator Eq (3.3)

Define the tracking error $u_{err} = u_{sen} - u_{com}$, and the tracking error dynamics can be presented as

$$\dot{u}_{err} = \frac{1}{m} F_{ctrl} \tag{3.7}$$

The PI control law for this loop is

$$F_{ctrl} = -K_{I_2} \int_0^t u_{err}(\tau)dt - K_{P_2} u_{err} \tag{3.8}$$

The close-loop state space error dynamics can be rewritten as

$$\begin{bmatrix} \dot{z} \\ \dot{u}_{err} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{m}K_{I_2} & -\frac{1}{m}K_{P_2} \end{bmatrix} \begin{bmatrix} \tilde{z} \\ u_{err} \end{bmatrix} \tag{3.9}$$

Here the controller coefficient $K_{I_1} = \omega_{n_2}^2$ and $K_{P_2} = 2\zeta_2\omega_{n_2}$. The output of this loop is

$$F_{com} = F_{nom} + F_{ctrl} \tag{3.10}$$

3.1.3 Guidance Allocation

The relationship between longitudinal force and the motor voltage was presented in detail in Chapter 3, from Eqs (3.1),(3.2). The motor voltage can be calculated as:

$$E_{m_{com}} = \frac{J_m R_a}{K_m}\dot{\omega}_{m_{nom}} + \left(\frac{B_m R_a}{K_m} + K_m\right)\omega_{m_{nom}} - \frac{N R_{\text{eff}} R_a}{K_m} F_{xL} \tag{3.11}$$

where

$$F_{xL} = -F_{x_{com}} - C_{rr}mg - C_D A_F \frac{\rho}{2} u_{nom}^2 \tag{3.12}$$

and ω_{nom} is calculated by inverting Eq (3.2)

$$\omega_{nom} = -\frac{C_\sigma u_{nom}}{N R_{\text{eff}}}\left(F_{x_{com}} + C_{rr}mg + C_D A_F \frac{\rho}{2} u_{nom}^2 - C_\sigma\right) \tag{3.13}$$

and $\dot{\omega}_{nom}$ is obtained by passing ω_{nom} to a pseudo-differentiator Eq (3.3)

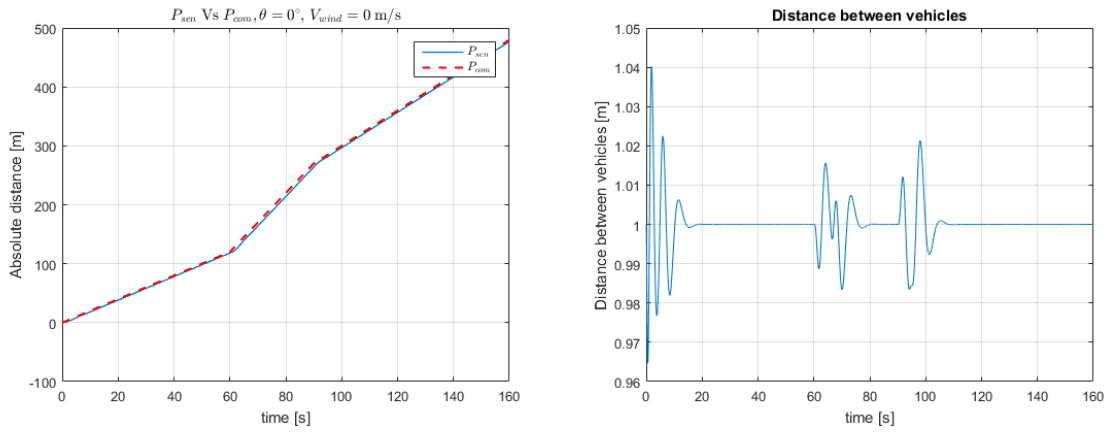
For the current work, the closed-loop dynamics for each loop is described by the characteristic equation $\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 = 0$. The value for each loop's ζ and ω_n are shown in Table 3.1, where the damping ratio coefficients ζ are set to 0.7 for all loops. The nature frequency ω_n for motor speed loop is chosen based on the experience of the motor property. Those for the outer loops are scaled down 3 times lower than the inner loop.

Table 3.1: Longitudinal Motion Controller Coefficients

	Distance Loop	Velocity Loop
λ	$\lambda_{1,2} = -0.7 \pm 0.7141i$	$\lambda_{1,2} = -2.1 \pm 2.1424i$
ζ	0.7	0.7
ω_n	1	3

3.2 Simulation and Results

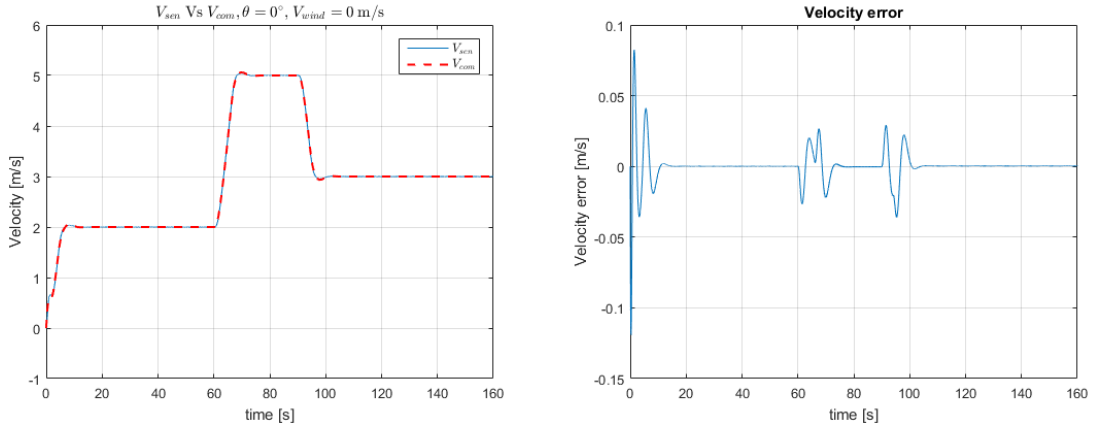
Figure 3.2 to 3.5 give the simulation results for longitudinal following without disturbance. The leading vehicle moves forward from static to 2m/s, and keep this speed for 60 seconds, and then accelerate to 5m/s and moving at 5m/s for 30s, finally decrease to 3m/s. The following vehicle moves forward but keep 1 meter behind the leading vehicle, and the ideal steady state is that it keeps the same velocity with the leading vehicle and 1 meters away from it. Figure 3.2 shows how far the vehicle travels and Figure 3.3 shows the velocity signals. The corresponding error signals show on the right side, Figure 3.2(b) and Figure 3.3(b).



(a) Absolute distance (b) distance between vehicles

Figure 3.2: Position Tracking

Here the distance between two vehicles is from the back side of the front vehicle to the front side of the following vehicle.



(a) Vehicle velocity (b) Vehicle velocity error

Figure 3.3: Vehicle Velocity Tracking

Figure 3.4 portrays the input voltage applied on the motor and Figure 3.5 is the motor rotational speed.

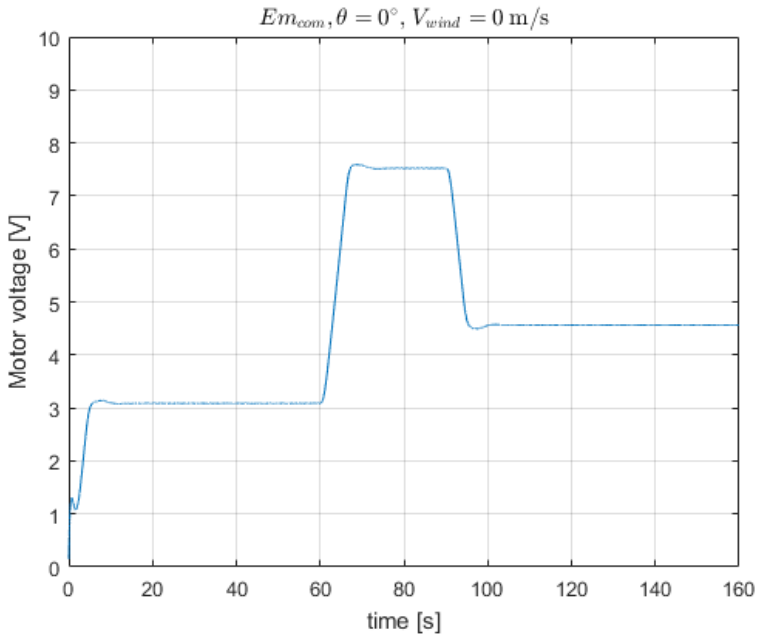


Figure 3.4: Input Voltage

The disturbance due to the effects of wind and road gradient are given as

$$\text{Dist} = C_{rr} mg\,(1 - \cos\theta) - mg\sin\theta - \frac{\rho}{2} C_D A_F \left(2u_{nom} V_{wind} + V_{wind}^2\right); \tag{3.14}$$

Figures 3.6 to 3.8 give the simulation results for the disturbed with $\theta = 15°$ and wind velocity $V_{wind} = 3$ m/s. Compare those two sets of results. First 3 pairs are similar; they can be verified from the error signals. But the input voltage has big difference, shown in Figure 3.8, which means more external disturbance, more power the vehicle needs to maintain the current state.

