Model Reference Adaptive Backstepping Control of an Autonomous Ground Vehicle

Labiba Quaiyum

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Mechanical Engineering

Alexander Leonessa, Chair
Steve Southward, Committee Member
Andrew Kurdila, Committe Member

November 30, 2015
Blacksburg, Virginia

Keywords: Adaptive Control, Backstepping, Nonholonomic Robot, Unicycle model

Model Reference Adaptive Backstepping Control of an Autonomous Ground Vehicle

Labiba Quaiyum

Abstract

With an increased push for commercial autonomous cars, the demand of high speed systems capable of performing in unstructured driving environments is growing. In this thesis, the behavior of a bio-inspired predator prey model is considered to stimulate a more organic response to obstacles and a moving target than existing algorithms. However, the current predator prey model has a disconnect between the desired velocities commanded and the torque signals provided to the motors due the dynamics of the vehicle not accounted for. This causes the vehicle to derail from its intended trajectory at sharp turns.

In this study, we start by adding dynamic behavior to the unicycle model to account for the varying dynamics of the vehicle. A backstepping algorithm is developed to connect the predator-prey model commanding desired velocities to an appropriate torque controller for the motors of the vehicle. To account for the unknown dynamic model parameters an adaptive control approach is utilized. Three different controllers are developed and evaluated.

Out of the three, the indirect MRAC backstepping controller is deemed unsuitable due to its limitations with handling unknown parameter structure. The direct MRAC backstepping is deemed suitable and therefore simulated and implemented on the vehicle. The newly derived controller is able to overcome the disconnect and allow the vehicle to optimally track its trajectory for a velocity range of $1\frac{m}{s}$ to $9\frac{m}{s}$ despite varying dynamics. Lastly, the $\mathcal{L}_1$ adaptive backstepping controller is introduced and simulated to provide an alternative, more robust solution to the direct MRAC backstepping controller.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Autonomous driving is considered to be the next big thing expected to revolutionize the everyday life of millions worldwide. Advantages include fewer traffic collisions, reduced traffic congestion, relief of vehicle occupants from driving, removal of constraints of occupants ability and state, and many more. Numerous research groups all over the world have allocated resources, time and energy in developing autonomous cars for commercial and personal use. With the current demand and extensive work being done on autonomous car worldwide, there has been an increasing demand for high speed systems capable of performing in unstructured driving environments.

1.1 Motivation

Many of the commonly used obstacle avoidance methods aim to generate a path and coerce the vehicle to follow the desired trajectory using potential-field, grid-based, sample-based or discrete optimization approaches, just to mention a few ([1]). However, in many cases the desired trajectory for the path-following algorithms is not smooth and does not account for the limits of the vehicle's physical capability ([1, 2]). To look for a more organic behavior we consider a bio-inspired example of predator-prey interaction. A cheetah does not mimic the path of his prey, but creates a relatively smooth path even if his prey is traveling in a chaotic fashion. This study begins with the framework laid out in [3], where a relationship similar to the predator-prey interaction is introduced and developed. In [3, 4], the authors consider a unicycle model for its simplicity and governing constraints. The Ackermann steering constraint enforces realistic restrictions on the achievable longitudinal and angular velocities, hence it is used to develop a velocity saturation for the predator-prey control model. The control law commands the velocity and angular rate, which allows the vehicle to track a virtual target at a time-varying distance ([4]). However, there is a disconnect between the desired velocity and angular rate with the actual torque signal implemented on the vehicle's

two motors ([4]). In particular, in [4], a low-level Proportional-Integral controller is used to track the commanded velocity as well as angular rate to provide appropriate torque signals to the vehicle's motors. However, the above method does not take into account the vehicle's velocity-dependent dynamics. To correct this shortcoming, an adaptive backstepping algorithm is proposed which will adapt according to the varying dynamics of the vehicle and command more appropriate torque signals to the motors.

1.2 Model Definition

In this section, the details regarding the mathematical model used for the implementation of the adaptive backstepping control algorithm will be described. The details include the description of the kinematic model for the vehicle, the dynamic model for the vehicle, and the saturation constraints introduced to enforce the desired behavior of the predator-prey interaction.

1.2.1 Kinematics and Dynamics of the Vehicle

The unicycle model considered for the controller design is given by [5]

$$\dot{p}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} v(t)\cos(\theta(t)) \\ v(t)\sin(\theta(t)) \end{bmatrix}, \tag{1.1}$$

$$\dot{\theta}(t) = \omega(t), \tag{1.2}$$

where $p(t) \triangleq [x(t)\ y(t)]^{\mathrm{T}} \in \mathbb{R}^2$ and $\theta(t) \in \mathbb{R}$ is the vehicle's position and orientation in the world reference frame, and the pair $(v(t), \omega(t)) \in \mathbb{R}^2$ is the input control encompassing the longitudinal and angular velocities, respectively.

The unicycle model is concerned with the geometric description of the trajectory disregarding where the forces producing the motion originated from in the vehicle. In order to account for vehicle dynamics, we consider the Newton's Second law of motion. The translational and rotational dynamics of the vehicle can be described as given by

$$m\dot{v}(t) = f(t) - c_v v(t), \tag{1.3}$$

$$j\dot{\omega}(t) = q(t) - c_w \omega(t), \tag{1.4}$$

where m is the vehicle's mass, j is the vehicle's inertia moment, $f(t)$ represents forces applied to the vehicle, $q(t)$ is the torque applied to the vehicle, c_v is the translational damping coefficient, and c_w is the rotational damping coefficient.

The motor signals are $\tau_{\mathrm{P}}(t)$ and $\tau_{\mathrm{s}}(t)$, where $\tau_{\mathrm{P}}(t)$ is the signal commanded to the forward motor of the vehicle and $\tau_{\mathrm{s}}(t)$ is the signal commanded to the steering motor of the

vehicle. The relationship between motor signals and forces are defined using

$$f(t) = g_v \tau_{\mathrm{P}}(t), \tag{1.5}$$
$$q(t) = g_w \tau_{\mathrm{s}}(t), \tag{1.6}$$

where g_v, and g_w are constants. Therefore, we have the following dynamic model for the vehicle given by

$$m\dot{v}(t) + c_v v(t) = g_v \tau_{\mathrm{P}}(t), \tag{1.7}$$
$$j\dot{\omega}(t) + c_w \omega(t) = g_w \tau_{\mathrm{s}}(t). \tag{1.8}$$

As seen in most literature, the dynamic model (1.7) and (1.8) can be rewritten as

$$M\dot{s}(t) + Cs(t) = G\tau(t), \tag{1.9}$$

where $M \in \mathbb{R}^{2\times 2}$ and $M > 0$ is the mass matrix, $s = [v(t); \omega(t)]^{\mathrm{T}}$ is the vehicle's velocity vector, $v(t) \in \mathbb{R}$ being the longitudinal velocity and $\omega(t) \in \mathbb{R}$ being the vehicle's angular velocity, $C \in \mathbb{R}^{2\times 2}$ is the damping matrix, $G \in \mathbb{R}^{2\times 2}$ is the input transformation matrix, and $\tau = [\tau_{\mathrm{P}}(t); \tau_{\mathrm{s}}(t)]^{\mathrm{T}}$ is the vehicle's motor signal vector.

In order, to simplify the model for the purposes of our problem, we will rearrange the above equation as shown below

$$\dot{s}(t) = As(t) + B\tau(t) \tag{1.10}$$

where $A \triangleq -M^{-1}C$ and $B \triangleq M^{-1}G$.

1.2.2 Saturation Constraints

To ensure that the controller proposed in [4] is able to better mimic the behavior of a predator chasing its prey, the following saturation constraints are introduced based on the Ackermann steering platform.

$$\left|\frac{\omega(t)}{v(t)}\right| \leq \frac{\tan(\phi_{\max})}{L}, \tag{1.11}$$

$$v_{\min} \leq v(t) \leq v_{\max}, \tag{1.12}$$

where $L > 0$ is the wheelbase length from rear to front of axle, $\phi_{\max} > 0$ is the maximum steering angle deflection refereed from the longitudinal body, $v_{\min}$ is a lower bound on the longitudinal velocity, and $v_{\max}$ is the upper bound on the longitudinal velocity.

More details on the need and implementation of the saturation constraints will be elaborated in [4].

1.3 Problem Formulation

In this section, the disconnect between control approach shown in [4] and the vehicle's varying dynamics is discussed, and an approach is developed for solving the problem.

1.3.1 PI Controller: Implementation and Results

In [4], a PI controller was used to provide the appropriate torque signals to the vehicle's motors to achieve the desired velocity and angular velocity. The algorithm implemented is shown below

$$\tau_{\mathrm{P}}(t) = k_{\mathrm{p,p}}(v_{\mathrm{des}}(t) - v(t)) + k_{\mathrm{i,p}} \int_0^t [v_{\mathrm{des}}(t) - v(t)]dt, \tag{1.13}$$

$$\tau_{\mathrm{s}}(t) = k_{\mathrm{p,s}}(\omega_{\mathrm{des}}(t) - \omega(t)) + k_{\mathrm{i,s}} \int_0^t [\omega_{\mathrm{des}}(t) - \omega(t)]dt, \tag{1.14}$$

where $\tau_{\mathrm{P}}(t) \in \mathbb{R}$ is the signal commanded to the forward motor of the vehicle, $\tau_{\mathrm{s}}(t) \in \mathbb{R}$ is the signal commanded to the steering motor of the vehicle, $v_{\mathrm{des}}(t) \in \mathbb{R}$ is the desired longitudinal velocity, $\omega_{\mathrm{des}}(t) \in \mathbb{R}$ is the desired angular velocity, $k_{\mathrm{p}} \in \mathbb{R}$ is the proportional gain, and $k_{\mathrm{i}} \in \mathbb{R}$ is the integral gain.

In addition to the above algorithm, gain scheduling based on angular velocity and integral windup prevention algorithm are also implemented. Gain scheduling is responsible for monitoring the current value of velocity of the vehicle, and then updating the control gains with a predetermined set of tuning parameters designed to improve the closed-loop performance at that particular velocity. Integral windup occurs when there is a large change in trajectory and the integral terms accumulates a significant error during the rise (windup), thus overshooting and continuing to increase as this accumulated error is unwound. This issue is solved by stopping the integration of the tracking error when this error rise is above a given threshold.

1.4 Goals

The PI controller is very simple to implement, however there are certain drawbacks. The proportional gains k_{p} and k_{i} need to be tuned extensively which can be time consuming process. More importantly, different virtual controller tuning gains need to be chosen based on the operating velocity of the vehicle. In other words, a simple PI controller does not allow the vehicle to track the required trajectory if the velocity changes drastically within a particular run-time. In order to obtain a more robust performance for this particular vehicle and to make the controller cross platforms compatible, we propose a method to ensure that

the controller takes into account both the kinematics and dynamics of the vehicle and is functional with the same set of parameters for all ranges of the operating speed $[2-9]\frac{m}{s}$. This method incorporates the principles of backstepping and adaptive control ([6, 7]).

Goals of this thesis are to

- Develop a baseline backstepping controller assuming all parameters are known;
- Incorporate on adaptive controls approach to the baseline backstepping controller to handle the uncertainty in parameters;
- Simulate the controller to investigate its performance;
- Implement the controller on a vehicle to test and validate the simulation results.

1.5 Organization of Thesis

This thesis presents an implementation of an indirect Model Reference Adaptive Control (MRAC), a direct MRAC, and an $\mathcal{L}_1$ adaptive backstepping controller for a nonholonomic vehicle. Chapter 2 introduces the proposed baseline backstepping controller which is used in later chapters. Chapter 3 presents the indirect MRAC with backstepping, and then discusses the limitation of this controller. Chapter 4 presents the direct MRAC with backstepping, followed by simulation and experimental results. Chapter 5 presents the $\mathcal{L}_1$ adaptive backstepping. Finally, Chapter 6 summarizes the key results and presents opportunities for future work.

Chapter 2

Backstepping Control

The three controllers presented in Chapter 3, 4 and 5 are built on the foundations of backstepping control. To avoid repetition in the following chapters, Chapter 2 is focused on elaborating the backstepping fundamentals and the theoretical derivation of the backstepping controller for our particular vehicle.

2.1 Theoretical Framework

Backstepping control is a recursive procedure which breaks a control design problem for the full system into a sequence of control design problems for lower order systems. Backstepping control can be applied to nonlinear systems which have a strict feedback form ([6]). The actual control input cascades down through a series of integrators to a fundamental subsystem. First the stabilizing controller law (virtual controller) for the fundamental system is designed. After that, we backstep through the integrators to derive the true control law. A more detailed explanation of the theory and derivation of the backstepping control procedure can be found in [6].

2.2 System Definition and Representation

The unicycle model considered for the control design is given by

$$\dot{p}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} v(t)\cos(\theta(t)) \\ v(t)\sin(\theta(t)) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & 0 \\ \sin(\theta(t)) & 0 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}, \tag{2.1}$$

$$\dot{\theta}(t) = \omega(t), \tag{2.2}$$

where $p(t) \triangleq [x(t)\ y(t)]^{\mathrm{T}} \in \mathbb{R}^2$ is the vehicle's position vector, $v(t) \in \mathbb{R}$ is the vehicle's longitudinal velocity, $\omega(t) \in \mathbb{R}$ is the vehicle's angular velocity, and $\theta(t) \in \mathbb{R}$ is the heading.

The dynamic model considered for the controller design was obtained in Chapter 1 and is given by

$$\dot{s}(t) = As(t) + B\tau(t), \tag{2.3}$$

where $s(t) \triangleq [v(t)\ \omega(t)]^{\mathrm{T}} \in \mathbb{R}^2$ is the vehicle's velocity vector, $A \in \mathbb{R}^{2\times 2}$ is the state matrix, $B \in \mathbb{R}^{2\times 2}$ is the input matrix, and $\tau(t)$ is the signals commanded to the vehicle's motors.

A reference system is introduced whose trajectory converges to the desired trajectory $(r_x(t), r_y(t))$ for the vehicle. It defines the ideal behavior that the adaptation mechanism should follow adjusting the parameters. The particular reference system that we selected is given by

$$\dot{x}_{\mathrm{ref}}(t) = -10x_{\mathrm{ref}}(t) + 10r_x(t), \tag{2.4}$$

$$\dot{y}_{\mathrm{ref}}(t) = -10y_{\mathrm{ref}}(t) + 10r_y(t), \tag{2.5}$$

$$p_{\mathrm{r}}(t) = \begin{bmatrix} x_{\mathrm{ref}}(t) \\ y_{\mathrm{ref}}(t) \end{bmatrix}, \tag{2.6}$$

where $p_{\mathrm{r}}(t) \in \mathbb{R}^2$ in the reference system's position. The controller design guarantees that the system in (2.1) and (2.2) will converge within an adequately small neighborhood of the reference system such that $||p(t) - p_{\mathrm{r}}(t)|| \longrightarrow d^*(t)$, as $t \to \infty$, where $d^*(t)$ is a user defined nominal following distance. This convergence allows the vehicle to follow the reference system in a manner that enforces the predator-prey model interaction. In other words, it ensures that the vehicle follows the reference system and there is a sufficiently large gap between the two such that the vehicle can track the reference without violating its nonholonomic constraints.