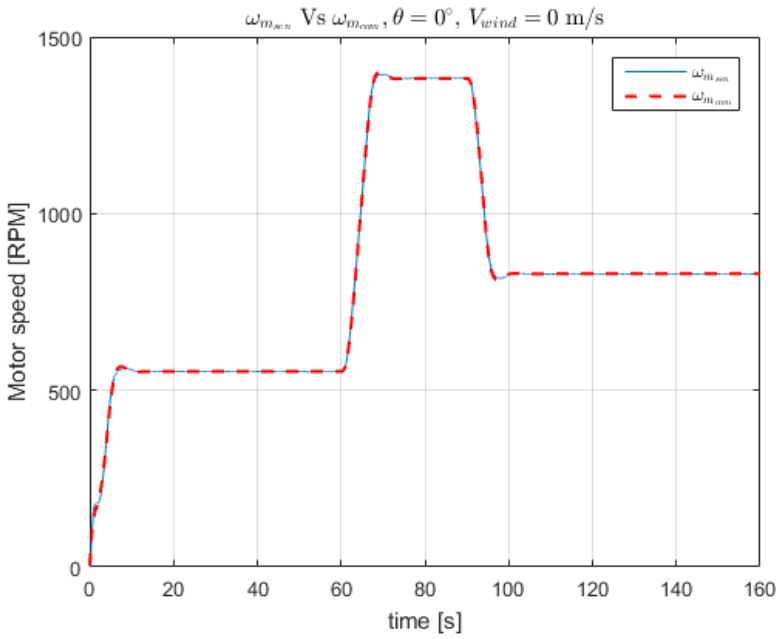
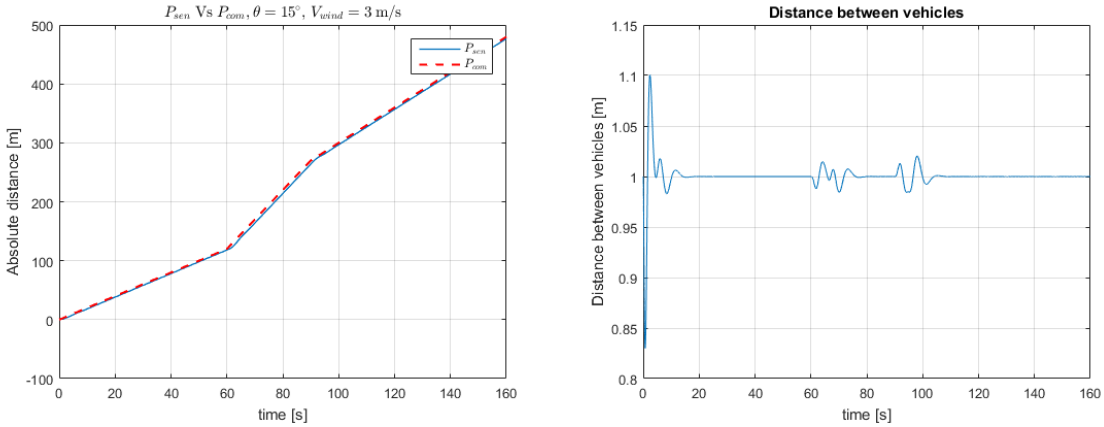


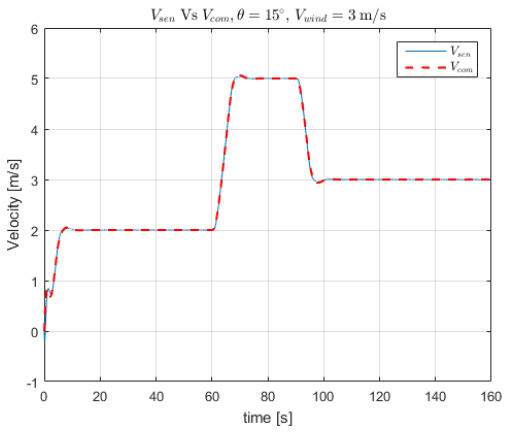
Figure 3.5: Motor Speed



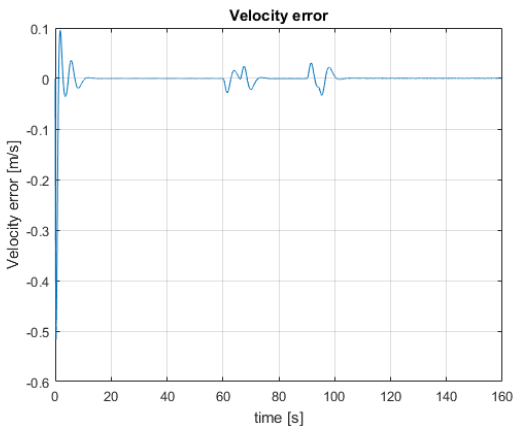
(a) Absolute distance

(b) Absolute distance error

Figure 3.6: Position Tracking

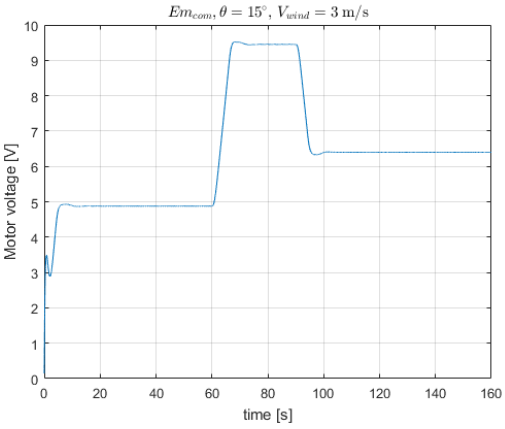


(a) Vehicle velocity

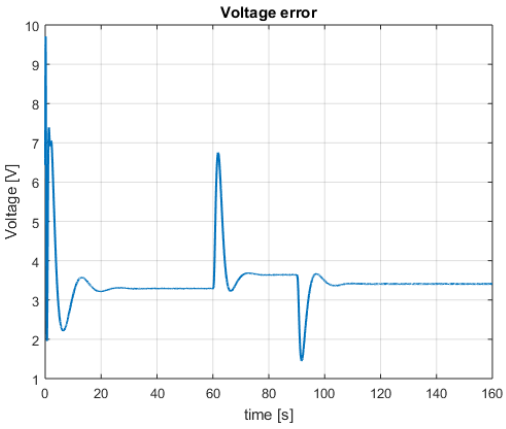


(b) Vehicle velocity error

Figure 3.7: Vehicle Velocity Tracking



(a) Input voltage



(b) Input voltage error

Figure 3.8: Input Voltage Tracking

In this chapter, a 1DOF vehicle trajectory tracking controller (ACC) was introduced. The main purpose of this design is to control the car at a prescribed constant cruise speed, and to speed up or slow down automatically in order to keep a prescribed safety distance

with the car in front of it. In our ACC design, the controller has the capability of keeping the cruise speed as well as the safety distance.

4 Autonomous Vehicle 3DOF Following

In this chapter, we will design a 3DOF controller for the car using the trajectory linearization control method. The controller for 3DOF car takes the position trajectory as command and calculates the corresponding voltage for the motor and the steering servo. The controller consists of two parts, guidance controller and attitude controller. The block diagram of the closed-loop system is shown in Figure 4.1

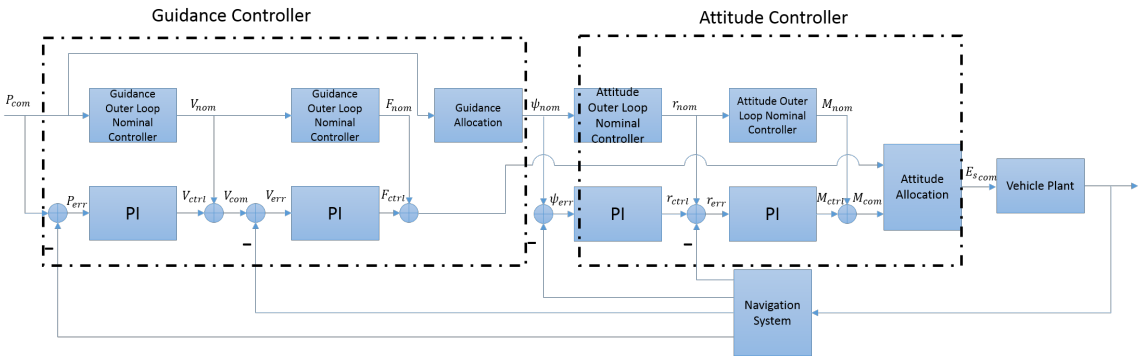


Figure 4.1: 3DOF Trajectory Linearization Controller Block Diagram

The overall closed-loop system consists of 4 loops. Each loop has the complete TLC structure as shown in Figure 1.5. The dynamic pseudo-inverse of the corresponding EOMs is used to generate the nominal control signal, and the PI feedback controller is used to stabilize the error.

4.1 Guidance Controller

The guidance controller consists of two loops: guidance outer loop and guidance inner loop, which control the position and body velocity of the vehicle respectively.

4.1.1 Guidance Outer Loop

The guidance outer loop takes position command and position measurement to calculate the velocity command for the inner loop. It has a nominal controller which is the dynamic pseudo-inverse of the EOM and a feedback PI controller.

The nominal body velocity can be obtained by inverting Eq (2.10) as $V_{nom} = B_1^{-1}(\gamma_{nom})\dot{P}_{nom}$, where $B_1(\gamma_{nom})$ is the matrix in Eq (2.10) with ψ replaced by its nominal value ψ_{nom}, and $\dot{P}_{nom}$ is obtained from position command by a pseudo-differentiator (a second-order low pass filter)

$$G_{\text{diff}}(s) = \frac{\omega_{n,\text{diff}}^2 s}{s^2 + 2\zeta_{\text{diff}}\omega_{n,\text{diff}} s + \omega_{n,\text{diff}}^2} \tag{4.1}$$

So the equation for V_{nom} can be written as

$$\begin{bmatrix} u_{nom} \\ v_{nom} \end{bmatrix} = \begin{bmatrix} \cos\psi_{nom} & -\sin\psi_{nom} \\ \sin\psi_{nom} & \cos\psi_{nom} \end{bmatrix}^{-1} \begin{bmatrix} \dot{x}_{e,nom} \\ \dot{y}_{e,nom} \end{bmatrix} \tag{4.2}$$

Then we proceed to focus on guidance PI feedback controller. Define the position tracking error $P_{err} = P_{sen} - P_{com}$. The error dynamics is given as $\dot{P}_{err} = B_1(\psi_{nom})V_{ctrl}$. The PI control law is given as

$$V_{ctrl} = -K_{P1}P_{err} - K_{I1}\int P_{err} \tag{4.3}$$

where the PI gain matrix are designed as

$$\begin{aligned} K_{P1} &= -B_1^{-1}(\psi_{nom})A_{1.1} = \begin{bmatrix} a_{111}\cos\psi_{nom} & a_{121}\sin\psi_{nom} \\ -a_{111}\sin\psi_{nom} & a_{121}\cos\psi_{nom} \end{bmatrix} \\ K_{I1} &= -B_1^{-1}(\psi_{nom})A_{1.2} = \begin{bmatrix} a_{112}\cos\psi_{nom} & a_{122}\sin\psi_{nom} \\ -a_{112}\sin\psi_{nom} & a_{122}\cos\psi_{nom} \end{bmatrix} \end{aligned} \tag{4.4}$$

where $A_{1.k} = \text{diag}\begin{bmatrix} -a_{11k} & -a_{12k} & -a_{13k} \end{bmatrix}, k = 1, 2$ are time-varying controller parameters which describe the closed-loop dynamics.

$$\begin{aligned} a_{ij1} &= 2\zeta_j \omega_{n,j} \\ a_{ij2} &= \omega_{n,j}^2 \end{aligned} \tag{4.5}$$

where $\omega_{n,j}$ are nature frequency of each loop, and ζ_j are the constant damping ratio of the desired dynamics; $j = 1, 2$ for the x and y channel, and $i = 1, 2, 3, 4$ for the four different loops. The output of the guidance outer loop combines the nominal controller output and feedback controller output

$$V_{com} = V_{nom} + V_{ctrl} \tag{4.6}$$

4.1.2 Guidance Inner Loop

The guidance inner loop of the guidance controller takes velocity command V_{com} from the the guidance outer loop and velocity measurement V_{sim} to calculate the force command for the guidance loop control allocation. It has a complete TLC structure: a nominal controller and a feedback controller.

We will start the design from the nominal controller. The nominal force is calculated from the transnational dynamics Eq (2.11)

$$\begin{aligned} F_{nom} &= m\left[\dot{V}_{nom} - A_2 r_{nom} V_{nom}\right] \\ => \begin{bmatrix} F_{x_{nom}} \\ F_{y_{nom}} \end{bmatrix} &= m \begin{bmatrix} \dot{u}_{nom} - r_{nom} v_{nom} \\ \dot{v}_{nom} + r_{nom} u_{nom} \end{bmatrix} \end{aligned} \tag{4.7}$$

where $\dot{V}_{nom} = \begin{bmatrix} \dot{u}_{nom} & \dot{v}_{nom} \end{bmatrix}^T$ is obtained by passing the V_{nom} through a pseudo-differentiator Eq (4.1).

Define the velocity tracking error as $V_{err} = V_{sen} - V_{nom}$, then the error dynamics of the body velocity is given as:

$$\dot{V}_{err} = A_2\left(r_{nom}\right) V_{err} + \frac{1}{m} F_{ctrl} \tag{4.8}$$

where $A_2\,(r_{nom})$ is the matrix in Eq (2.11) with r replaced by its nominal value r_{nom}. The guidance inner loop PI control law is defined as

$$F_{ctrl} = -K_{P2}V_{err} - K_{I2}\int V_{err} \tag{4.9}$$

The body velocity PI parameters are given as

$$\begin{aligned} K_{P2} &= m\left[A_2\,(r_{nom}) - A_{2.1}\right] = m\begin{bmatrix} a_{211} & r_{nom} \\ -r_{nom} & a_{221} \end{bmatrix} \\ K_{I2} &= -mA_{2.2} = m\begin{bmatrix} a_{212} & 0 \\ 0 & a_{222} \end{bmatrix} \end{aligned} \tag{4.10}$$

The output of the guidance inner loop is then given by

$$F_{com} = F_{nom} + F_{ctrl} \tag{4.11}$$

4.1.3 Guidance Control Allocation

The guidance control allocation unit is used to calculate the attitude angle and the voltage applied on the motor. Under the rigid body and level ground assumptions, the rotational motion is constraint to yaw angle ψ. Thus, the roll angle ϕ and pitch angle θ can either be constrained to zero or treated as disturbance. The force F_x is allocated to the motor and F_y is allocated to the steering servo.

Generally, the course angle χ is defined as $\tan^{-1}\chi = \frac{\dot{x}_{e,nom}}{\dot{y}_{e,nom}}$ and $\chi = \psi + \beta$, where β is body side slip angle. Here, assume $\beta_{nom} = \frac{F_{y,nom}}{C_\alpha}$, then $\tan^{-1}\psi_{nom} = \frac{\dot{x}_{e,nom}}{\dot{y}_{e,nom}}$, which can be used as the input to the attitude outer loop.

The relationship between longitudinal force and the motor voltage was presented in detail in Chapter 3, from Eq (3.1) and (3.2). The motor voltage can be calculated as:

$$E_{m_{com}} = \frac{J_m R_a}{K_m}\dot{\omega}_{m_{nom}} + \left(\frac{B_m R_a}{K_m} + K_m\right)\omega_{m_{nom}} - \frac{NR_{\text{eff}}R_a}{K_m}F_{xL} \tag{4.12}$$

where

$$F_{xL} = -F_{x_{com}} - C_{rr}mg - C_D A_F \frac{\rho}{2} u_{nom}^2 \tag{4.13}$$

and ω_{nom} is calculated by inverting Eq (3.2)

$$\omega_{nom} = -\frac{C_\sigma u_{nom}}{NR_{\text{eff}}} \left(F_{x_{com}} + C_{rr}mg + C_D A_F \frac{\rho}{2} u_{nom}^2 - C_\sigma \right) \tag{4.14}$$

and $\dot{\omega}_{nom}$ is obtained by passing ω_{nom} through a pseudo differentiator (4.1)

4.2 Attitude Controller

The attitude controller also consists of two loops, an outer loop and an inner loop, which control the yaw angle ψ and the angular rate $\dot{\psi}$ of the vehicle. The procedure is very similar to the guidance controller design.

4.2.1 Attitude Outer Loop

The outer loop of the attitude controller takes attitude command from the guidance allocation and the attitude measurement to calculate the body angular rate command for the inner loop. By inverting Eq (2.12)

$$r_{nom} = \dot{\psi}_{nom} \tag{4.15}$$

where $\dot{\psi}_{nom}$ is obtained by passing ψ_{nom} through the pseudo-differentiator (4.1). The tracking error control law for this loop is

$$r_{ctrl} = -K_{P3}\psi_{err} - K_{I3} \int_0^t \psi_{err}(\tau) dt \tag{4.16}$$

where $\psi_{err} = \psi_{sen} - \psi_{nom}$ is the attitude tracking error. The control gains are given as

$$\begin{aligned} K_{P3} &= a_{331} \\ K_{I3} &= a_{332} \end{aligned} \tag{4.17}$$

The output of attitude outer loop is

$$r_{com} = r_{nom} + r_{ctrl} \tag{4.18}$$

4.2.2 Attitude Inner Loop

The attitude inner loop takes body angular rate command and body angular rate measurement to calculate the torque command for the attitude control allocation.

By inverting Eq (2.13), the nominal body torque is given as

$$M_{z_{nom}} = I_{zz}\dot{r}_{nom} \tag{4.19}$$

The PI control law for the inner loop is

$$M_{z_{ctrl}} = -K_{P4}r_{err} - K_{I4}\int_0^t r_{err}(\tau)dt \tag{4.20}$$

where $r_{err} = r_{sen} - r_{nom}$ is the body angular rate tracking error and the control gains are given as

$$\begin{aligned} K_{P4} &= I_{zz}a_{431} \\ K_{I4} &= I_{zz}a_{432} \end{aligned} \tag{4.21}$$

The output of the attitude inner loop is

$$M_{z_{com}} = M_{z_{nom}} + M_{z_{ctrl}} \tag{4.22}$$

4.2.3 Attitude Control Allocation

The actuator command $E_{com} = \begin{bmatrix} E_{m_{com}} & E_{s_{com}} \end{bmatrix}^T$ which are the voltages applied to the vehicle motor and steering servo, respectively. Here $E_{m_{com}}$ is determined from the longitudinal force command $F_{x_{com}}$ calculated from guidance loop, and $E_{s_{com}}$ is determined from torque command $M_{z_{com}}$.