2.3 Controller Design

The backstepping controller is developed in two steps. First, a virtual controller is chosen which stabilizes the origin of the tracking error dynamics. Secondly, a torque controller is derived to control the $v(t)$ and $\omega(t)$ to converge to the desired $v_{\mathrm{d}}(t)$ and $\omega_{\mathrm{d}}(t)$. Based on the kinematic and dynamic model given by (2.1)-(2.3), we will proceed to designing our backstepping controller

Virtual Controller. The first step is to design a virtual controller, which would be able to control system's coordinates $p(t)$ and cause it to converge to a small neighborhood of the reference system's coordinates $p_{\mathrm{r}}(t)$, therefore allowing the vehicle to track the reference system. We start by defining the error between the vehicle and the reference system

$$e(t) \triangleq R^{\mathrm{T}}(\theta(t))(p_{\mathrm{r}}(t) - p(t)), \tag{2.7}$$

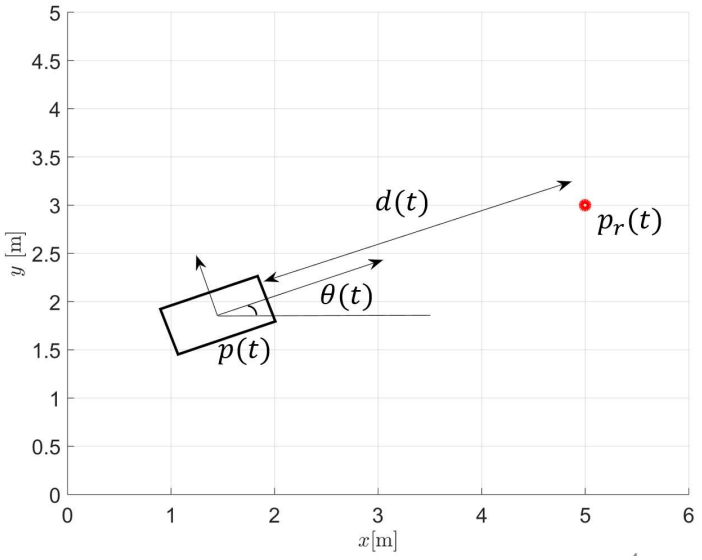

Figure 2.1: The vehicle tracking the reference system with a nominal following distance

where $R(\theta(t)) \triangleq \begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) \\ \sin(\theta(t)) & \cos(\theta(t)) \end{bmatrix}$ is the orthonormal transformation from the body fixed coordinate frame to the global inertial coordinate frame. The distance vector is given by

$$\delta(t) \triangleq \begin{bmatrix} d(t) \\ 0 \end{bmatrix}, \tag{2.8}$$

where $d(t)$ is the following distance and $\delta(t) \in \mathbb{R}^2$. By guaranteeing that the tracking error $e(t)$ converges to the distance vector $\delta(t)$, the longitudinal error will converge to $d(t)$ while the lateral error converges to zero. The following distance $d(t)$, is designed such that $d(t) \to d^*(t)$ as $t \to \infty$.

By taking the time derivative of $e(t)$ along the trajectories of (2.1), (2.2), and (2.6) we obtain

$$\dot{e}(t) = \dot{R}^{\mathrm{T}}(\theta(t))(p_{\mathrm{r}}(t) - p(t)) + R^{\mathrm{T}}(\theta(t))(\dot{p}_{\mathrm{r}}(t) - \dot{p}(t)). \tag{2.9}$$

Defining $S(\omega(t)) \triangleq \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix}$, results in $\dot{R}(\theta(t)) = R(\theta(t))S(\omega(t))$, and

$$\dot{e}(t) = S^{\mathrm{T}}(\omega(t))R^{\mathrm{T}}(\theta(t))(p_{\mathrm{r}}(t) - p(t)) + R^{\mathrm{T}}(\theta(t))(\dot{p}_{\mathrm{r}}(t) - \dot{p}(t)). \tag{2.10}$$

Using the definition of $R(\theta(t))$ and $\dot{p}(t)$,

$$\begin{aligned} \dot{e}(t) \;=\; & S^{\mathrm{T}}(\omega(t))R^{\mathrm{T}}(\theta(t))(p_{\mathrm{r}}(t) - p(t)) + R^{\mathrm{T}}(\theta(t))\dot{p}_{\mathrm{r}}(t) \\ & - \begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) \\ \sin(\theta(t)) & \cos(\theta(t)) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} v(t)\cos(\theta(t)) \\ v(t)\sin(\theta(t)) \end{bmatrix}. \end{aligned} \tag{2.11}$$

Simplifying it further yields,

$$\dot{e}(t) = S^{\mathrm{T}}(\omega(t))e(t) + R^{\mathrm{T}}(\theta(t))\dot{p}_{\mathrm{r}}(t) - \begin{bmatrix} v(t) \\ 0 \end{bmatrix}. \tag{2.12}$$

Using $S^{\mathrm{T}}(\omega(t)) = -S(\omega(t))$

$$\begin{aligned} \dot{e}(t) &= -S(\omega(t))e(t) + R^{\mathrm{T}}(\theta(t))\dot{p}_{\mathrm{r}}(t) - \begin{bmatrix} 1 & 0 \\ 0 & d(t) \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix} \begin{bmatrix} d(t) \\ 0 \end{bmatrix}. \end{aligned} \tag{2.13}$$

Finally, introducing $\Delta(t) \triangleq \begin{bmatrix} 1 & 0 \\ 0 & d(t) \end{bmatrix}$ and using the definition of $\delta(t)$,

$$\dot{e}(t) = -S(\omega(t))e(t) + R^{\mathrm{T}}(\theta(t))\dot{p}_{\mathrm{r}}(t) - \Delta(t)s(t) + S(\omega(t))\delta(t). \tag{2.14}$$

The virtual controller used for this backstepping controller is given by ([4])

$$\alpha(t) \triangleq \begin{bmatrix} v_{\mathrm{d}}(t) \\ \omega_{\mathrm{d}}(t) \end{bmatrix} = \Delta^{-1}(t)(K\tanh(e(t) - \delta(t)) + R^{\mathrm{T}}(\theta(t))\dot{p}_{\mathrm{r}}(t) - \dot{\delta}(t)), \tag{2.15}$$

where $K \triangleq \begin{bmatrix} k_v & 0 \\ 0 & k_w \end{bmatrix}$, and k_v, k_w are tuning constants.

In order to ensure that $\Delta^{-1}(t)$ is defined for all $t \geq 0$, we must guarantee that $d(t) > 0$ for all $t \geq 0$ which is taken care by Theorem 1 in [4]. In particular, the dynamics of the $d(t)$ are chosen as follows

$$\dot{d}(t) = \begin{cases} \Gamma(t), & d(t) \geq \beta, \\ \Gamma(t) + \frac{\beta - d(t)}{d(t) - (\beta - \epsilon),} & d(t) < \beta, \end{cases}, \quad d(0) \geq \beta, \tag{2.16}$$

where $\Gamma(t) \triangleq \dot{d}^*(t) - \lambda(d(t) - d^*(t))$, λ, β, and ϵ as tuning parameters.

Error Dynamics. Based on the virtual controller, we will proceed to define new variables and hence the error dynamics

$$e_1(t) \triangleq e(t) - \delta(t), \quad \dot{e}_1(t) = \dot{e}(t) - \dot{\delta}(t), \tag{2.17}$$

$$e_2(t) \triangleq s(t) - \alpha(t), \quad \dot{e}_2(t) = \dot{s}(t) - \dot{\alpha}(t) = A\dot{s}(t) + B\tau(t) - \dot{\alpha}(t) \tag{2.18}$$

Substituting (2.14) into (2.17), yields the following expression for $\dot{e}_1(t)$,

$$\begin{aligned} \dot{e}_1(t) &= -Se_1(t) + R^{\mathrm{T}}(\theta(t))\dot{p}_{\mathrm{r}}(t) - \Delta(t)s(t) - \dot{\delta}(t) \\ &= -Se_1(t) + R^{\mathrm{T}}(\theta(t))\dot{p}_{\mathrm{r}}(t) - \Delta(t)e_2(t) - \Delta(t)\alpha(t) - \dot{\delta}(t). \end{aligned} \tag{2.19}$$

Substituting the virtual controller $\alpha(t)$ into $\dot{e}_1(t)$ yields,

$$\dot{e}_1(t) = -S(\omega(t))e_1(t) - K\tanh(e_1(t)) - \Delta(t)e_2(t). \tag{2.20}$$

It is shown in [4], that this error dynamics is asymptotically stable, hence $e_1(t) \to 0$ as $t \to 0$.

Torque Controller. In order to enable control of $v(t)$ and $\omega(t)$ to converge to the desired $v_{\mathrm{d}}(t)$ and $\omega_{\mathrm{d}}(t)$ based on the virtual controller, we will need to design a torque controller. The controller is derived using the following Lyapunov function candidate

$$V(t) = \frac{1}{2}e_1^{\mathrm{T}}(t)e_1(t) + \frac{1}{2}(d(t) - d^*(t))^2 + \frac{1}{2}e_2^{\mathrm{T}}(t)e_2(t). \tag{2.21}$$

Differentiating $V(t)$ along the trajectories of (2.20), (2.18), and (2.16), we obtain

$$\dot{V}(t) = e_1^{\mathrm{T}}(t)\dot{e}_1(t) - \lambda(d(t) - d^*(t))^2 + e_2^{\mathrm{T}}(t)\dot{e}_2(t). \tag{2.22}$$

Substituting in the expressions for $\dot{e}_1(t)$ and $\dot{e}_2(t)$ yields,

$$\begin{aligned}\dot{V}(t) &= e_1^{\mathrm{T}}(t)(-S(\omega(t))e_1(t) - K\tanh(e_1(t)) - \Delta(t)e_2(t)) - \lambda(d(t) - d^*(t))^2 \\ &\quad + e_2^{\mathrm{T}}(t)(As(t) + B\tau(t) - \dot{\alpha}(t)) \\ &= -e_1^{\mathrm{T}}(t)S(\omega(t))e_1(t) - e_1^{\mathrm{T}}(t)K\tanh(e_1(t)) - \lambda(d(t) - d^*(t))^2 \\ &\quad - e_1^{\mathrm{T}}(t)\Delta(t)e_2(t) + e_2^{\mathrm{T}}(t)(As(t) + B\tau(t) - \dot{\alpha}(t)).\end{aligned} \tag{2.23}$$

Finally, choosing the torque controller

$$\tau(t) = B^{-1}[-As(t) + \dot{\alpha}(t) - Qe_2(t) + \Delta(t)e_1(t)], \tag{2.24}$$

where $Q > 0$ is a positive definite matrix, we obtain

$$\dot{V}(t) = -e_1^{\mathrm{T}}(t)S(\omega(t))e_1(t) - e_1^{\mathrm{T}}(t)K\tanh(e_1(t)) - \lambda(d(t) - d^*(t))^2 - e_2^{\mathrm{T}}(t)Qe_2(t) < 0 \tag{2.25}$$

which, according to Lyapunov's Stability Theorem in [6], implies that the equilibrium point $(e_1 = 0, d = d^*, e_2 = 0)$ is globally asymptotically stable, as we have identified a positive definite and radially unbounded Lyapunov function $V(t)$ with a negative definite derivative $\dot{V}(t)$ along the trajectories of the error dynamics.

Closed-loop Error Dynamics. Based on the virtual controller and newly derived torque controller, we now derive the closed-loop error dynamics. In particular, the dynamics of $\dot{e}_1(t)$ are unchanged,

$$\dot{e}_1(t) = -S(\omega(t))e_1(t) - K\tanh(e_1(t)) - \Delta(t)e_2(t), \tag{2.26}$$

while the dynamics for $\dot{e}_2(t)$ can be rewritten as

$$\begin{aligned}\dot{e}_2(t) &= \dot{s}(t) - \dot{\alpha}(t) \\ &= As(t) + B\tau(t) - \dot{\alpha}(t) \\ &= As(t) + B\left(B^{-1}[-As(t) + \dot{\alpha}(t) - Qe_2(t) + \Delta(t)e_1(t)]\right) - \dot{\alpha}(t) \\ &= -Qe_2(t) + \Delta(t)e_1(t).\end{aligned} \tag{2.27}$$

2.4 Summary for Controller

Table 2.1: Summary of the Backstepping Controller

System	$\dot{p}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(\mathrm{t})) & 0 \\ \sin(\theta(\mathrm{t})) & 0 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}$ $\dot{\theta}(t) = \omega(t)$ $\dot{s}(t) = As(t) + B\tau(t)$
Virtual Controller	$\alpha(t) = \begin{bmatrix} v_{\mathrm{d}}(t) \\ \omega_{\mathrm{d}}(t) \end{bmatrix} = \Delta^{-1}(t)(K\tanh(e(t) - \delta(t)) + R^{\mathrm{T}}(\theta(t))\dot{p}_{\mathrm{r}}(t) - \dot{\delta}(t))$
Torque Controller	$\tau(t) = B^{-1}[-As(t) + \dot{\alpha}(t) - Qe_2(t) + \Delta(t)e_1(t)]$
Error Dynamics	$\dot{e}(t) = -S(\omega(t))(e(t) - \delta(t)) + R^{\mathrm{T}}(\theta(t))\dot{p}_{\mathrm{r}}(t) - \Delta(t)s(t)$ $\dot{e}_1(t) = -S(\omega(t))e_1(t) - K\tanh(e_1(t)) - \Delta(t)e_2(t)$ $\dot{e}_2(t) = -Qe_2(t) + \Delta(t)e_1(t)$

Chapter 3

Indirect MRAC Adaptive Backstepping

This chapter incorporates the backstepping controller presented in Chapter 2 and an indirect MRAC approach to obtain an indirect MRAC backstepping controller for the vehicle. Since the model parameters A and B for the vehicle are unknown, we will use an adaptive control approach to derive estimates of the unknown model parameters $\hat{A}(t)$ and $\hat{B}(t)$ and use this estimated values to compute the torque controller. Before deriving the update laws for $\hat{A}(t)$ and $\hat{B}(t)$, the theoretical framework for indirect adaptive control will be discussed.

3.1 Theoretical Framework

Adaptive control is an approach for designing control systems which have both parametric and dynamic uncertainties. In many control system problems it is computationally expensive and unrealistic for one to have true values for all the system's parameters. Adaptive control allows for the estimation of unknown parameters based on measured systems signals (feedback signals) and then use of the estimated parameters to compute the desired control law for the system. A model reference adaptive control (MRAC) system is composed of four parts: a system containing unknown parameters, a reference model to specify the desired output of the control system, a feedback control law containing adjustable parameters, and an adaptation mechanism for updating the adjustable parameters. There are two possible approaches to implement an adaptive control algorithm, indirect and direct. Indirect MRAC estimates the plant parameters and uses those for control design, while direct MRAC directly estimates the controller's parameters. In this Chapter, we will be focusing on the indirect MRAC approach.

3.1.1 Single Input Single Output Indirect MRAC Example

A simple indirect MRAC is implemented on a SISO system to act as an example. Consider the following system

$$\dot{x}(t) = ax(t) + bu(t), \quad x(0) = x_0, \tag{3.1}$$

where $x(t) \in \mathbb{R}$ is the state variable, $u(t) \in \mathbb{R}$ is the control signal, a and b are unknown, but the sign of b is known.

The reference system is given by

$$\dot{x}_{\mathrm{m}}(t) = a_{\mathrm{m}}x_{\mathrm{m}}(t) + b_{\mathrm{m}}r(t), \quad x_{\mathrm{m}}(0) = x_{\mathrm{m}0}, \tag{3.2}$$

where $a_{\mathrm{m}} < 0$ and $b_{\mathrm{m}} \in \mathbb{R}$ specify the desired performance and $r(t) \in \mathbb{R}$ is a continuous bounded reference input. A controller is chosen such that $x(t)$ asymptotically tracks $x_{\mathrm{m}}(t)$ while all the signals remain bounded,

$$u(t) = \frac{1}{\hat{b}}(-\hat{a}x(t) + a_{\mathrm{m}}x_{\mathrm{m}}(t) + b_{\mathrm{m}}r(t)), \tag{3.3}$$

where $\hat{a}$ and $\hat{b}$ are the estimated values.