The relationship between steering servo voltage input $E_{s_{com}}$ and steering angle δ_{com} is

$$\delta_{com} = \frac{\delta_{max}}{E_{s_{max}}}E_{s_{com}} \tag{4.23}$$

and steering angle δ_{com} can be calculated from moment torque $M_{z_{com}}$

$$\ddot{\psi} = -\frac{2l_fC_{\alpha f} - 2l_rC_{\alpha r}}{I_{zz}u}v - \frac{2l_f^2C_{\alpha f} + 2l_r^2C_{\alpha r}}{I_{zz}u}\dot{\psi} + \frac{2l_fC_{\alpha f}}{I_{zz}}\delta \tag{4.24}$$

In our vehicle modeling, we assumed $l_f = l_r = \frac{L}{2}$ and $C_{\alpha f} = C_{\alpha r} = \frac{1}{2}C_\alpha$, then Eq (4.24) can be simplified as

$$\ddot{\psi} = \frac{M_z}{I_{zz}} = -\frac{\frac{1}{2}L^2 C_\alpha}{I_{zz}u}\dot{\psi} + \frac{\frac{1}{2}LC_\alpha}{I_{zz}}\delta \tag{4.25}$$

so the steering angle

$$\delta_{com} = \frac{2M_{z_{com}}u + L^2 C_\alpha r_{nom}}{u_{nom}LC_\alpha} \tag{4.26}$$

here we substitute $\dot{\psi} = r_{nom}$.

4.3 Simulation and Results

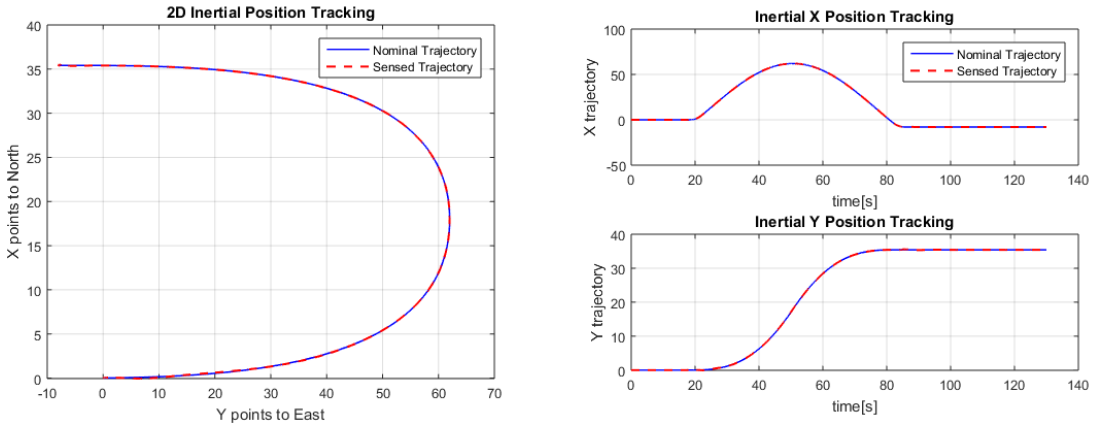
For the present work, constant controller parameters ζ and ω_n define the LTI closed-loop dynamics for each of the 4 loops with the characteristic equation $\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 = 0$. Table 4.1 shows the values of ζ and ω_n for each loop. Damping coefficients ζ are set to 0.7 for all 4 loops which consider the overshoot to be less then 5%. The natural frequency ω_n for the attitude inner loop is chosen based on the actuator bandwidth, and the outer loop are scaled down 4 times lower than its corresponding inner loop.

Table 4.1: Directional Motion Controller Coefficients

	Guidance Outer Loop	Guidance Inner Loop	Attitude Outer Loop	Attitude Inner Loop
λ	$\lambda_{1,2} =$ $-0.14 \pm 0.143i$	$\lambda_{1,2} =$ $-0.56 \pm 0.57i$	$\lambda_{1,2} =$ $-2.24 \pm 2.29i$	$\lambda_{1,2} =$ $-8.96 \pm 9.14i$
ζ	$\begin{bmatrix}0.7 & 0.7\end{bmatrix}$	$\begin{bmatrix}0.7 & 0.7\end{bmatrix}$	0.7	0.7
ω_n	$\begin{bmatrix}0.2 & 0.2\end{bmatrix}$	$\begin{bmatrix}0.8 & 0.8\end{bmatrix}$	3.2	12.8

Figure 4.2 to Figure 4.4 gives the Matlab/Simulink simulation results for 3DOF following motion. The 2D nominal trajectory command is shown in Figure 4.2(a). Figure

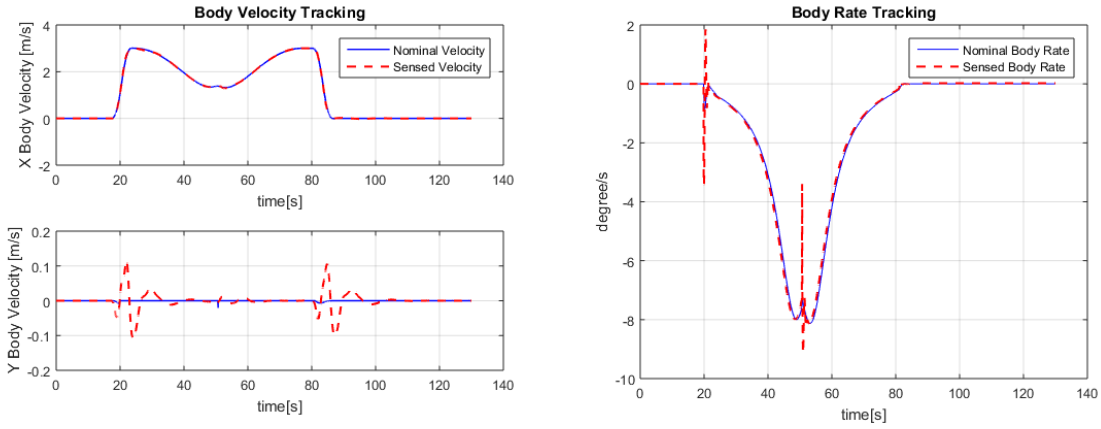
4.2(b) presents the position tracking performance for each states. Velocity tracking performance is shown in Figure 4.3(a). The longitudinal velocity $u(t)$ tracking is perfect, while the lateral velocity $v(t)$ exhibits oscillation; but it does not affect the position tracking due to small magnitude of the nominal signal. The attitude tracking performance is shown in Figure 4.4(a) which is very good. In the Figure 4.3(b), body rate tracking is given; the sensed signal has a small peak due to actuator noise.



(a) Trajectory Tracking 2D plot

(b) Trajectory Tracking Command

Figure 4.2: Directional Tracking Part 1



(a) Body Velocity Tracking (b) Body Rate Tracking

Figure 4.3: Directional Tracking Part 2

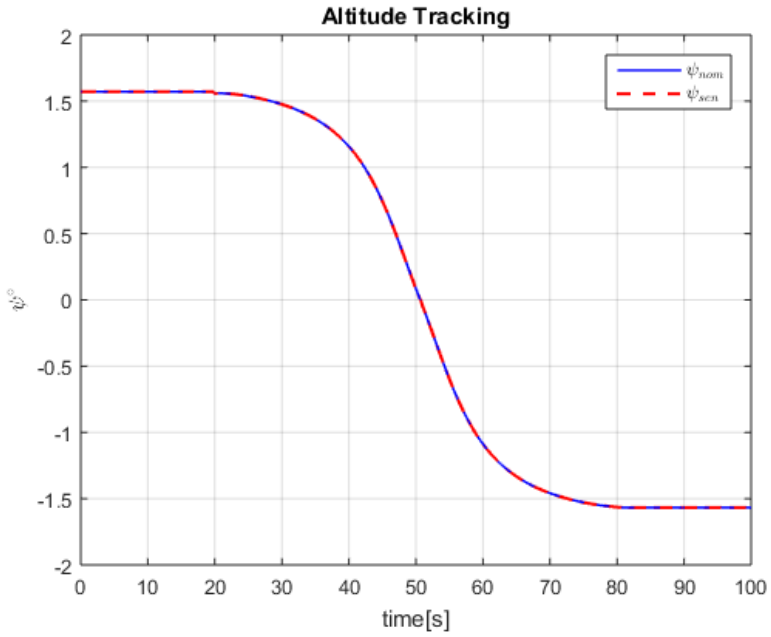


Figure 4.4: Attitude Tracking

3DOF following control is a crucial capability of the autonomous vehicle/self-driving cars in the near future in order to track a car in front or any feasible trajectory. Future work includes implementing the 3DOF controller presented on the OU ground vehicles Traxxas shown in Figure 1.6 for hardware in loop test. Parameters also need future tuning for actuator dynamics and modeling errors.

5 UGV Guidance For 3DOF Car Following

To achieve the research objective, an autonomous GN&C algorithm which makes a ground vehicle or mobile robot following a leading vehicle at a safety distance is needed. This algorithm can be used in many applications, either for cooperative motion such as a vehicle platoon, or passive motion, like adaptive cruise control, or even adversarial, like unmanned police car chasing. The 3DOF position tracking controller in Cartesian coordinate designed in chapter 4 has already existed, but it is not suitable for large tracking error. In this chapter, a guidance controller which is used to generate a guidance trajectory from the current vehicle position P_{sen} to the target position P_{tgt} is developed by using Pure Pursuit Guidance (PPG) [24]. Figure 5.1 is the diagram for guidance logic.

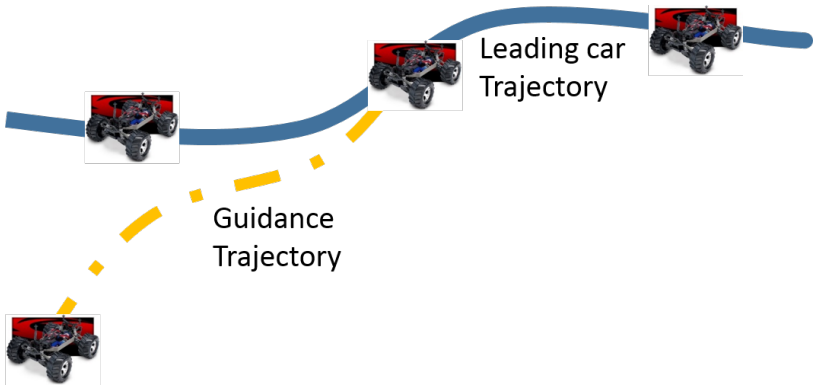


Figure 5.1: Diagram for Guidance Logic

5.1 Pure Pursuit Guidance

Pure pursuit guidance is a method which takes a traditional tracking-error-correction [24] approach to generate the guidance trajectory. This methodology produces guidance commands in relation to Line of Sight $\boldsymbol{L}$. Assume the UGV is pursuing a virtual way point with velocity of $\boldsymbol{V}$ in the inertial frame as portrayed in Figure 5.2, the following equation

satisfied if $\boldsymbol{L}$ and the velocity vector $\boldsymbol{V}$ are heading in the same direction

$$\boldsymbol{V} \times \boldsymbol{L} = 0 \tag{5.1}$$

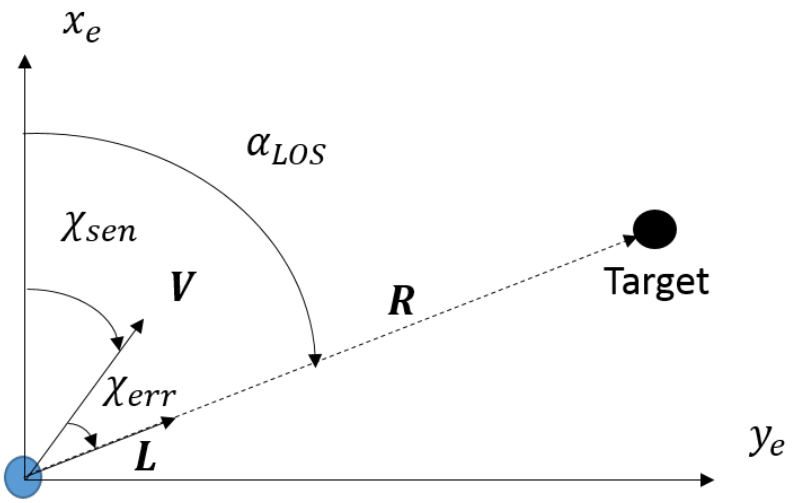


Figure 5.2: Position View of the UGV and the Target

Figure 5.3 presents the the trajectory of the following vehicle over several time steps. The vehicle starts from a location far away from the target and eventually converges to the desired trajectory. So the current work consists of stabilizing two control objectives to zero: 1) $R_{err} \to 0$ as $t \to \infty$; 2) $\chi_{err} \to 0$ as $t \to \infty$. In order to achieve the above objectives, two PID controllers are been chosen:

$$\begin{aligned} \dot{\chi}_g &= K_{i1} \int_0^t \chi_{err}(\tau)dt + K_{p1}\chi_{err} + K_{d2}\dot{\chi}_{err} \\ V_g &= K_{i2} \int_0^t R_{err}(\tau)dt + K_{p2}R_{err} + K_{d2}\dot{R}_{err} \end{aligned} \tag{5.2}$$

The main advantage of PPG control design is that the inertial position of the leading vehicle is not necessary, hence it will save the cost for the GPS sensor. Instead, a target seeker which consists of a servo-based camera and a laser range finder is needed. The target seeker locks the target and always maintain it in the center of the view by using

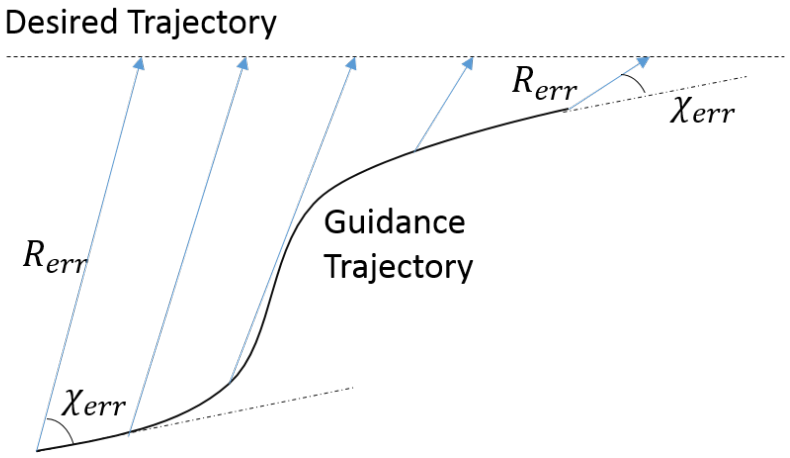
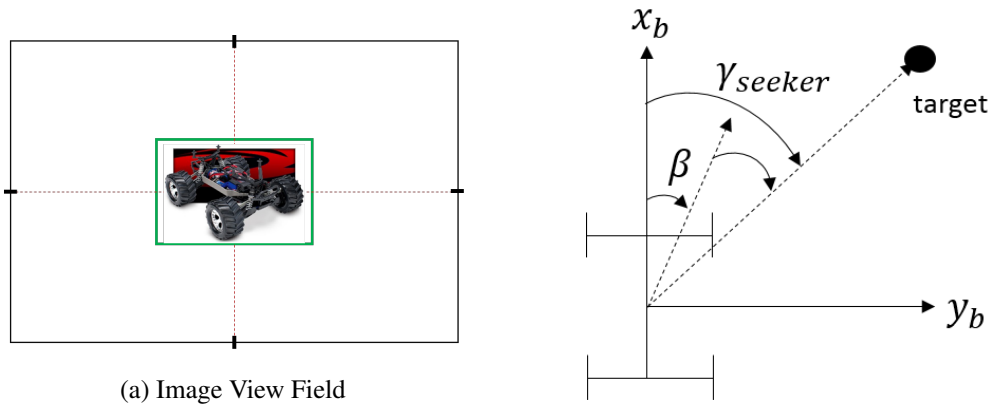


Figure 5.3: PPG Discrete Representation

image processing technology assisting with servo as shown in Figure 5.4a. The schematic diagram for the seeker under body-frame is shown in Figure 5.4b. Servo ration angle γ_{seeker} can be approximately equals to α_{LoS} and the laser range turn the range between vehicles back.



(a) Image View Field

(b) Target Seeker diagram

Figure 5.4: Target Seeker

Constrain the side slip angle β and R_{err} to 0, and then

$$\begin{aligned} \chi_{err} &= \gamma_{seeker} - \beta \\ R_{err} &= R_{laser} - C \end{aligned} \tag{5.3}$$

where γ_{seeker} is the servo rotation angle, R_{laser} is laser range data and C is the safety distance between vehicles. In general, C equals 6 to 10 times the length of the vehicle at different speed. In this project, we set $C = 2$ m.

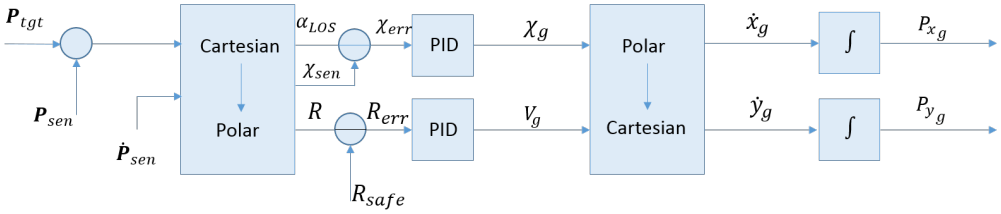


Figure 5.5: Diagram for Guidance Controller

For the MATLAB/SIMULINK implementation, we use the target position P_{tgt} and the vehicle sensed position P_{sen} to simulate target seeker. Figure 5.5 is the diagram for guidance controller. Heading angle error χ_{err} and range error R_{err} signals can be obtained by transferring the position signals in Cartesian to polar coordinate using Eq (5.4).

$$\begin{aligned} R &= \sqrt{\left(P_{x_{tgt}} - P_{x_{sen}}\right)^2 + \left(P_{y_{tgt}} - P_{y_{sen}}\right)^2} \\ \alpha_{LoS} &= \arctan \frac{P_{y_{tgt}} - \dot{P}_{y_{sen}}}{P_{x_{tgt}} - \dot{P}_{x_{sen}}} \\ \chi_{sen} &= \arctan \frac{\dot{P}_{y_{sen}}}{\dot{P}_{x_{sen}}} \\ \chi_{err} &= \alpha_{LoS} - \chi_{sen} \end{aligned} \tag{5.4}$$

The velocity vector can be obtained by converting the polar coordinate back to Cartesian coordinate Eq (5.5)

$$\begin{aligned} P_{x_g} &= V_g \sin \chi_g \\ P_{y_g} &= V_g \cos \chi_g \end{aligned} \tag{5.5}$$

Based on the tuning result, the parameters for the PID controller can be set as in Table 5.1. where PID1 is for the angle control and PID2 is for velocity control.