Rewriting the system dynamics yields,

$$\begin{aligned} \dot{x}(t) &= ax(t) + bu(t) + \hat{a}x(t) + \hat{b}u(t) - \hat{a}x(t) - \hat{b}u(t) \\ &= \hat{a}x(t) + \hat{b}u(t) - \tilde{a}x(t) - \tilde{b}u(t), \end{aligned} \tag{3.4}$$

where $\tilde{a} \triangleq \hat{a} - a$ and $\tilde{b} \triangleq \hat{b} - b$.
Substituting (3.3) into (3.4) yields,

$$\begin{aligned} \dot{x}(t) &= \hat{a}x(t) + \hat{b}\left[\frac{1}{\hat{b}}(-\hat{a}x(t) + a_{\mathrm{m}}x_{\mathrm{m}}(t) + b_{\mathrm{m}}r(t))\right] - \tilde{a}x(t) - \tilde{b}u(t) \\ &= a_{\mathrm{m}}x_{\mathrm{m}}(t) + b_{\mathrm{m}}r(t)) - \tilde{a}x(t) - \tilde{b}u(t). \end{aligned} \tag{3.5}$$

The tracking error signal is $e(t) \triangleq x(t) - x_{\mathrm{m}}(t)$ and the error dynamics is derived by computing its derivatives along the trajectories of (3.2) and (3.5)

$$\begin{aligned} \dot{e}(t) &= a_{\mathrm{m}}x_{\mathrm{m}}(t) + b_{\mathrm{m}}r(t)) - \tilde{a}x(t) - \tilde{b}u(t) - a_{\mathrm{m}}x_{\mathrm{m}}(t) - b_{\mathrm{m}}r(t) \\ &= a_{\mathrm{m}}e(t) - \tilde{a}x(t) - \tilde{b}u(t). \end{aligned} \tag{3.6}$$

Next, considering the following Lyapunov function

$$V(t) = e^2(t) + \frac{1}{\gamma_a}\tilde{a}^2 + \frac{1}{\gamma_b}\tilde{b}^2, \tag{3.7}$$

and computing its time derivative along the trajectories of (3.6)

$$\begin{aligned}\dot{V}(t) &= 2e(t)\dot{e}(t) + \frac{2}{\gamma_a}\tilde{a}\dot{\tilde{a}} + \frac{2}{\gamma_b}\tilde{b}\dot{\tilde{b}} \\ &= 2a_{\mathrm{m}}e^2(t) + 2\tilde{a}(\frac{1}{\gamma_a}\dot{\tilde{a}} - e(t)x(t)) + 2\tilde{b}(\frac{1}{\gamma_b}\dot{\tilde{b}} - e(t)u(t)). \end{aligned} \tag{3.8}$$

Choosing the update laws

$$\dot{\tilde{a}} = \gamma_a x(t)e(t), \tag{3.9}$$

$$\dot{\tilde{b}} = \gamma_b u(t)e(t), \tag{3.10}$$

yields,

$$\dot{V}(t) = 2a_{\mathrm{m}}e^2(t) \leq 0, \tag{3.11}$$

which shows that the system given by (3.6), (3.9), and (3.10) is Lyapunov stable and according to LaSalle Yoshizawa theorem ([8]), $e(t) \to 0$ as $t \to \infty$.

3.2 Indirect MRAC for Vehicle

As derived in Chapter 2, we will be using the following torque controller

$$\tau(t) = B^{-1}[-As(t) + \dot{\alpha}(t) - Qe_2 + \Delta(t)e_1(t)], \tag{3.12}$$

where $Q > 0$ is a positive definite matrix.

3.2.1 Indirect Adaptive Update Laws

In order to derive the update laws for $\hat{A}(t)$ and $\hat{B}(t)$, we consider the following Lyapunov function candidate $V_a(t)$ and the new controller $\hat{\tau}(t)$ with estimated values of the unknown parameters A and B as shown below

$$\begin{aligned}V_a(t) &= \frac{1}{2}e_1^{\mathrm{T}}(t)e_1(t) + \frac{1}{2}(d(t) - d^*(t))^2 + \frac{1}{2}e_2^{\mathrm{T}}(t)e_2(t) + \frac{1}{2}\mathrm{tr}(\tilde{A}(t)\Gamma_a^{-1}\tilde{A}^{\mathrm{T}}(t)) \\ &\quad + \frac{1}{2}\mathrm{tr}(\tilde{B}(t)\Gamma_b^{-1}\tilde{B}^{\mathrm{T}}(t)), \end{aligned} \tag{3.13}$$

where $\tilde{A}(t) = \hat{A}(t) - A$, $\tilde{B}(t) = \hat{B}(t) - B$, and $\Gamma_a^{-1} \in \mathbb{R}^{2\times 2}$, $\Gamma_b^{-1} \in \mathbb{R}^{2\times 2}$ are positive definite tuning parameters,

$$\hat{\tau}(t) = \hat{B}(t)^{-1}(-\hat{A}(t)s(t) + \dot{\alpha}(t) - Qe_2(t) + \Delta(t)e_1(t)). \tag{3.14}$$

By differentiating $V_a(t)$ along the trajectories of $e_1(t)$, $d(t)$, and $e_2(t)$, we obtain

$$\begin{aligned}\dot{V}_a(t) &= e_1^{\mathrm{T}}(t)(-S(\omega(t))e_1(t) - K\tanh(e_1(t)) - \Delta(t)e_2(t)) - \lambda(d(t)-d^*(t))^2 \\ &\quad + e_2^{\mathrm{T}}(t)(As(t) + B\hat{\tau}(t) - \dot{\alpha}(t)) + \mathrm{tr}(\tilde{A}(t)\Gamma_a^{-1}\dot{\tilde{A}}^{\mathrm{T}}(t)) + \mathrm{tr}(\tilde{B}(t)\Gamma_b^{-1}\dot{\tilde{B}}^{\mathrm{T}}(t)) \\ &= e_1^{\mathrm{T}}(t)(-S(\omega(t))e_1(t) - K\tanh(e_1(t)) - \Delta(t)e_2(t)) - \lambda(d(t)-d^*(t))^2 \\ &\quad + e_2^{\mathrm{T}}(t)(As(t) - [\tilde{B}(t) - \hat{B}(t)]\hat{\tau}(t) - \dot{\alpha}(t)) + \mathrm{tr}(\tilde{A}(t)\Gamma_a^{-1}\dot{\tilde{A}}^{\mathrm{T}}(t)) \\ &\quad + \mathrm{tr}(\tilde{B}(t)\Gamma_b^{-1}\dot{\tilde{B}}^{\mathrm{T}}(t)). \end{aligned} \tag{3.15}$$

Substituting in the expression for $\hat{\tau}(t)$ and further simplifying yields

$$\begin{aligned}\dot{V}_a(t) &= -e_1^{\mathrm{T}}(t)S(\omega(t))e_1(t) - e_1^{\mathrm{T}}(t)K\tanh(e_1(t)) - \lambda(d(t)-d^*(t))^2 - e_1^{\mathrm{T}}(t)\Delta(t)e_2(t) \\ &\quad + e_2^{\mathrm{T}}(t)(-\tilde{A}(t)s(t) - \tilde{B}(t)\hat{\tau}(t) - Qe_2(t) + \Delta(t)e_1(t)) + \mathrm{tr}(\tilde{A}(t)\Gamma_a^{-1}\dot{\tilde{A}}^{\mathrm{T}}(t)) \\ &\quad + \mathrm{tr}(\tilde{B}(t)\Gamma_b^{-1}\dot{\tilde{B}}^{\mathrm{T}}(t)) \\ &= -e_1^{\mathrm{T}}(t)S(\omega(t))e_1(t) - e_1^{\mathrm{T}}(t)K\tanh(e_1(t)) - \lambda(d(t)-d^*(t))^2 - e_2^{\mathrm{T}}(t)\tilde{A}(t)s(t) \\ &\quad - e_2^{\mathrm{T}}(t)\tilde{B}(t)\hat{\tau}(t) - e_2^{\mathrm{T}}(t)Qe_2(t) + \mathrm{tr}(\tilde{A}(t)\Gamma_a^{-1}\dot{\tilde{A}}^{\mathrm{T}}(t)) + \mathrm{tr}(\tilde{B}(t)\Gamma_b^{-1}\dot{\tilde{B}}^{\mathrm{T}}(t)). \end{aligned} \tag{3.16}$$

To derive the update law for $\tilde{A}(t)$ and $\tilde{B}(t)$, we must ensure that

$$\mathrm{tr}(-\tilde{A}(t)s(t)e_2^{\mathrm{T}}(t) + \tilde{A}(t)\Gamma_a^{-1}\dot{\tilde{A}}^{\mathrm{T}}(t)) = 0, \tag{3.17}$$

$$\mathrm{tr}(-\tilde{B}(t)\tau(t)e_2^{\mathrm{T}}(t) + \tilde{B}(t)\Gamma_b^{-1}\dot{\tilde{B}}^{\mathrm{T}}(t)) = 0, \tag{3.18}$$

which is achieved by choosing,

$$\dot{A}(t) = e_2(t)s^{\mathrm{T}}(t)\Gamma_a, \tag{3.19}$$

$$\dot{B}(t) = e_2(t)\tau^{\mathrm{T}}(t)\Gamma_b. \tag{3.20}$$

Substituting the adaptive laws (3.19), (3.20) into (3.16), we obtain

$$\begin{aligned}\dot{V}_a(t) &= -e_1^{\mathrm{T}}(t)S(\omega(t))e_1(t) - e_1^{\mathrm{T}}(t)K\tanh(e_1(t)) - \lambda(d(t)-d^*(t))^2 \\ &\quad - e_2^{\mathrm{T}}(t)Qe_2(t) \leq 0 \end{aligned} \tag{3.21}$$

which shows that the system given by (3.19), (3.20), (2.26) and (2.27) is Lyapunov stable and according to LaSalle Yoshizawa theorem ([8]), $e_1(t) \to 0$, and $e_2(t) \to 0$ as $t \to \infty$.

Closed-loop Error Dynamics with Adaptive Laws. Based on the virtual controller, newly derived torque controller and adaptive laws, we now derive the closed-loop error dynamics. In particular, the dynamics of $\dot{e}_1(t)$ are unchanged

$$\dot{e}_1(t) = -S(\omega(t))e_1(t) - K\tanh(e_1(t)) - \Delta(t)e_2(t), \tag{3.22}$$

while the dynamics of $\dot{e}_2(t)$ can be rewritten as

$$\begin{aligned}\dot{e}_2(t) &= \dot{s}(t) - \dot{\alpha}(t) \\ &= As(t) + B\hat{\tau}(t) - \dot{\alpha}(t) \\ &= As(t) - \tilde{B}(t)\hat{\tau}(t) + \hat{B}(t)\left(\hat{B}(t)^{-1}[-\hat{A}(t)s(t) + \dot{\alpha}(t) - Qe_2(t) + \Delta(t)e_1(t)]\right) - \dot{\alpha}(t) \\ &= -Qe_2(t) + \Delta(t)e_1(t) - \tilde{A}(t)s(t) - \tilde{B}(t)\hat{\tau}(t). \end{aligned} \tag{3.23}$$

3.2.2 Limitations with Current Controller

To compute the backstepping torque controller utilizing the estimated parameters $\hat{A}(t)$ and $\hat{B}(t)$, there are certain restrictions that need to be placed on $\hat{B}(t)$. $\hat{B}(t)$ has be to an invertible matrix at all times. Therefore $\hat{B}(t)$ has to be a nonsingular matrix, and similar to the structure seen below

$$\hat{B}(t) = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}, \tag{3.24}$$

where $det(\hat{B}(t)) \neq 0$. Projection algorithm was implemented to enforce $\hat{B}(t)$ to be invertible. However, in the simulation phase, the non-global results cause difficulties in obtaining a solution for the indirect MRAC backstepping controller when the B values were chosen are too close to a nonsingular matrix. We solved this problem by switching to a Direct MRAC scheme which provides a global result and the development and implementation of that will be discussed in Chapter 4.

Chapter 4

Direct MRAC Adaptive Backstepping

In this chapter, we are going to introduce a slightly different control structure which solves the difficulties associated with the previous one (indirect MRAC backstepping). This chapter incorporates a direct MRAC approach to the backstepping controller presented in Chapter 2 to obtain direct MRAC backstepping controller for the vehicle. Since the model parameters A and B are unknown, we will use an adaptive control approach to derive estimates of the unknown controller parameters $\hat{\theta}_s(t)$ and $\hat{\theta}_{\mathrm{r}}(t)$ and use them for the calculation of the torque controller.

4.1 Theoretical Framework

4.1.1 Single Input Single Output Direct MRAC Example

A simple direct MRAC is implemented on a SISO system to be shown as an example. Consider the following system

$$\dot{x}(t) = ax(t) + bu(t), \quad x(0) = x_0, \tag{4.1}$$

where $x(t) \in \mathbb{R}$ is the state variable, $u(t) \in \mathbb{R}$ is the control signal, a and b are unknown, but the sign of b is known.

The reference system is given by

$$\dot{x}_{\mathrm{m}}(t) = a_{\mathrm{m}}x_{\mathrm{m}}(t) + b_{\mathrm{m}}r(t), \quad x_{\mathrm{m}}(0) = x_{\mathrm{m}0}, \tag{4.2}$$

where $a_{\mathrm{m}} < 0$ and $b_{\mathrm{m}} \in \mathbb{R}$ specify the desired performance and $r(t) \in \mathbb{R}$ is a continuous bounded reference input.

A controller is chosen such that $x(t)$ asymptotically tracks $x_{\rm m}(t)$ while all the signals remain bounded,

$$u(t) = k_x^* x(t) + k_r^* r(t), \tag{4.3}$$

where k_x^* and k_r^* are the real adaptive gains.

Substituting (4.3) into (4.1), yields the following system dynamics

$$\dot{x}(t) = (a + bk_x^*)x(t) + bk_r^* r(t). \tag{4.4}$$

By comparing (4.2) with the system dynamics in (4.4), we derive the compatibility equations

$$a_{\rm m} = a + bk_x^*, \tag{4.5}$$
$$b_{\rm m} = bk_r^*. \tag{4.6}$$

The tracking error signal is $e(t) \triangleq x(t) - x_{\rm m}(t)$ and the error dynamics is derived by computing the derivative along the trajectories of (4.4) and (4.2)

$$\dot{e}(t) = a_{\rm m} e(t) + b\tilde{k}_x(t)x(t) + b\tilde{k}_r(t)r(t), \tag{4.7}$$

where $\hat{k}_s(t)$ and $\hat{k}_r(t)$ are the estimated adaptive gains, while $\tilde{k}_x(t) = \hat{k}_x(t) - k_x^*$ and $\tilde{k}_r(t) = \hat{k}_r(t) - k_r^*$.

Next, consider the following Lyapunov function

$$V(t) = \frac{e(t)^2}{2} + \frac{1}{2}|b|(\tilde{k}_x^2(t) + \tilde{k}_r^2(t)), \tag{4.8}$$

and compute its time derivative along the trajectory of (4.7)

$$\begin{aligned} \dot{V}(t) &= e(t)\dot{e}(t) + |b|\tilde{k}_x(t)\dot{\tilde{k}}_x(t) + |b|\tilde{k}_r(t)\dot{\tilde{k}}_r(t) \\ &= e(t)(a_{\rm m}e(t) + b\tilde{k}_x(t)x(t) + b\tilde{k}_r(t)r(t)) + |b|\tilde{k}_x(t)\dot{\tilde{k}}_x(t) + |b|\tilde{k}_r(t)\dot{\tilde{k}}_r(t). \end{aligned} \tag{4.9}$$

Choosing the adaptive update laws

$$\dot{\tilde{k}}_x(t) = -\frac{b}{|b|}x(t)e(t), \tag{4.10}$$
$$\dot{\tilde{k}}_r(t) = -\frac{b}{|b|}r(t)e(t), \tag{4.11}$$

yields,

$$\dot{V}(t) = 2a_{\rm m}e(t)^2 \leq 0, \tag{4.12}$$

which shows that the system given by (4.7), (4.10), and (4.11) is Lyapunov stable and according to LaSalle Yoshizawa theorem ([8]), $e(t) \to 0$ as $t \to \infty$.