Table 5.1: PID Parameters for Guidance Trajectory Design

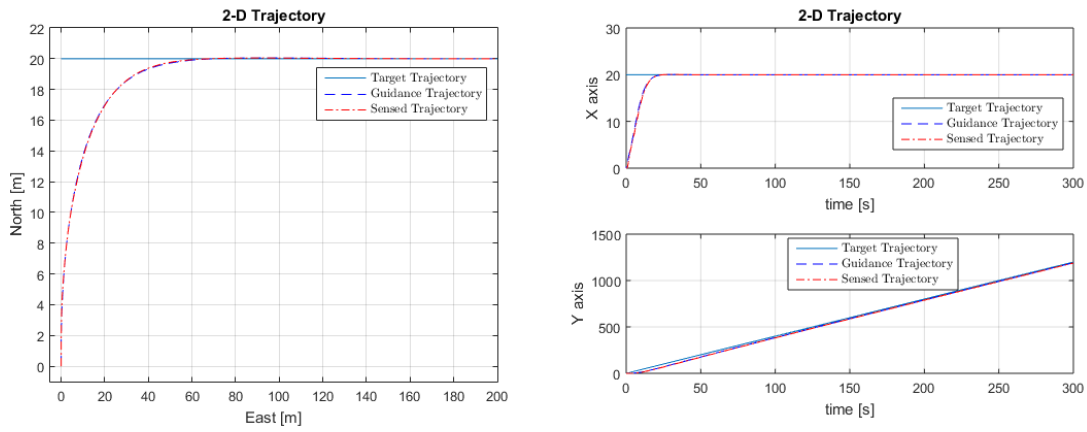
	K_p	K_i	K_d
PID1	0.35	0.01	0.1
PID2	0.25	0.004	0.5

5.2 Simulation and Results

In this section, the MATLAB/SIMULINK results for the guidance controller are present. There are three test cases: straight line guidance tracking, semi circle guidance tracking and S-shape guidance tracking.

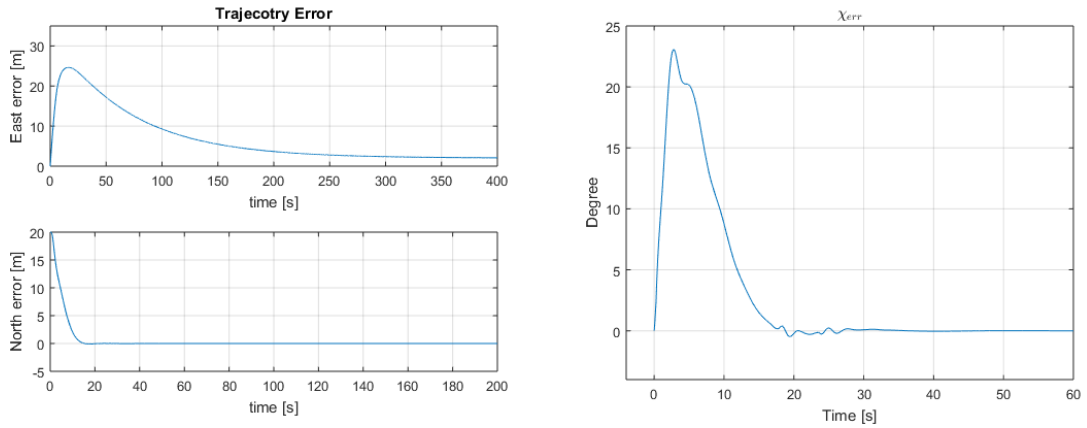
5.2.1 Straight Line Tracking

Figure 5.6 to Figure 5.9 are the MATLAB/SIMULINK results for the straight line guidance tracking. Figure 5.6 represents the trajectory tracking: the leading vehicle, which is the blue line in the plot, is moving East with a speed of 4 m/s; it starts at 20 meters to the north. The blue dotted line is the controller generated guidance trajectory and the red dotted line is the following vehicle real trajectory.



(a) Trajectory Tracking (b) Trajectory State Tracking

Figure 5.6: Straight Line Trajectory Tracking 1



(a) Trajectory tracking error (b) Heading angle error

Figure 5.7: Straight Line Trajectory Tracking 2

The trajectory tracking error is shown in Figure 5.7(a). Figure 5.7(b) portrayed the heading angle error χ_{err}. At the very beginning, the following vehicle points to the North, so χ_{err} is 0 at starting point. As both vehicles move, χ_{err} increases and then reduces as the following vehicle turns to the East.

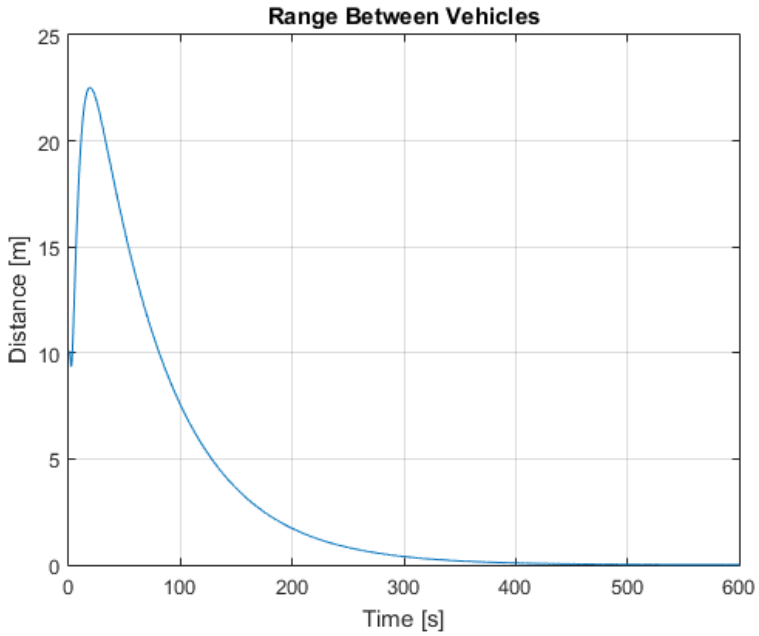


Figure 5.8: Range Error R_{err}

Range error R between two vehicles in Figure 5.8 has the similar tendency with χ_{err}. Figure 5.9 represents the following vehicle velocity; it speeds up at first and finally stays at $4m/s$ which is same as the leading vehicle.

5.2.2 Semi Circle Guidance Tracking

In this section, we will give a set of MATLAB/SIMULINK simulation results for semi circle tracking as shown in Figure 5.10 to Figure 5.13.

Figure 5.10 represents the trajectory tracking. Solid blue line is the leading vehicle's trajectory; it first moves as a semi circle and then move towards North. The controller

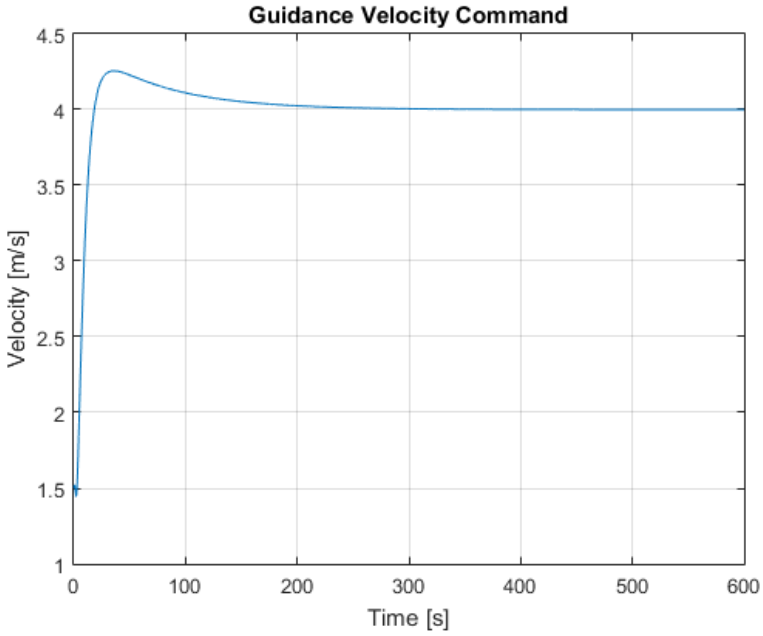
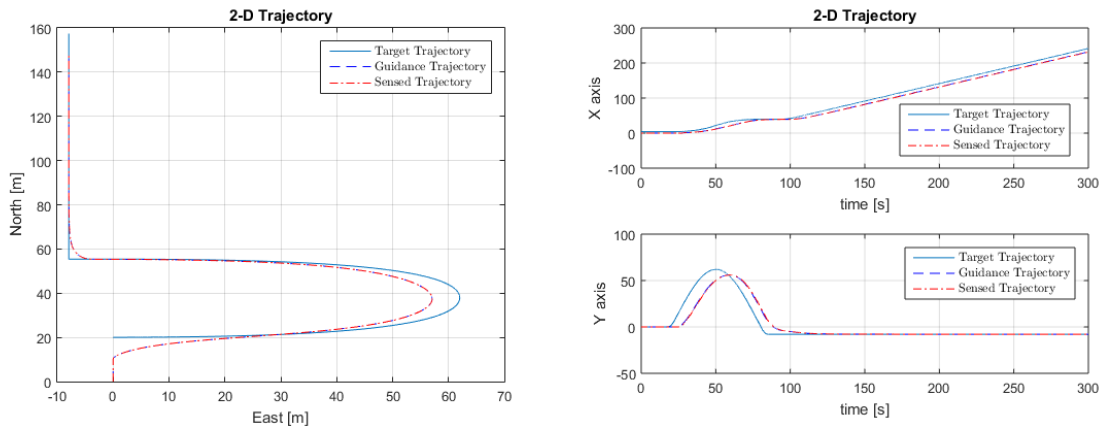


Figure 5.9: Following Vehicle Speed

generates a smooth guidance trajectory which is given as blue dot line, and the red dot line is the following vehicle's trajectory. The following vehicle does not track the leading car accurately but attempts a short cut trajectory instead, which can save power.



(a) Semi Circle Tracking

(b) Semi Circle State Tracking

Figure 5.10: Semi Circle Guidance Tracking

The leading vehicle's heading angle changes between $\begin{bmatrix} -90^\circ & 90^\circ \end{bmatrix}$, while the following vehicle changes between $\begin{bmatrix} -80^\circ & 80^\circ \end{bmatrix}$ as shown in Figure 5.12. Heading angle error χ_{err} and the range distance error converges to 0 as time goes to infinity, shown in Figure 5.11 and 5.13.

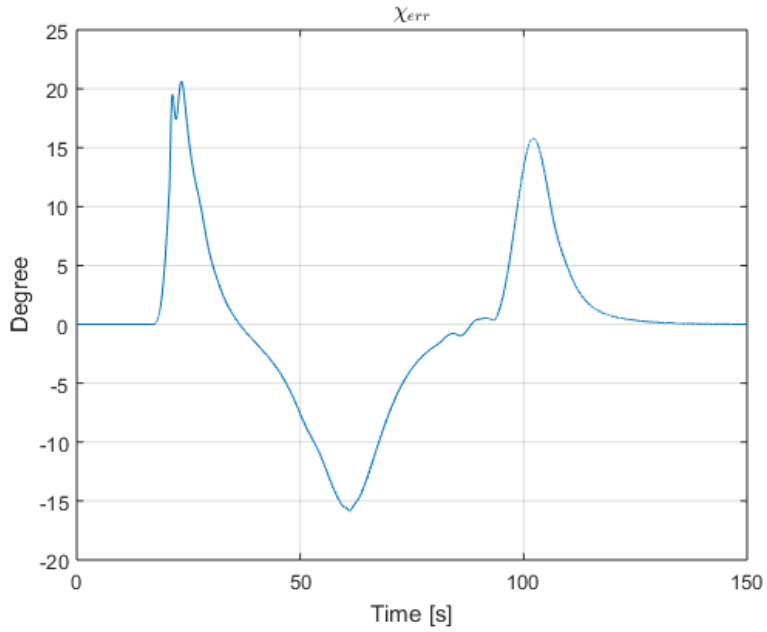


Figure 5.11: Semi Circle Line of Sight Angle σ

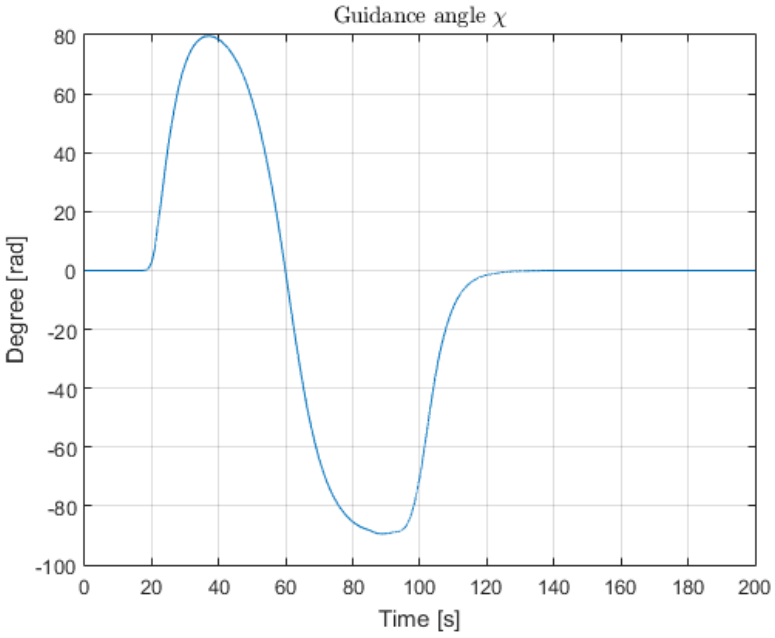


Figure 5.12: Semi Circle Heading Angle χ

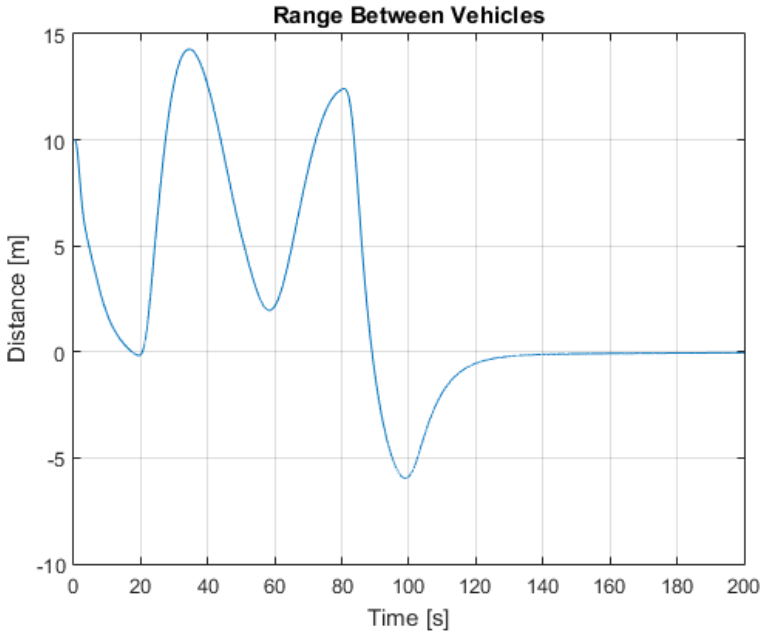
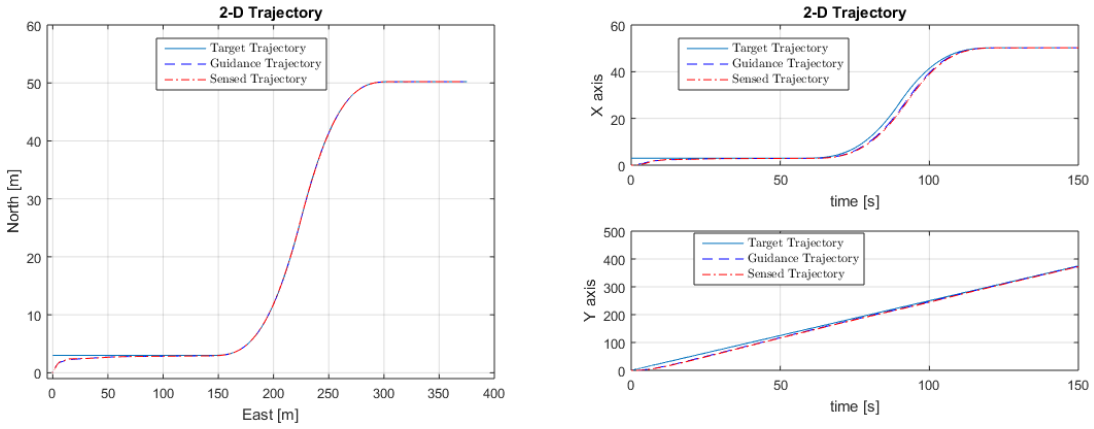


Figure 5.13: Range Error Between Vehicles

5.2.3 S-shape Guidance Tracking

Figure 5.14 to Figure 5.17 portrayed S-shape tracking. Figure 5.14 represents the trajectory. Solid blue line is the leading vehicle's trajectory. The trajectory has smooth turns, and the following vehicle tracks perfectly, see Figure 5.15 tracking error plot. Figure 5.16 (a) represents the following vehicle's heading angle and Figure 5.16 (b) is heading angle error χ_{err}. χ_{err} changes sharply at beginning time, since at very beginning, leading vehicle is moving at a constant speed while the following vehicle starting at 0m/s, then heading angle error changes sharply during this time interval; this will be improved in the future tune. Figure 5.17 shows the range error command between vehicles; the peak at t=20 s is introduced by the initial value setting; the initial position for the leading vehicle is $\begin{bmatrix} 10 & 0 \end{bmatrix}$ m.