4.2 Direct MRAC for Vehicle

As derived from Chapter 2, we will be using the following torque controller.

$$\tau(t) = B^{-1}[-As(t) + \dot{\alpha}(t) - Qe_2(t) + \Delta(t)e_1(t)], \tag{4.13}$$

where $Q > 0$ is a positive definite matrix.

Direct Torque Controller. In order to pursue a direct MRAC adaptive backstepping controller, we need to represent the torque controller in terms of controller parameters instead of the unknown parameters of the system namely A and B. Therefore the previously defined torque controller is rewritten in terms of the adaptive controller parameters as

$$\begin{aligned} \eta(t) &= \dot{\alpha}(t) - Qe_2(t) + \Delta(t)e_1(t), & (4.14)\\ \tau(t) &= \theta_s^* s(t) + \theta_{\mathrm{r}}^* \eta(t), & (4.15) \end{aligned}$$

where $\theta_s^* = -B^{-1}A$ and $\theta_{\mathrm{r}}^* = B^{-1}$ are the true adaptive gain values.

To determine the compatibility equations, we will derive the new closed-loop error dynamics based on the new direct torque controller. The dynamics of $\dot{e}_1(t)$ are unchanged

$$\dot{e}_1(t) = -S(\omega(t))e_1(t) - K\tanh(e_1(t)) - \Delta(t)e_2(t), \tag{4.16}$$

while the dynamics of $\dot{e}_2(t)$ can be rewritten as

$$\begin{aligned} \dot{e}_2(t) &= \dot{s}(t) - \dot{\alpha}(t)\\ &= As(t) + B\tau(t) - \dot{\alpha}(t),\\ &= As(t) + B[B^{-1}[\theta_s^* s(t) + \theta_{\mathrm{r}}^* \eta(t)] - \dot{\alpha}(t). \end{aligned} \tag{4.17}$$

Now, if θ_s^* and θ_{r}^* satisfy the following compatibility equations

$$\begin{aligned} A + B\theta_s^* &= 0, & (4.18)\\ B\theta_{\mathrm{r}}^* &= I_2, & (4.19) \end{aligned}$$

$e_2(t)$ yields to be

$$\dot{e}_2(t) = -Qe_2(t) + \Delta(t)e_1(t). \tag{4.20}$$

4.2.1 Direct Adaptive Update Laws

In order to derive the update laws, we consider the following Lyapunov function candidate V_{a} given by

$$\begin{aligned} V_{\mathrm{a}}(t) &= \frac{1}{2}e_1^{\mathrm{T}}(t)e_1(t) + \frac{1}{2}(d(t) - d^*(t))^2 + \frac{1}{2}e_2^{\mathrm{T}}(t)e_2(t) + \frac{1}{2}\mathrm{tr}(B\tilde{\theta}_s(t)\Gamma_s^{-1}\tilde{\theta}_s^{\mathrm{T}}(t)) \quad (4.21)\\ &\quad + \frac{1}{2}\mathrm{tr}(B\tilde{\theta}_{\mathrm{r}}(t)\Gamma_{\mathrm{r}}^{-1}\tilde{\theta}_{\mathrm{r}}^{\mathrm{T}}(t)), \end{aligned}$$

where $\tilde{\theta}_s(t) = \hat{\theta}_s(t) - \theta_s^*$, $\tilde{\theta}_{\rm r}(t) = \hat{\theta}_{\rm r}(t) - \theta_{\rm r}^*$, and $\Gamma_s^{-1} \in \mathbb{R}^{2\times 2}$, $\Gamma_{\rm r}^{-1} \in \mathbb{R}^{2\times 2}$ are positive definite tuning parameters and the new controller $\hat{\tau}$ with estimated values of the unknown adaptive gains ($\hat{\theta}_s(t)$ and $\hat{\theta}_{\rm r}(t)$) given by

$$\hat{\tau}(t) = \hat{\theta}_s(t)s(t) + \hat{\theta}_{\rm r}(t)\eta(t). \tag{4.22}$$

Differentiating $V_{\rm a}(t)$ along the trajectories of $e_1(t)$, $d(t)$, and $e_2(t)$, we obtain

$$\begin{aligned} \dot{V}_{\rm a}(t) &= e_1^{\rm T}(t)\dot{e}_1(t) - \lambda(d(t) - d^*(t))^2 + e_2^{\rm T}(t)\dot{e}_2(t) \\ &\quad + {\rm tr}(B\tilde{\theta}_s(t)\Gamma_s^{-1}\dot{\tilde{\theta}}_s^{\rm T}(t)) + {\rm tr}(B\tilde{\theta}_{\rm r}(t)\Gamma_{\rm r}^{-1}\dot{\tilde{\theta}}_{\rm r}^{\rm T}(t)). \end{aligned} \tag{4.23}$$

Substituting in the expression for $\dot{e}_1(t)$ and $\dot{e}_2(t)$ yields,

$$\begin{aligned} \dot{V}_{\rm a}(t) &= e_1^{\rm T}(t)(-S(\omega(t))e_1(t) - K\tanh(e_1(t)) - \Delta(t)e_2(t)) - \lambda(d(t) - d^*(t))^2 \\ &\quad + e_2^{\rm T}(t)(As(t) + B\hat{\tau}(t) - \dot{\alpha}(t)) + {\rm tr}(B\tilde{\theta}_s(t)\Gamma_s^{-1}\dot{\tilde{\theta}}_s^{\rm T}(t)) + {\rm tr}(B\tilde{\theta}_{\rm r}(t)\Gamma_{\rm r}^{-1}\dot{\tilde{\theta}}_{\rm r}^{\rm T}(t)). \end{aligned} \tag{4.24}$$

Substituting (4.22) into (4.24) and using the first compatibility equation, we obtain

$$\begin{aligned} \dot{V}_{\rm a}(t) &= -e_1^{\rm T}(t)S(\omega(t))e_1(t) - e_1^{\rm T}(t)K\tanh(e_1(t)) - \lambda(d(t) - d^*(t))^2 \\ &\quad - e_1^{\rm T}(t)\Delta(t)e_2(t) + e_2^{\rm T}(t)(-B\theta_s^*s(t) + B[\hat{\theta}_s(t)s(t) + \hat{\theta}_{\rm r}(t)\eta(t)] - \dot{\alpha}(t)) \\ &\quad + {\rm tr}(B\tilde{\theta}_s(t)\Gamma_s^{-1}\dot{\tilde{\theta}}_s^{\rm T}(t)) + {\rm tr}(B\tilde{\theta}_{\rm r}(t)\Gamma_{\rm r}^{-1}\dot{\tilde{\theta}}_{\rm r}^{\rm T}(t)), \end{aligned} \tag{4.25}$$

and further simplifying yields,

$$\begin{aligned} \dot{V}_{\rm a}(t) &= -e_1^{\rm T}(t)S(\omega(t))e_1(t) - e_1^{\rm T}(t)K\tanh(e_1(t)) - \lambda(d(t) - d^*(t))^2 \\ &\quad - e_1^{\rm T}(t)\Delta(t)e_2(t) + e_2^{\rm T}(t)(-B\theta_s^*s(t) + B\hat{\theta}_s(t)s(t) + B[\tilde{\theta}_{\rm r}(t)\eta(t) + \theta_{\rm r}^*(t)\eta(t)] - \dot{\alpha}(t)) \\ &\quad + {\rm tr}(B\tilde{\theta}_s(t)\Gamma_s^{-1}\dot{\tilde{\theta}}_s^{\rm T}(t)) + {\rm tr}(B\tilde{\theta}_{\rm r}(t)\Gamma_{\rm r}^{-1}\dot{\tilde{\theta}}_{\rm r}^{\rm T}(t)) \\ &= -e_1^{\rm T}(t)S(\omega(t))e_1(t) - e_1^{\rm T}(t)K\tanh(e_1(t)) - \lambda(d(t) - d^*(t))^2 \\ &\quad - e_1^{\rm T}(t)\Delta(t)e_2(t) + e_2^{\rm T}(t)(B\tilde{\theta}_s(t)s(t) + B[\tilde{\theta}_{\rm r}(t)\eta(t) + \theta_{\rm r}^*(t)\eta(t)] - \dot{\alpha}(t)) \\ &\quad + {\rm tr}(B\tilde{\theta}_s(t)\Gamma_s^{-1}\dot{\tilde{\theta}}_s^{\rm T}(t)) + {\rm tr}(B\tilde{\theta}_{\rm r}(t)\Gamma_{\rm r}^{-1}\dot{\tilde{\theta}}_{\rm r}^{\rm T}(t)). \end{aligned} \tag{4.26}$$

Lastly, using the second compatibility equation yields,

$$\begin{aligned} \dot{V}_{\rm a}(t) &= -e_1^{\rm T}(t)S(\omega(t))e_1(t) - e_1^{\rm T}K\tanh(e_1(t)) - \lambda(d(t) - d^*(t))^2 \\ &\quad - e_1^{\rm T}(t)\Delta(t)e_2(t) + e_2^{\rm T}(B\tilde{\theta}_s(t)s(t) + B\tilde{\theta}_{\rm r}(t)\eta(t) - Qe_2(t) + \Delta(t)e_1(t)) \\ &\quad + {\rm tr}(B\tilde{\theta}_s(t)\Gamma_s^{-1}\dot{\tilde{\theta}}_s^{\rm T}(t)) + {\rm tr}(B\tilde{\theta}_{\rm r}(t)\Gamma_{\rm r}^{-1}\dot{\tilde{\theta}}_{\rm r}^{\rm T}(t)). \end{aligned} \tag{4.27}$$

To derive the update law for $\tilde{\theta}_s(t)$ and $\tilde{\theta}_{\rm r}(t)$, we must ensure that

$$\begin{aligned} {\rm tr}(e_2^{\rm T}(t)B\tilde{\theta}_s(t)s(t) + B\tilde{\theta}_s(t)\Gamma_s^{-1}\dot{\tilde{\theta}}_s^{\rm T}(t)) &= 0, & (4.28) \\ {\rm tr}(e_2^{\rm T}(t)B\tilde{\theta}_{\rm r}(t)\eta(t) + B\tilde{\theta}_{\rm r}(t)\Gamma_{\rm r}^{-1}\dot{\tilde{\theta}}_{\rm r}^{\rm T}(t)) &= 0, & (4.29) \end{aligned}$$

which is achieved by choosing

$$\dot{\theta}_s(t) = -e_2(t)s^{\mathrm{T}}(t)\Gamma_s, \tag{4.30}$$
$$\dot{\theta}_{\mathrm{r}}(t) = -e_2(t)\eta^{\mathrm{T}}(t)\Gamma_{\mathrm{r}}. \tag{4.31}$$

Substituting the adaptive laws (4.30), (4.31) into (4.27) yields

$$\dot{V}_{\mathrm{a}}(t) = -e_1^{\mathrm{T}}(t)S(\omega(t))e_1(t) - e_1^{\mathrm{T}}(t)K\tanh(e_1(t)) - \lambda(d(t)-d^*(t))^2 - e_2^{\mathrm{T}}(t)Qe_2(t) \leq 0, \tag{4.32}$$

which shows that the system given by (4.30), (4.31), (2.26) and (2.27) is Lyapunov stable and according to LaSalle Yoshizawa theorem ([8]), $e_1(t) \to 0$, and $e_2(t) \to 0$ as $t \to \infty$.

Closed-loop Error Dynamics with Adaptive Laws. Based on the new adaptive laws, we derive the new closed-loop error dynamics. The dynamics of $\dot{e}_1(t)$ are unchanged

$$\dot{e}_1(t) = -S(\omega(t))e_1(t) - K\tanh(e_1(t)) - \Delta(t)e_2(t), \tag{4.33}$$

while the dynamics of $\dot{e}_2(t)$ can be rewritten as

$$\begin{aligned}
\dot{e}_2(t) &= \dot{s}(t) - \dot{\alpha}(t) \\
&= As(t) + B\hat{\tau}(t) - \dot{\alpha}(t) \\
&= -B\theta_s^* s(t) + B\hat{\theta}_s(t)s(t) + B[\hat{\theta}_{\mathrm{r}}(t)\eta(t)] - \dot{\alpha}(t) \\
&= B\tilde{\theta}_s(t)s(t) + B[\tilde{\theta}_{\mathrm{r}}(t) + \theta_{\mathrm{r}}^*]\eta(t) - \dot{\alpha}(t) \\
&= B\tilde{\theta}_s(t)s(t) + B\tilde{\theta}_{\mathrm{r}}(t)\eta(t) + I_2\eta(t) - \dot{\alpha}(t) \\
&= -Qe_2(t) + \Delta(t)e_1(t) + B\tilde{\theta}_s(t)s(t) + B\tilde{\theta}_{\mathrm{r}}(t)\eta(t).
\end{aligned} \tag{4.34}$$

4.3 Simulation Implementation and Results

In this section, the effectiveness of the direct MRAC backstepping controller is investigated through the use of computer simulation. This section include the description of the simulation, results and analysis. All simulations presented were carried out with either a trajectory dictated by a sine wave or a trajectory dictated by GPS waypoints as a reference system.

4.3.1 Computer Simulation

Computer simulations using the software MATLAB were used to test the controller. The details of the simulation for the direct MRAC backstepping controller will be presented via a block diagram shown in Figure 4.1. Each block will be discussed individually.

(1) **Plant.** This block is the plant, which simulates the vehicle using the unicycle and dynamic model defined in Chapter 1. It takes in torque control commands as inputs and outputs the trajectory and the longitudinal and angular velocities of the vehicle.

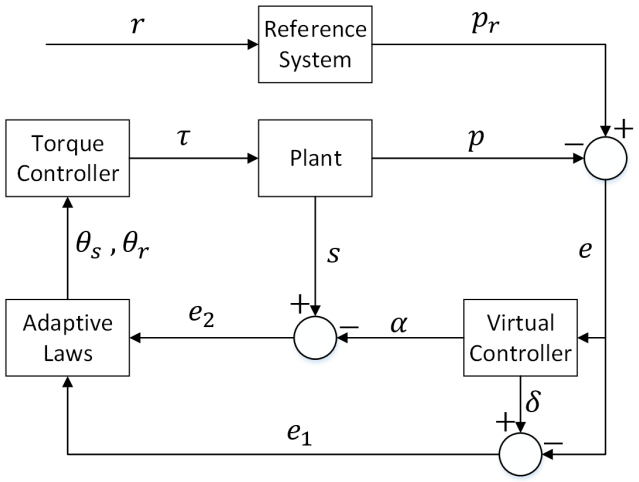

Figure 4.1: MRAC Backstepping Algorithm Block Diagram

(2) **Reference system.** This block is for the reference plant, based on the model we are using. It is either, the sine-wave dictated trajectory or the waypoint dictated trajectory. This block is responsible for simulating a reference trajectory that the vehicle should follow.

(3) **Virtual Controller.** This block is responsible for computing the desired longitudinal and angular velocities based on the error computed between the position of the vehicle and the position of the reference system.

(4) **Adaptive Laws.** This block is responsible for updating the adaptive gains based on the error computed between the position of the vehicle and the position of the reference system, and the error computed between the desired velocities and the actual velocities of the vehicle.

(5) **Torque Controller.** This block is responsible for computing the torque commands, using the adaptive controller parameters, for the motors of the vehicle based on the adaptive gains and the current velocities of the vehicle.

4.3.2 Reference System: Sine Wave Dictated Trajectory

A sine wave dictated reference system is introduced whose trajectory converges to the desired trajectory $r_x(t)$and $r_y(t)$ for the vehicle. The particular reference system that we selected is

given by

$$\begin{aligned} r_x(t) &= 0.5t, \quad \dot{r}_x(t) = 0.5, && (4.35)\\ r_y(t) &= 10\sin(0.5t), \quad \dot{r}_y(t) = 5\cos(0.5t), && (4.36)\\ \dot{x}_{\text{ref}}(t) &= -10x_{\text{ref}}(t) + 10r_x(t), \quad \dot{y}_{\text{ref}}(t) = -10y_{\text{ref}}(t) + 10r_y(t), && (4.37)\\ p_{\text{r}}(t) &= \begin{bmatrix} x_{\text{ref}}(t) \\ y_{\text{ref}}(t) \end{bmatrix}, && (4.38) \end{aligned}$$

where $p_{\text{r}}(t) \in \mathbb{R}^2$ in the reference system's position.

4.3.3 Reference System: Waypoints Dictated Trajectory

A waypoint dictated reference system is introduced to establish the desired target and trajectory for the vehicle. The reference system is implemented similar to the simple potential field algorithm found in [9–11]. This particular method was used for its' intuitive velocity behavior and computationally cheap characteristics. To establish the algorithm, the distance between the reference system and the destination is defined as d_{des}

$$d_{\text{des}}(t) \triangleq \sqrt{(x_{\text{des,i}} - x_{\text{r}}(t))^2 + (y_{\text{des,i}} - y_{\text{r}}(t))^2}, \tag{4.39}$$

where $d_{\text{des}} \in \mathbb{R}$, $x_{\text{des,i}} \in \mathbb{R}$ is the x coordinate of the i^{th} destination waypoint, and $y_{\text{des,i}} \in \mathbb{R}$ is the y coordinate of the current destination waypoint. Based on d_{des}, the virtual force vectors are defined

$$F_a(t) \triangleq \frac{F_{ac}}{d_{\text{des(t)}}} \begin{bmatrix} x_{\text{des,i}} - x_{\text{r}}(t) \\ y_{\text{des,i}} - y_{\text{r}}(t) \end{bmatrix}, \tag{4.40}$$

where F_{ac} is a tuning constant and $F_a(t) \in \mathbb{R}^2$ is the attractive force vector.

The reference system is incorporated into a dynamic system defined as

$$m\ddot{p}_{\text{r}}(t) + c\dot{p}_{\text{r}}(t) = F_a(t), \tag{4.41}$$

where $m > 0$,and $c > 0$ are tuning constants for inertia and damping of the virtual system. The mass and damping is defined as follows $m = \frac{2E}{v_{\text{rmax}}^2}$, and $c = \frac{F_{ac}}{v_{\text{rmax}}}$, where v_{rmax} is the maximum velocity in the specified trajectory, and E is the user defined kinetic energy of the reference system ([4]).

4.3.4 Results

This section expands on the simulation results generated for the sine-wave dictated and then the waypoint dictated trajectory.

Unsaturated Sine Reference. For this category, the sine wave dictated trajectory is shown below in Figure 4.2 and the actual and desired velocities compared are shown in Figure 4.3. The following model parameters were used: $A^* = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$, and $B^* = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$. The following initialization values were used: $x_{\mathrm{r}}(0) = 0$, $y_{\mathrm{r}}(0) = 0$, $x(0) = -0.1$, $y(0) = 0$, $\theta(0) = 0$, $d(0) = 0.1$, $v(0) = 0$, $\omega(0) = 0$, $v_{\mathrm{d}}(0) = 0$, $\omega_{\mathrm{d}}(0) = 0$, $\theta_s(0) = \left[\begin{smallmatrix} -4 & 0 \\ 0 & -4 \end{smallmatrix}\right]$, and $\theta_{\mathrm{r}}(0) = \left[\begin{smallmatrix} 2 & 0 \\ 0 & 2 \end{smallmatrix}\right]$. And finally the following tuning parameters were used: $k_v = 1$, $k_w = 1$, $\alpha = 0.5$, $\beta = 0.1$, $\lambda = 1$, $\Gamma_s = \left[\begin{smallmatrix} 0.01 & 0 \\ 0 & 0.01 \end{smallmatrix}\right]$, $\Gamma_{\mathrm{r}} = \left[\begin{smallmatrix} 0.01 & 0 \\ 0 & 0.01 \end{smallmatrix}\right]$, and $Q = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$.

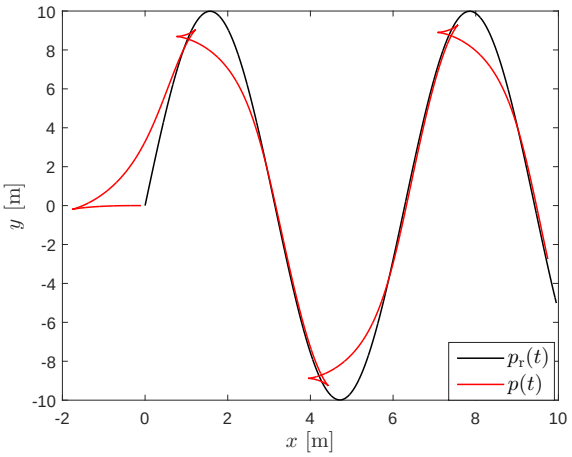

Figure 4.2: Sinusoidal trajectory without saturation

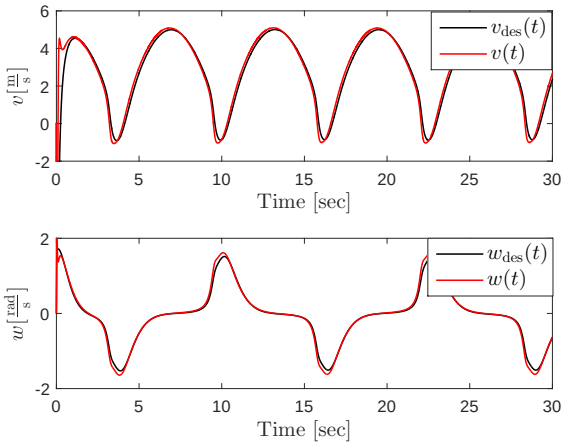

Figure 4.3: Velocity without saturation

Unsaturated Waypoint Reference. For this category, the way-point dictated trajectory is shown below in Figure 4.4 and the actual and desired velocities compared are shown in Figure 4.5. The following model parameters were used: $A^* = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$, and $B^* = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$. The following initialization values were used: $x_{\mathrm{r}}(0) = 0$, $y_{\mathrm{r}}(0) = 0$, $x(0) = 20$, $y(0) = 1$, $\theta(0) = 0$,

$d(0) = 0.1$, $v(0) = 0$, $\omega(0) = 0$, $v_{\mathrm{d}}(0) = 0$, $\omega_{\mathrm{d}}(0) = 0$, $\theta_s(0) = \left[\begin{smallmatrix} -4 & 0 \\ 0 & -4 \end{smallmatrix}\right]$, and $\theta_{\mathrm{r}}(0) = \left[\begin{smallmatrix} 2 & 0 \\ 0 & 2 \end{smallmatrix}\right]$. The following tuning parameters were used: $k_v = 1$, $k_w = 1$, $\alpha = 0.5$, $\beta = 0.1$, $\lambda = 1$, $\Gamma_s = \left[\begin{smallmatrix} 0.01 & 0 \\ 0 & 0.01 \end{smallmatrix}\right]$, $\Gamma_{\mathrm{r}} = \left[\begin{smallmatrix} 0.01 & 0 \\ 0 & 0.01 \end{smallmatrix}\right]$, and $Q = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$. And finally, the following tuning parameters for the potential field algorithm were used: $F_{\mathrm{ac}} = 10$, $F_{\mathrm{rc}} = 8$, $W = 2.5$, $n = 2$, and $E = 8$.

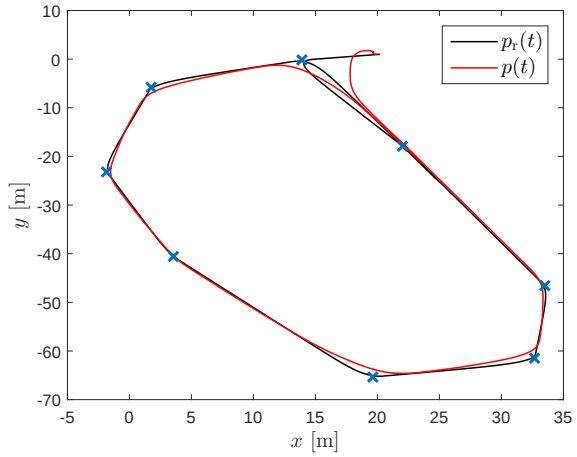

Figure 4.4: Waypoint trajectory without saturation

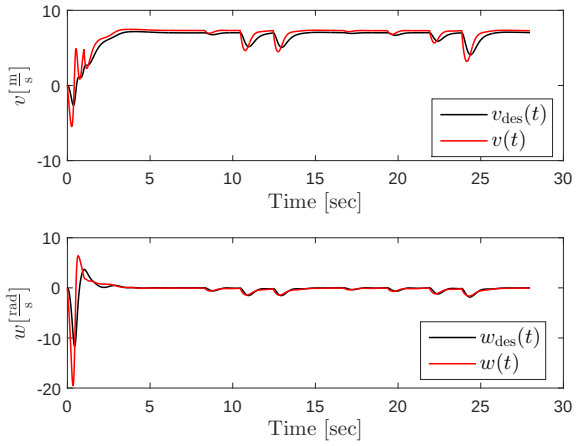

Figure 4.5: Velocity without saturation

Analysis of Results. As seen in Figures 4.2 and 4.4, the simulated vehicle is able to follow the reference trajectory. In addition as seen in Figure 4.3 and 4.5, the longitudinal velocity and angular velocity are able to follow the desired values. Both the trajectory and velocities act similarly to the version of the predator-prey interaction controller without the adaptive backstepping as seen in [4]. However, similar to [4], many of the largest angular rate commands, with regards to magnitude, correspond to the lowest velocity commands. This ratio experiences singularities. Furthermore, the desired velocity at times demands that the

vehicle go backwards which does not agree with the biological nature of a predator chasing the prey. Therefore, we implement the velocity saturation as shown in [4] to enforce a more intuitive predator-prey model.

4.4 Simulation Results with Velocity Saturation

In this section, the effectiveness of the direct MRAC backstepping controller with the addition of the velocity saturation is investigated through the use of computer simulation. This section include the results and analysis of the results. All simulations presented were carried out with either a trajectory dictated by a sine wave or a trajectory dictated by GPS waypoints as a reference system.

Saturated Sine Reference. For this category, the sine wave dictated trajectory is shown, with velocity saturation in Figure 4.6 and the actual and desired velocities are compared in Figure 4.7. The following model parameters were used: $A^* = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$, and $B^* = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$. The following initialization values were used: $x_{\mathrm{r}}(0) = 0$, $y_{\mathrm{r}}(0) = 0$, $x(0) = 20$, $y(0) = 1$, $\theta(0) = 0$, $d(0) = 0.1$, $v(0) = 0$, $\omega(0) = 0$, $v_{\mathrm{d}}(0) = 0$, $\omega_{\mathrm{d}}(0) = 0$, $\theta_s(0) = \left[\begin{smallmatrix} -4 & 0 \\ 0 & -4 \end{smallmatrix}\right]$, and $\theta_{\mathrm{r}}(0) = \left[\begin{smallmatrix} 2 & 0 \\ 0 & 2 \end{smallmatrix}\right]$. The following tuning parameters were used: $k_v = 1$, $k_w = 1$, $\alpha = 0.5$, $\beta = 0.1$, $\lambda = 1$, $\Gamma_s = \left[\begin{smallmatrix} 0.01 & 0 \\ 0 & 0.01 \end{smallmatrix}\right]$, $\Gamma_{\mathrm{r}} = \left[\begin{smallmatrix} 0.01 & 0 \\ 0 & 0.01 \end{smallmatrix}\right]$, and $Q = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$. And finally the tuning parameters for the velocity saturation were used: $d^* = 0.1$, $\dot{d}^* = 0$, $strip = 0.1$, $w_d = 2$, $\zeta_d = 0.85$, $L = 0.3556$m, $\gamma_{\max} = 25^o$, $v_{\max} = 10\frac{m}{s}$, and $v_{\min} = 1\frac{m}{s}$.

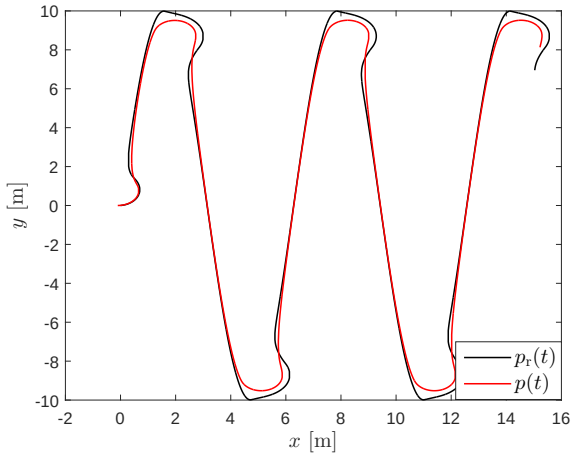

Figure 4.6: Sinusoidal trajectory with saturation

Saturated Waypoint Reference. For this category, the way-point dictated trajectory is shown, with velocity saturation in Figure 4.8 and the actual and desired velocities are compared in Figure 4.9. The following model parameters were used: $A^* = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$, and

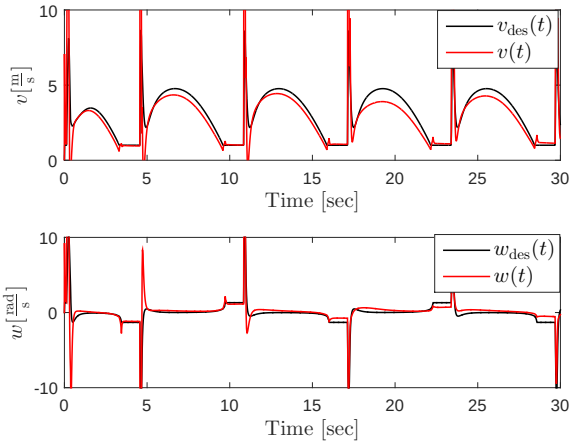

Figure 4.7: Velocity with saturation

$B^* = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$. The following initialization values were used: $x_{\text{r}}(0) = 0$, $y_{\text{r}}(0) = 0$, $x(0) = 20$, $y(0) = 1$, $\theta(0) = 0$, $d(0) = 0.1$, $v(0) = 0$, $\omega(0) = 0$, $v_{\text{d}}(0) = 0$, $\omega_{\text{d}}(0) = 0$, $\theta_s(0) = \left[\begin{smallmatrix} -4 & 0 \\ 0 & -4 \end{smallmatrix}\right]$, and $\theta_{\text{r}}(0) = \left[\begin{smallmatrix} 2 & 0 \\ 0 & 2 \end{smallmatrix}\right]$. The following tuning parameters were used: $k_v = 1$, $k_w = 1$, $\alpha = 0.5$, $\beta = 0.1$, $\lambda = 1$, $\Gamma_s = \left[\begin{smallmatrix} 0.01 & 0 \\ 0 & 0.01 \end{smallmatrix}\right]$, $\Gamma_{\text{r}} = \left[\begin{smallmatrix} 0.01 & 0 \\ 0 & 0.01 \end{smallmatrix}\right]$, and $Q = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$. The tuning parameters for the velocity saturation were used: $d^* = 0.1$, $\dot{d}^* = 0$, $strip = 0.1$, $w_d = 2$, $\zeta_d = 0.85$, $L = 0.3556$m, $\gamma_{\max} = 25^o$, $v_{\max} = 10\frac{m}{s}$, and $v_{\min} = 1\frac{m}{s}$. And finally the following tuning parameters for the potential field algorithm were used: $F_{\text{ac}} = 10$, $F_{\text{rc}} = 8$, $W = 2.5$, $n = 2$, and $E = 8$.