(a) S-shape Tracking (b) S-shape State Tracking

Figure 5.14: S-shape Guidance Tracking

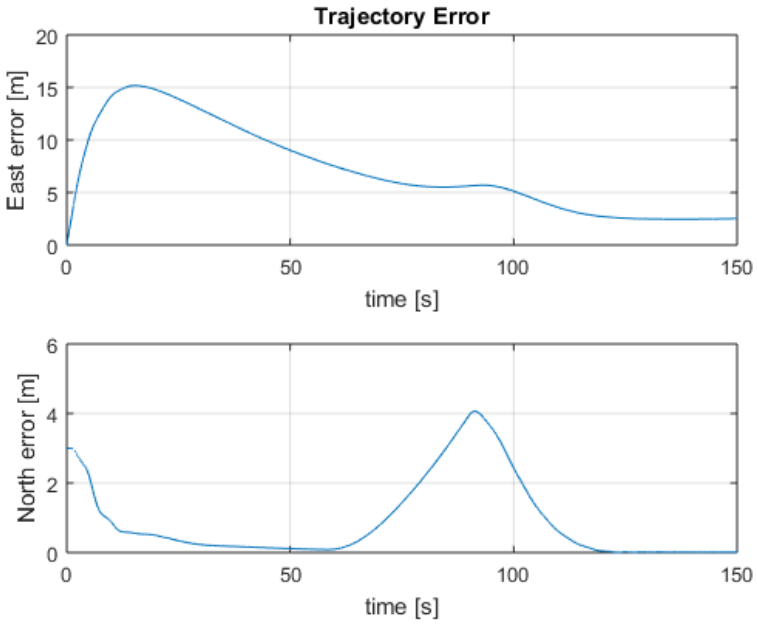
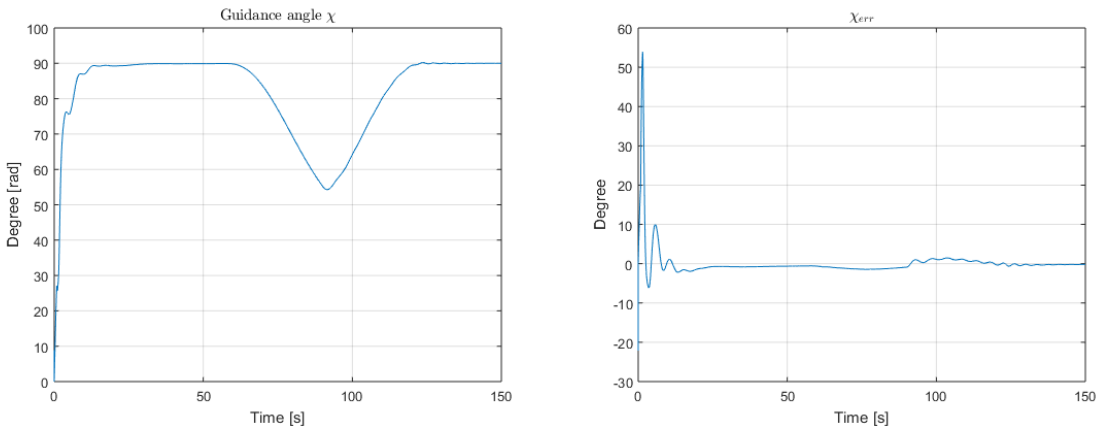


Figure 5.15: S-shape Tracking Error



(a) Heading Angle (b) S-shape State Tracking

Figure 5.16: S-shape Guidance Tracking

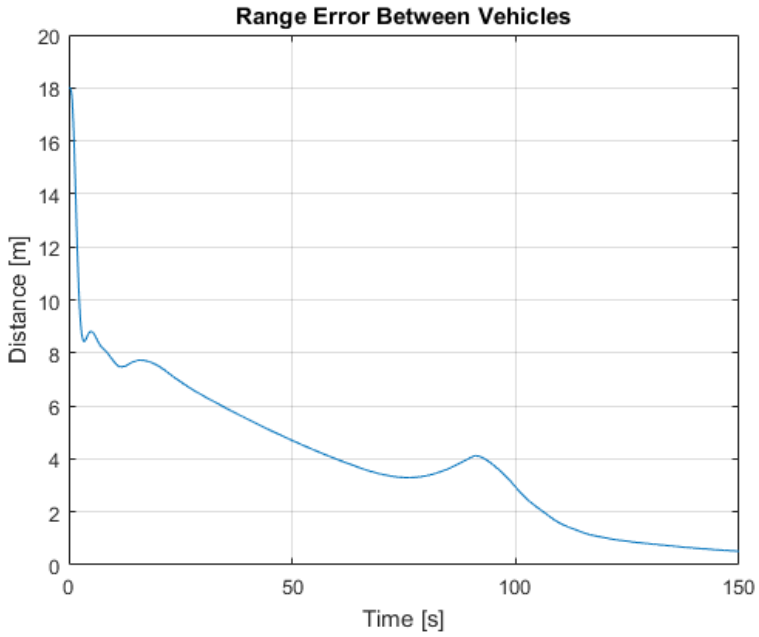


Figure 5.17: Range Error Between Vehicles

In this chapter, we used PPG method to generate a feasible guidance trajectory for 3DOF trajectory tracking controller use. In our design, the guidance controller and the position tracking controller are two independent parts; it is the main difference from the traditional PPG method.

6 Conclusion and Future Work

The main results of this thesis are a 3DOF TLC vehicle controller design and simulation testing for the autonomous ground vehicle. It is the first time that TLC has been applied to a 3DOF ground vehicle control and verified with simulations. Current work addresses the rigid body car model directly.

Future work is needed to perform in-depth research on the stability of the nonlinear controller and plant by using advanced methods, like applying passivity to prove the stability of the system and using gain scheduling to optimize the guidance performance of the results. Moreover, sensitivity studies should be performed by integrating different error sources, like measuring noise, communication noise and internal disturbance, etc. Those analysis would use random perturbation or simulated signals.

To facilitate the 3DOF vehicle controller, a real time simulation plant based on the OU RC car will be built. The car will be mounted on a bearings-frame which can make the car complete 3DOF movement but still keep static relative to the ground. The RC car has a wireless link to a ground command station (GCS) such that any computer equipped with MATLAB software and an RF modem can act as a GCS. The GCS will act as a sender to send the cars commands and monitor the cars during operation.

REFERENCES

[1] Luettel, Thorsten, Michael Himmelsbach, and Hans-Joachim Wuensche. "Autonomous ground vehicles concepts and a path to the future." Proceedings of the IEEE 100.Special Centennial Issue, 2012.

[2] Luettel, Thorsten, et al. "Autonomous offroad navigation under poor GPS conditions." Proceedings of 3rd Workshop On Planning, Perception and Navigation for Intelligent Vehicles (PPNIV), IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009.

[3] Balch, Tucker, and Ronald C. Arkin. "Behavior-based formation control for multirobot teams." Robotics and Automation, IEEE Transactions on 14.6, 1998.

[4] Wallace, Richard, et al. "First Results in Robot Road-Following." IJCAI, 1985.

[5] Klappstein, Jens, Fridtjof Stein, and Uwe Franke. "Monocular motion detection using spatial constraints in a unified manner." Intelligent Vehicles Symposium, 2006 IEEE. IEEE, 2006.

[6] Adaptive Cuise Control. Retrieved from https://en.wikipedia.org/wiki/Autonomous_cruise_control_system (Accessed on 03/16/2016)

[7] Google self driving car test. Retrieved from https://www.youtube.com/watch?v=cdgQpa1pUUE (Accessed on 03/28/2012)

[8] State-Space Reprentation for LTI Systems. Retrieved from http://web.mit.edu/2.14/www/Handouts/StateSpace.pdf (Accessed Oct 2002)

[9] Urmson, Chris, et al. "Autonomous driving in urban environments: Boss and the urban challenge." Journal of Field Robotics 25.8, 2008.

[10] Loose, Heidi, and Uwe Franke. ”B-spline-based road model for 3d lane recognition.” Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on. IEEE, 2010.

[11] Galluzzo, T. ”Simultaneous Planning and Control for Autonomous Ground Vehicles”, 2006.

[12] Oishi, Shuji, et al. ”ND voxel localization using large-scale 3D environmental map and RGB-D camera.” Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on. IEEE, 2013.

[13] Yang, Xinxin, et al. ”An intelligent predictive control approach to path tracking problem of autonomous mobile robot.” Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on. Vol. 4. IEEE, 1998.

[14] Normey-Rico, Julio E., Juan Gmez-Ortega, and Eduardo F. Camacho. ”A Smith-predictor-based generalised predictive controller for mobile robot path-tracking.” Control Engineering Practice 7.6, 1999.

[15] Boyden, F. Demick, and Steven Velinsky. ”Dynamic modeling of wheeled mobile robots for high load applications.” Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on. IEEE, 1994.

[16] Sarkar, Nilanjan, Xiaoping Yun, and Vijay Kumar. ”Dynamic path following: A new control algorithm for mobile robots.” Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on. IEEE, 1993.

[17] Kim, B., D. Necsulescu, and J. Sasiadek. ”Model predictive control of an autonomous vehicle.” Advanced Intelligent Mechatronics, 2001. Proceedings. 2001 IEEE/ASME International Conference on. Vol. 2. IEEE, 2001.

[18] Coelho, Paulo, and Urbano Nunes. ”Path-following control of mobile robots in presence of uncertainties.” Robotics, IEEE Transactions on 21.2, 2005.

[19] Adami, Tony M., and J. Jim Zhu. ”6DOF flight control of fixed-wing aircraft by trajectory linearization.” American Control Conference (ACC), 2011. IEEE, 2011.

[20] Liu, Yong, et al. ”Omni-directional mobile robot controller based on trajectory linearization.” Robotics and Autonomous Systems 56.5, 2008.

[21] Huang, Rui, Yong Liu, and J. Jim Zhu. ”Guidance, Navigation, and Control System Design for Tripropeller Vertical-Take-Off-and-Landing Unmanned Air Vehicle.” Journal of Aircraft 46.6, 2009.

[22] Zhu, J., Brad D. Banker, and Charles E. Hall. ”X-33 ascent flight controller design by trajectory linearization-a singular perturbational approach.” AIAA-2000-4159, Proceedings, AIAA Guidance, Navigation and Control Conference, 2001.

[23] Williams, Robert L., and Douglas A. Lawrence. *Linear state-space control systems*. John Wiley & Sons, 2007.

[24] Yamasaki, Takeshi, Hiroyuki Takano, and Yoriaki Baba. ”Robust path-following for UAV using pure pursuit guidance”. Ed. T. M. Lam. INTECH Open Access Publisher, 2009.

[25] Rajamani, R. *Vehicle Dynamics and Control*. Springer Science & Business Media, 2011.

[26] Wong, Jo Yung. *Theory of ground vehicles*. John Wiley & Sons, 2001.

[27] Tipler, Paul A., and Gene Mosca. *Physics for scientists and engineers*. Macmillan, 2007.

[28] Khalil, Hassan K., and J. W. Grizzle. *Nonlinear systems*. New Jersey: Prentice hall, 1996.

Thesis and Dissertation Services