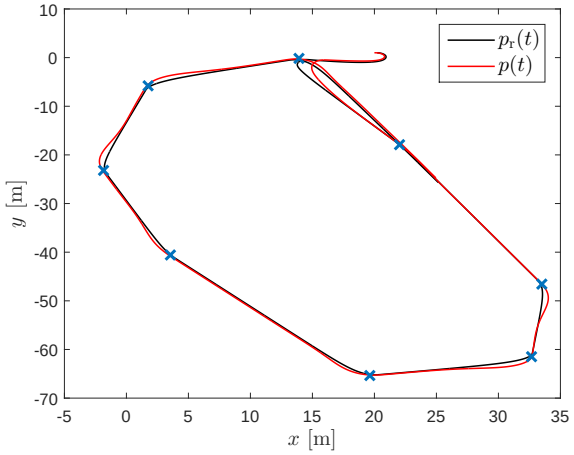

Figure 4.8: Waypoint trajectory with saturation

Analysis of Results. Figure 4.6 and 4.8 show the resulting trajectories with the addition of the velocity saturation for the two different reference systems. In both cases,

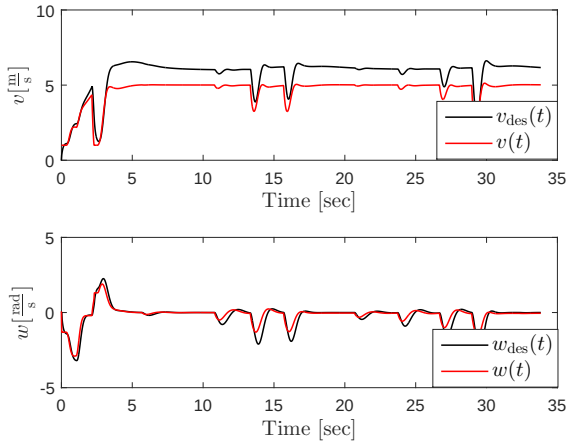

Figure 4.9: Velocity with saturation

the vehicle is still following the desired trajectory set by the reference system. In the case of sharp turns, instead of having negative longitudinal velocity, the vehicle continues on forward in a circular path until it can rejoin the desired path. Therefore it is observed that the vehicle is closer to the desired behavior of a predator-prey interaction. In Figure 4.7 and 4.9, the velocity is between $1\frac{m}{s}$ and $10\frac{m}{s}$ which is in the desired range. It can be concluded that the results shown here with the direct MRAC backstepping is comparable and similar to the results in [4] with the addition of the velocity saturation.

4.5 Experimental Implementation and Results

In this section, the experimental results used to evaluate the controller performance will be discussed. All experiments were carried out with waypoints dictated trajectory as a reference. This section consists of the description of the experimental setup, Projection algorithm, and the acceleration saturation algorithm implemented on the controller and the resulting controller behavior.

Experimental Setup. For testing, the vehicle featured in Figure 4.10 was used. The vehicle hosts an Inertial Measurement Unit (IMU) and GPS. The IMU senses Euler angles with an accuracy of $\pm 2^o$, acceleration on 3 axes with an accuracy of $\pm 0.01g$, and angular rate with an accuracy of $\pm 0.2^o/sec$. The GPS provides position feedback with an accuracy of $\pm$1.5m, and velocity measurements with an accuracy of $0.5\frac{\text{m}}{\text{s}}$. The sensor data from the IMU and GPS were merged by an extended Kalman filter, developed specifically for vehicle. The filter provides data on position, heading,and velocity.

The actuation on the vehicle is provided by a brushless motor for speed and a servo for steering, both of the motors are controlled using a servo controller. The controller, Kalman

filter and overall algorithm are run on a onboard computer using LabVIEW at an rate of 40 Hz with a time step of 0.025 seconds.

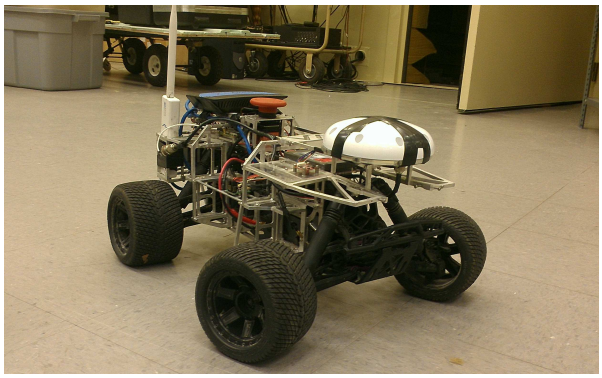

Figure 4.10: Experimental testing platform.

4.5.1 Projection

An adaptive control system yields Lyapunov stability and convergence of the tracking error to 0, provided that the system is disturbance free. However, it has been observed and proven that in applications with unknown bounded disturbances, parameter drift can occur, where the estimated parameters drift to infinity with time ([12, 13]). This phenomenon is corrected using the projection algorithm, where the estimated parameter is subject to some constraints. In other words, the estimated parameter is only allowed to adapt within the specified bounds. The algorithm is defined as shown below ([14]). We consider a convex compact set with a smooth boundary given by

$$\Omega_c = \{\theta \in \mathbb{R}^n | f(\theta) \leq c\}, \tag{4.42}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a smooth convex function given by

$$f(\theta) = \frac{\theta^{\mathrm{T}}\theta - \theta_{\max}^2}{\epsilon_\theta \theta_{\max}^2}, \tag{4.43}$$

where θ_{max} is the norm bound imposed on the parameter vector θ, and ϵ_θ is the convergence tolerance of our choice. The true value of the parameter θ, is defined by θ^* and belongs to Ω_0. Then, the projection algorithm is defined as follows

$$\mathrm{Proj}(\theta, \dot{\theta}) = \begin{cases} \dot{\theta} & \text{if} f(\theta) < 0, \\ \dot{\theta} & \text{if} f(\theta) \geq 0, \text{and} \nabla f^{\mathrm{T}} \dot{\theta} \leq 0, \\ \dot{\theta} - \frac{\nabla f}{\|\nabla f\|} \langle \frac{\nabla f^{\mathrm{T}}}{\|\nabla f\|}, \dot{\theta} \rangle f(\theta) & \text{if} f(\theta) \geq 0, \text{and} \nabla f^{\mathrm{T}} \dot{\theta} > 0. \end{cases} \tag{4.44}$$

Considering that the function $f(\theta)$ is a convex, the following property follows

$$(\theta^* - \theta)^{\mathrm{T}} \nabla f(\theta) \leq f(\theta^*) - f(\theta) \tag{4.45}$$

which guarantees that the projection algorithm does not affect the sign of the Lyapunov derivative $\dot{V}_{\mathrm{a}}(t)$ negativity.

4.5.2 Acceleration Saturation

The predator-prey controller commands longitudinal and angular velocities based on the error between the position of the vehicle and the reference system, without any bounds on the longitudinal acceleration of the vehicle. It can command velocities in the range of $1\frac{m}{s}$ to $10\frac{m}{s}$ and can command velocities that increase drastically in a very short time. In other words, the acceleration can be arbitrarily high. This proved to be an issue when the direct MRAC controller was implemented and tested as the vehicle would accelerate too fast and caused the front wheels to come off the ground due to the sudden torque applied to the rear wheels (wheelie). To eliminate this undesired behavior, the following acceleration saturation algorithm is developed and implemented as given

$$v_{\mathrm{d}} = \begin{cases} \min(v_{\max}, v_{\mathrm{d}}^{\mathrm{old}} + a_{\max}\Delta t) & \text{if } v_{\mathrm{d}} > (v_{\mathrm{d}}^{\mathrm{old}} + a_{\max}\Delta t), \\ \max(v_{\min}, v_{\mathrm{d}}^{\mathrm{old}} - a_{\max}\Delta t) & \text{if } v_{\mathrm{d}} < (v_{\mathrm{d}}^{\mathrm{old}} - a_{\max}\Delta t), \end{cases} \tag{4.46}$$

where $v_{\max}(v_{\min})$ is the maximum(minimum) velocity allowed, $a_{\max}$ is the maximum acceleration allowed, $v_{\mathrm{d}}^{\mathrm{old}}$ is the desired longitudinal velocity at time $t - \Delta t$, and Δt is a fixed time step. The acceleration saturation does not allow the vehicle to accelerate or decelerate faster than the assigned $a_{\max}$.

Maintaining Lyapunov Criteria. The acceleration saturation algorithm alters the virtual controller commands as needed. In order to verify that the asymptotic stability of the error dynamics is still obtained, the virtual controller commands are backsolved to alter the reference system using the given equation.

$$\dot{p}_{\mathrm{r}}^{\mathrm{sat}}(t) = R(\theta(t))(\Delta(t) \begin{bmatrix} v^{\mathrm{sat}}(t) \\ \omega^{\mathrm{sat}}(t) \end{bmatrix} - K \tanh(e(t) - \delta(t)) + \dot{\delta}(t)), \tag{4.47}$$

where $\dot{p}_{\mathrm{r}}^{\mathrm{sat}}(t)$ is the new resulting time derivative, and v^{sat} and ω^{sat} are the saturated velocities after the saturation algorithm. More details can be seen in [4].

4.5.3 Results

This section presents the experimental results for the direct MRAC backstepping. Each test is run with the same set of waypoints. The reference system follows a virtual target based on the potential field algorithm presented earlier. In the three test results that is presented, the reference velocities are set to constant values of $2\frac{m}{s}, 5\frac{m}{s}$, and $9\frac{m}{s}$.

Results for $2\frac{m}{s}$. The following figures showcase the results of the vehicle testing when the direct MRAC backstepping controller was implemented with the reference velocity set to $v_{\text{rmax}} = 2\frac{m}{s}$. The following initialization values were used: $x_{\text{r}}(0) = 2.16$, $y_{\text{r}}(0) = 0.714$, $x(0) = 8.18$, $y(0) = 4.11$, $\theta(0) = 0$, $d(0) = 0.1$, $v(0) = 0.057$, $\omega(0) = -0.93$, $v_{\text{d}}(0) = 0.048$, $\omega_{\text{d}}(0) = 4.68$, $\theta_s(0) = \begin{bmatrix} -0.005 & 0 \\ 0 & -0.005 \end{bmatrix}$, and $\theta_{\text{r}}(0) = \begin{bmatrix} 0.02 & 0 \\ 0 & 0.02 \end{bmatrix}$. The following tuning parameters for the controller were used: $k_v = 1$, $k_w = 1$, $\alpha = 0.5$, $\beta = 0.1$, $\lambda = 1$, $\Gamma_s = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}$, $\Gamma_{\text{r}} = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}$, and $Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$. The following tuning parameters for the potential field algorithm were used: $F_{\text{ac}} = 10$, $F_{\text{rc}} = 8$, $W = 2.5$, $n = 2$, and $E = 8$. And finally, the following tuning parameters for the velocity saturation were used: $d^* = 0.1$, $\dot{d}^* = 0$, $strip = 0.1$, $w_d = 2$, $\zeta_d = 0.85$, $L = 0.3556$m, $\gamma_{\max} = 25^o$, $v_{\max} = 10\frac{m}{s}$, and $v_{\min} = 1\frac{m}{s}$.

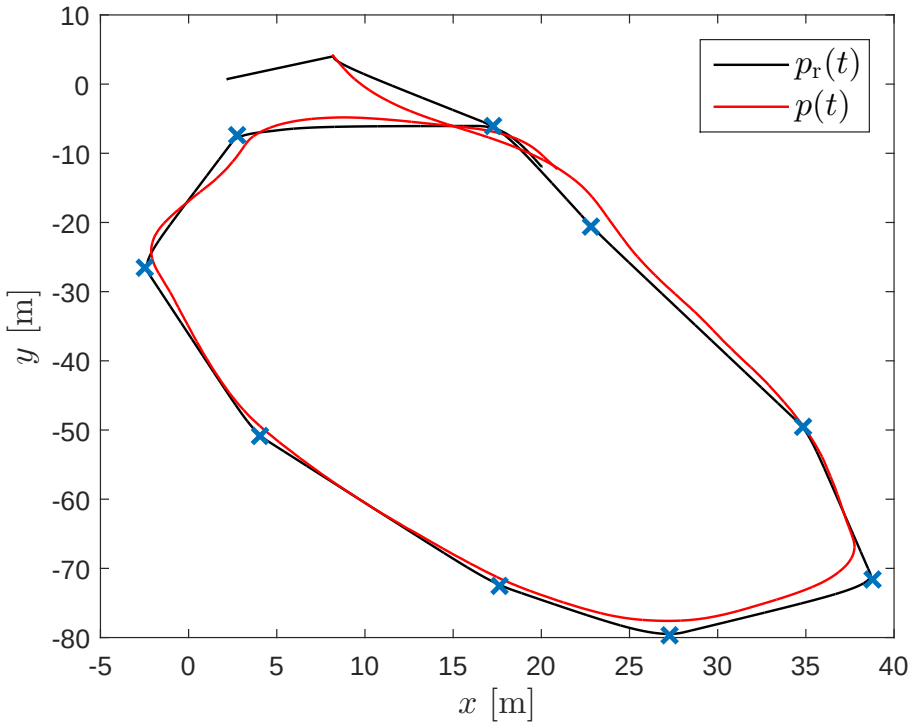

Figure 4.11: Trajectory at 2 $\frac{m}{s}$

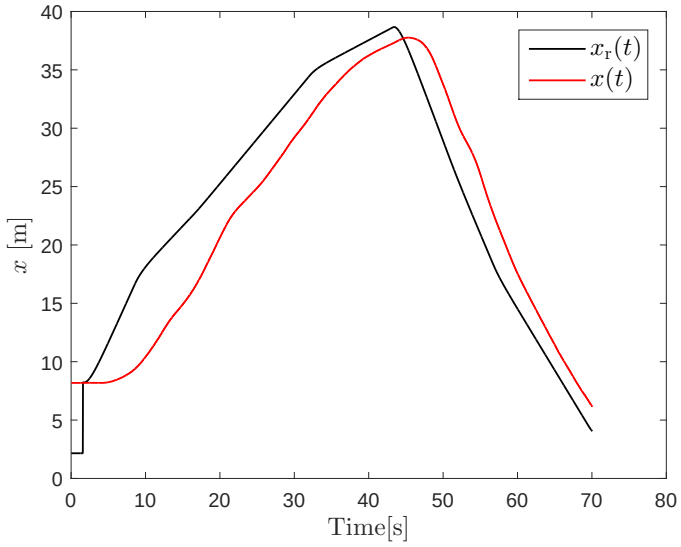

Figure 4.12: x Response at 2 $\frac{m}{s}$

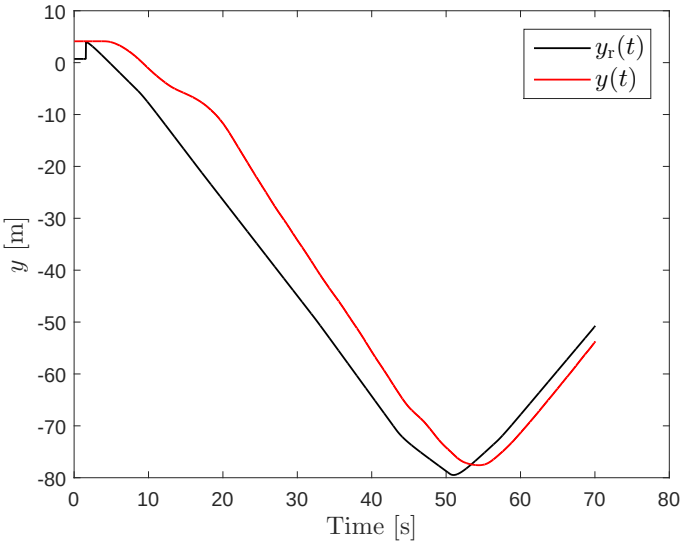

Figure 4.13: y Response at 2 $\frac{m}{s}$

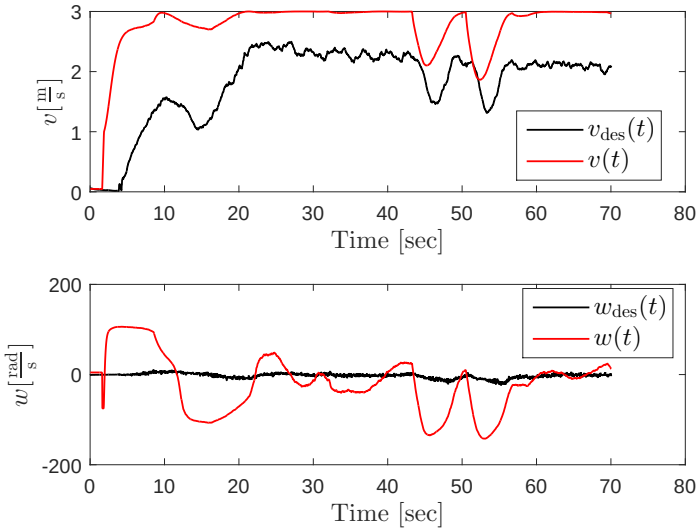

Figure 4.14: Velocity at 2 $\frac{m}{s}$

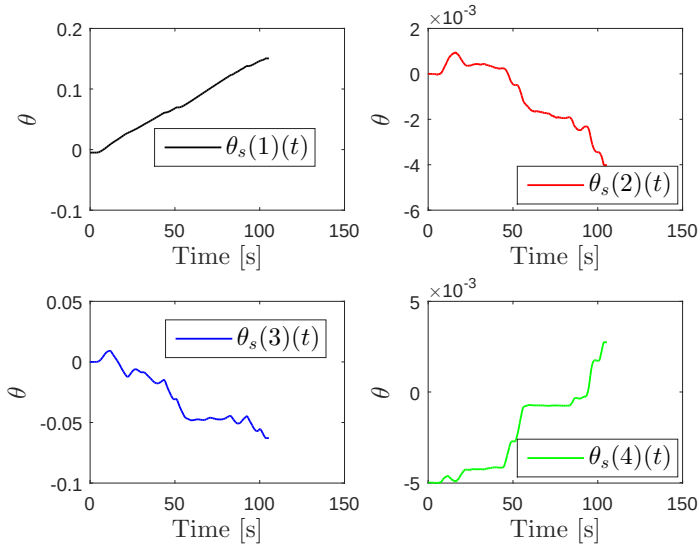

Figure 4.15: θ_s at 2 $\frac{m}{s}$

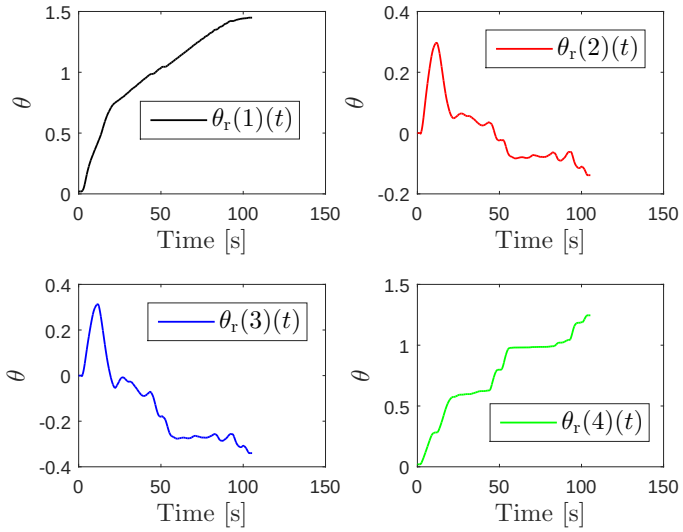

Figure 4.16: θ_r at 2 $\frac{m}{s}$

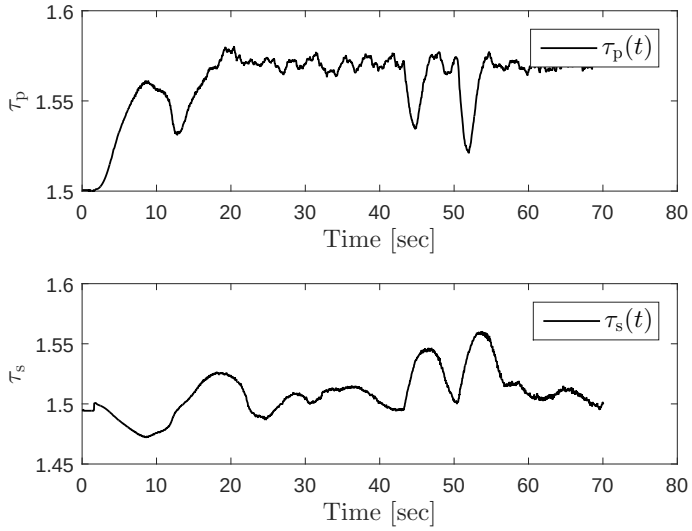

Figure 4.17: Commands signals to motors at $2\frac{m}{s}$

Results for $5\frac{m}{s}$. The following figures showcase the results of the vehicle testing when the direct MRAC backstepping controller was implemented with the reference velocity set to $v_{\text{rmax}} = 5\frac{m}{s}$. The following initialization values were used: $x_{\text{r}}(0) = 20.01$, $y_{\text{r}}(0) = -11.93$, $x(0) = 8.58$, $y(0) = 6.26$, $\theta(0) = 0$, $d(0) = 0.1$, $v(0) = 0.01$, $\omega(0) = 0.23$, $v_{\text{d}}(0) = 2.85$, $\omega_{\text{d}}(0) = -59.29$, $\theta_s(0) = \begin{bmatrix} -0.005 & 0 \\ 0 & -0.005 \end{bmatrix}$, and $\theta_{\text{r}}(0) = \begin{bmatrix} 0.02 & 0 \\ 0 & 0.02 \end{bmatrix}$. The following tuning parameters for the controller were used: $k_v = 1$, $k_w = 1$, $\alpha = 0.5$, $\beta = 0.1$, $\lambda = 1$, $\Gamma_s = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}$, $\Gamma_{\text{r}} = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}$, and $Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$. The following tuning parameters for the potential field algorithm were used: $F_{\text{ac}} = 10$, $F_{\text{rc}} = 8$, $W = 2.5$, $n = 2$, and $E = 8$. And finally, the following tuning parameters for the velocity saturation were used: $d^* = 0.1$, $\dot{d}^* = 0$, $strip = 0.1$, $w_d = 2$, $\zeta_d = 0.85$, $L = 0.3556\text{m}$, $\gamma_{\text{max}} = 25^o$, $v_{\text{max}} = 10\frac{m}{s}$, and $v_{\text{min}} = 1\frac{m}{s}$.

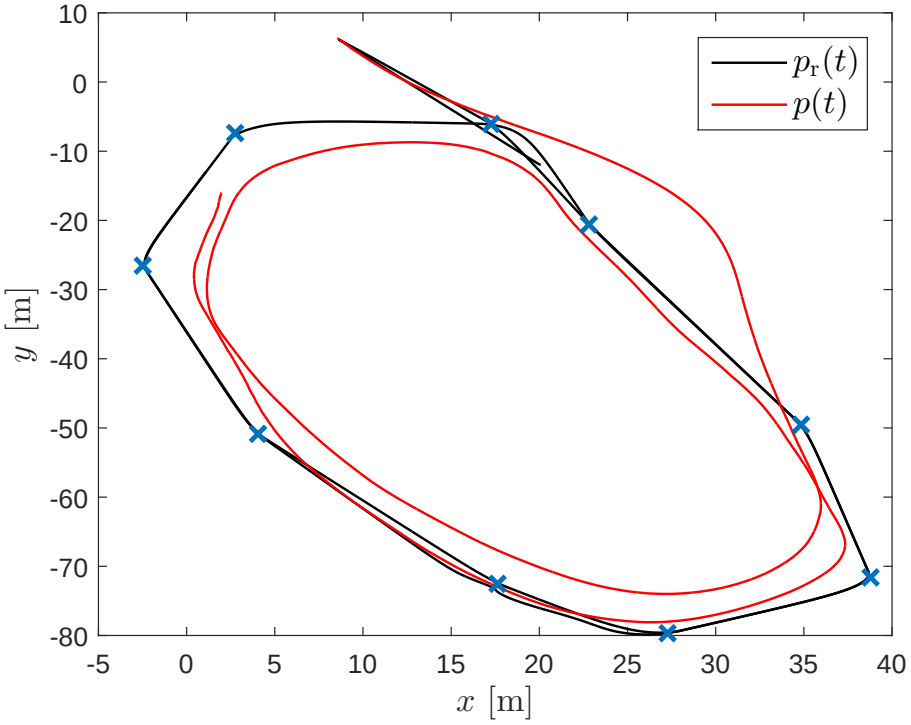

Figure 4.18: Trajectory at $5\ \frac{m}{s}$

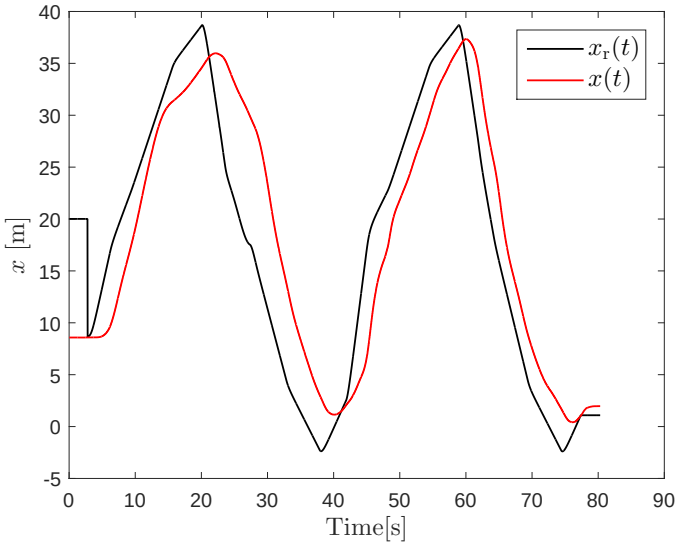

Figure 4.19: x Response at 5 $\frac{m}{s}$

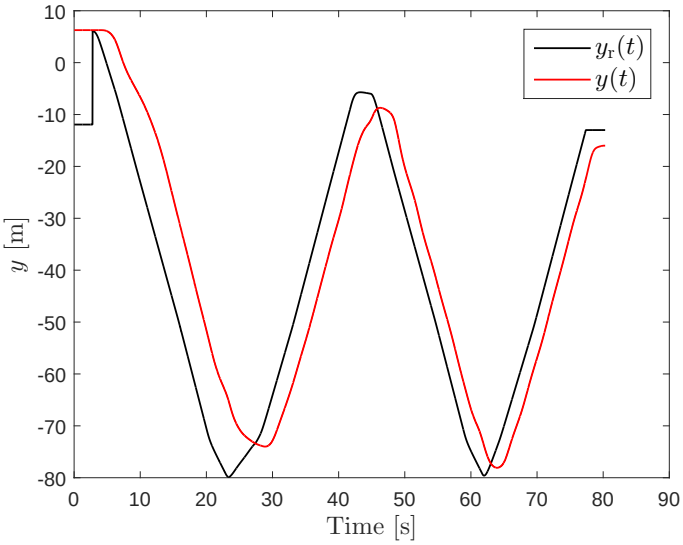

Figure 4.20: y Response at 5 $\frac{m}{s}$

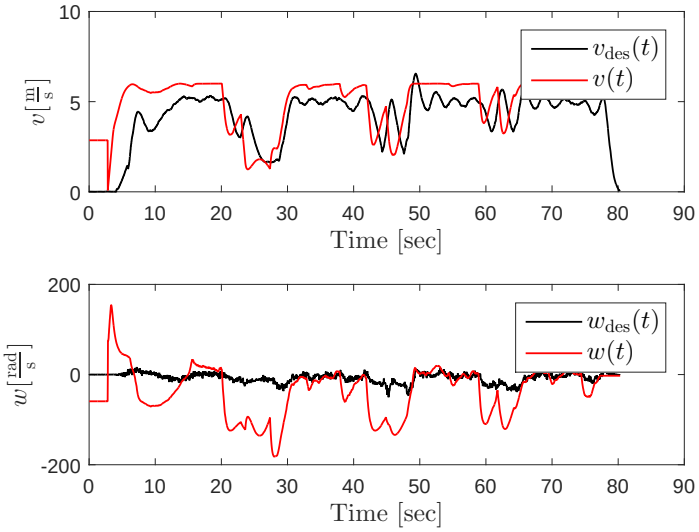

Figure 4.21: Velocities at 5 $\frac{m}{s}$

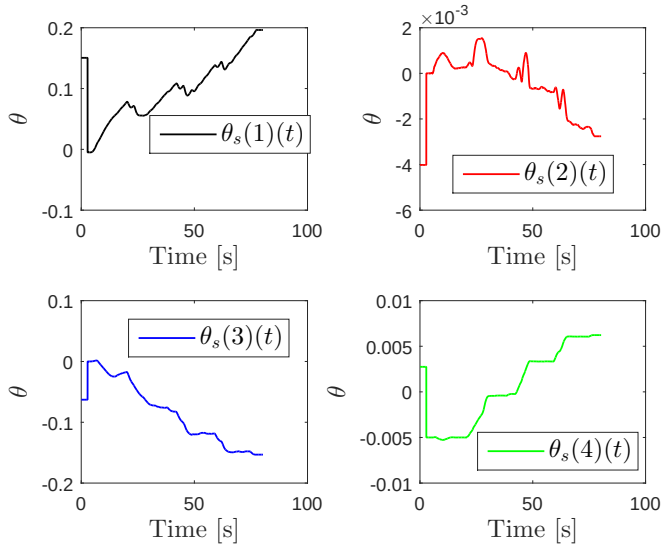

Figure 4.22: θ_s at 5 $\frac{m}{s}$

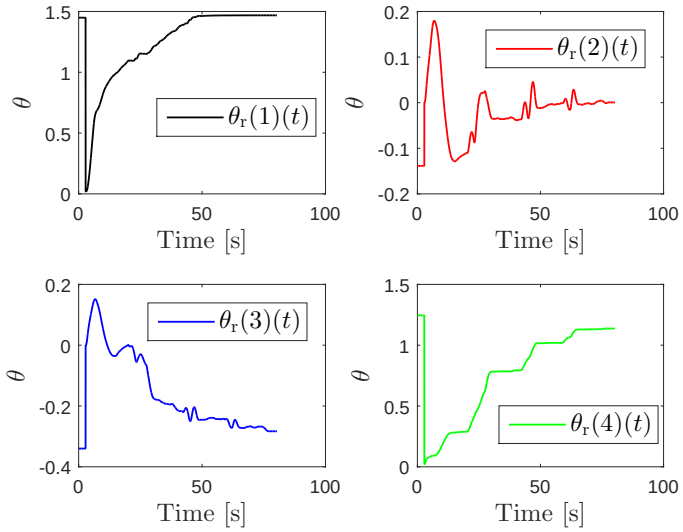

Figure 4.23: θ_r at 5 $\frac{m}{s}$

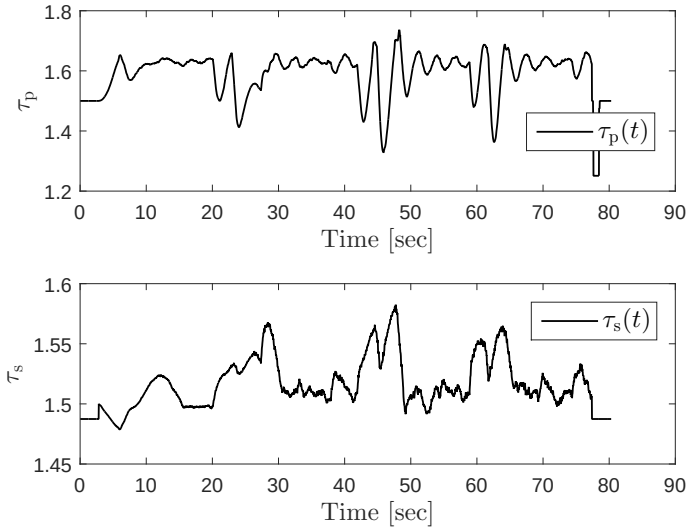

Figure 4.24: Commands signals to motors at 5 $\frac{m}{s}$

Results for 9 $\frac{m}{s}$. The following figures showcase the results of the vehicle testing when the direct MRAC backstepping controller was implemented with the reference velocity set to $v_{\text{rmax}} = 9\frac{m}{s}$. The following initialization values were used: $x_{\text{r}}(0) = 1.09$, $y_{\text{r}}(0) = -12.99$, $x(0) = 8.79$, $y(0) = 5.19$, $\theta(0) = 0$, $d(0) = 0.1$, $v(0) = 0.08$, $\omega(0) = 0.17$, $v_{\text{d}}(0) = 5.91$, $\omega_{\text{d}}(0) = -2.62$, $\theta_s(0) = \begin{bmatrix} -0.005 & 0 \\ 0 & -0.005 \end{bmatrix}$, and $\theta_{\text{r}}(0) = \begin{bmatrix} 0.02 & 0 \\ 0 & 0.02 \end{bmatrix}$. The following tuning parameters for the controller were used: $k_v = 1$, $k_w = 1$, $\alpha = 0.5$, $\beta = 0.1$, $\lambda = 1$, $\Gamma_s = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}$, $\Gamma_{\text{r}} = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}$, and $Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$. The following tuning parameters for the potential field algorithm were used: $F_{\text{ac}} = 10$, $F_{\text{rc}} = 8$, $W = 2.5$, $n = 2$, and $E = 8$. Finally, The following tuning parameters for the velocity saturation were used: $d^* = 0.1$, $\dot{d}^* = 0$, $strip = 0.1$, $w_d = 2$, $\zeta_d = 0.85$, $L = 0.3556$m, $\gamma_{\text{max}} = 25^o$, $v_{\text{max}} = 10\frac{m}{s}$, and $v_{\text{min}} = 1\frac{m}{s}$.

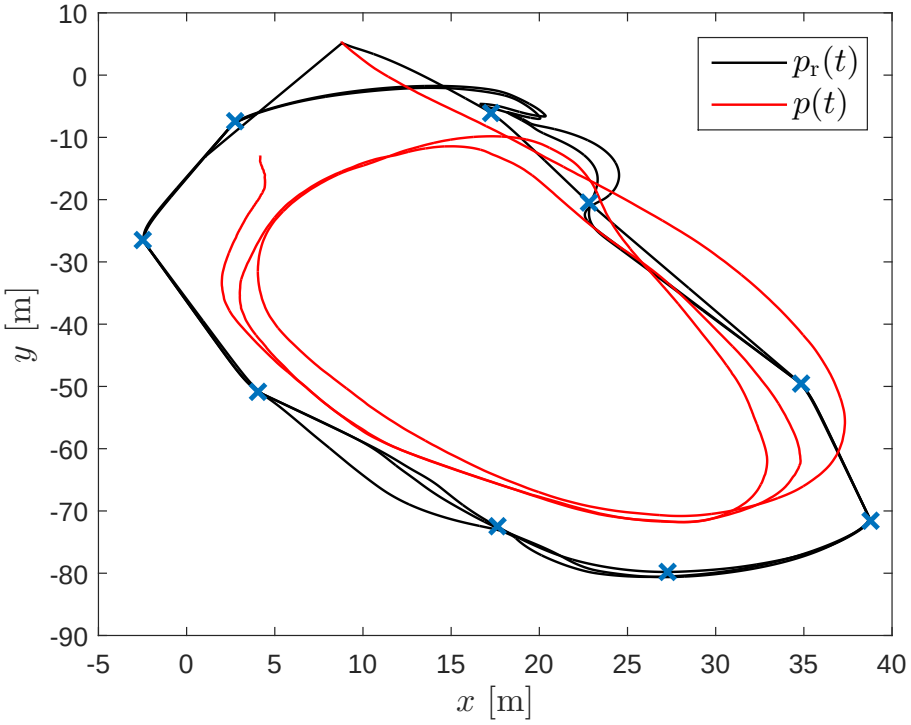

Figure 4.25: Trajectory at 9 $\frac{m}{s}$

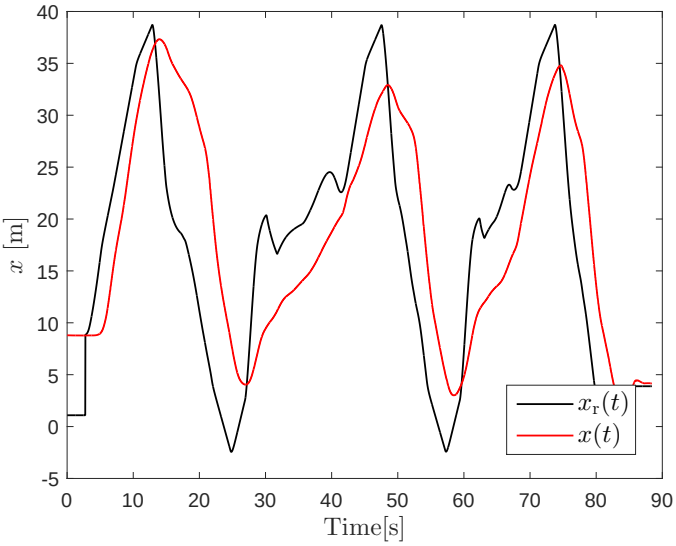

Figure 4.26: x Response at 9 $\frac{m}{s}$

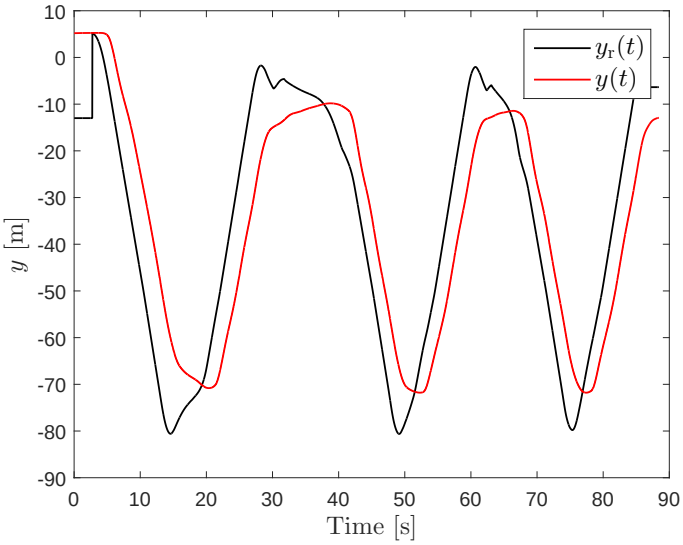

Figure 4.27: y Response at 9 $\frac{m}{s}$

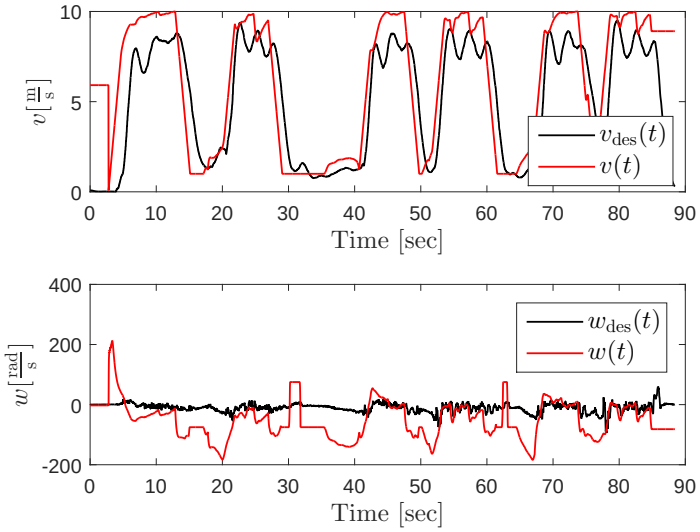

Figure 4.28: Velocities at 9 $\frac{m}{s}$

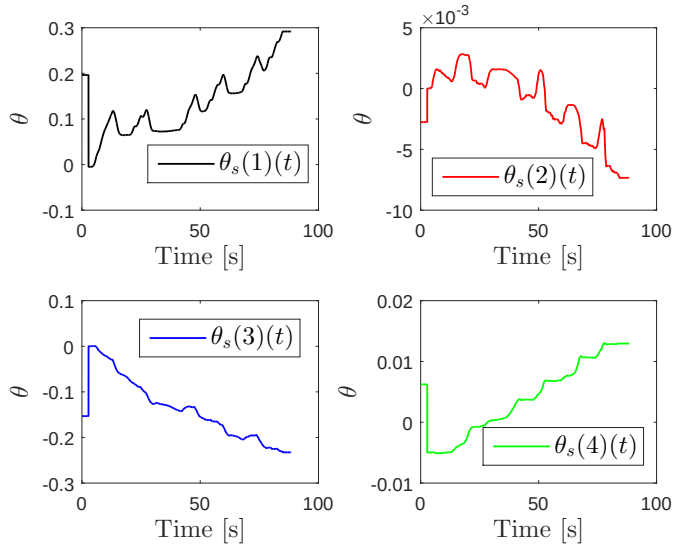

Figure 4.29: θ_s at 9 $\frac{m}{s}$

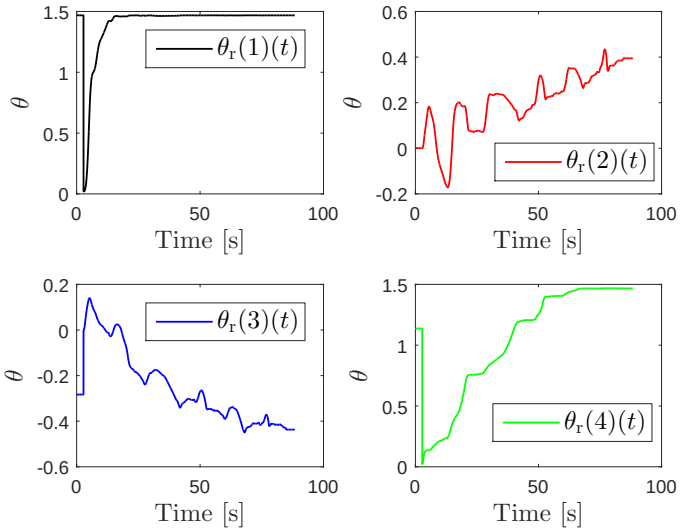

Figure 4.30: θ_r at 9 $\frac{m}{s}$

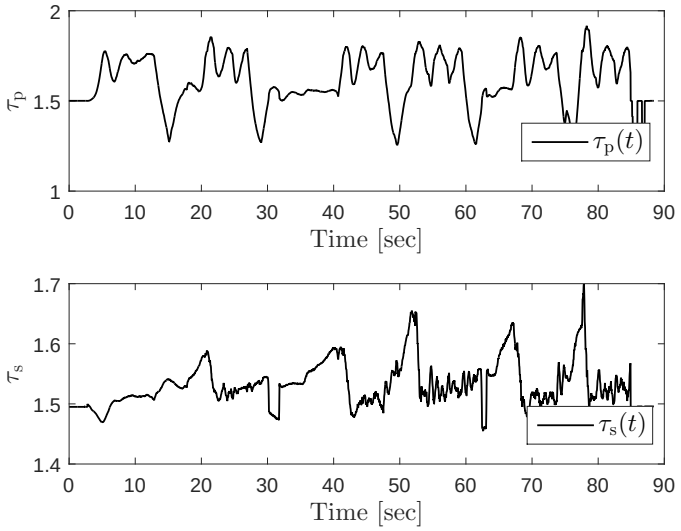

Figure 4.31: Commands signals to motors at 9 $\frac{m}{s}$

Analysis of Results. For the **low** speed of $2\frac{m}{s}$, as seen in Figure 4.11, the vehicle maintains a small following distance and therefore tracks the reference trajectory tightly. This behavior is expected because the following distance between the vehicle and the reference system is proportional to the longitudinal velocity of the vehicle. Due to this close following distance and the time it takes for the direct MRAC backstepping controller to adapt to the varying dynamics, the vehicle slightly deviates from the reference trajectory at sharp turns. While there are minor deviations at considerably high sharp turns, the overall low speed performance is favorable.

Figures 4.12 and 4.13 show the desired positions along with the actual positions. It can be observed that the actual position matches the reference position closely but with a slight lag. Figure 4.14 shows the comparison of the desired velocities with the actual velocities. Figures 4.15 and 4.16 show the controller parameter gains adapting with passing time. Figure 4.17 shows the $\tau(t)$ values commanded to the motors of the vehicle.

For the **medium** speed of $5\frac{m}{s}$, as seen in Figure 4.18, at first, the direct MRAC adaptive backstepping controller on the vehicle is adapting and therefore the vehicle does not closely follow the reference system. Over time, the vehicle tracks the reference system more closely and eventually the vehicle starts cutting inside the reference system's path. This behavior is ideal based on the derivation of the controller. As velocity increases, the following distance proportionally increases and the vehicle starts correcting earlier than when compared to the behavior at lower speeds. Therefore as a result, the vehicle starts moving into the reference system's path which can be confirmed in the experimental results.

Figures 4.19 and 4.20 show the desired positions compared to the actual positions. It can be observed that the actual positions matches the reference positions closely but with a slight lag as before. Figure 4.21 shows the comparison of the desired velocities with the actual velocities. Figures 4.22 and 4.23 show the controller parameter gains adapting with passing time. Figure 4.24 shows the $\tau(t)$ values commanded to the motors of the vehicle.

For the **high** speed of $9\frac{m}{s}$, as seen in Figure 4.25, the behavior of the vehicle is similar to that of the medium speed but more prominent. Due to higher speed and longer following distance, the vehicle cuts inside the reference system's path more quickly than that of the medium speed. The choppiness of the reference system at the turns is due to the acceleration saturation implemented. As the velocity is saturated to minimize excessive accelerations, the reference system is accordingly altered to ensure the asymptotic stability of the error dynamics. The choppiness, however, does not affect the actual vehicle path adversely. This behavior is collaborated by the proposed predator prey model which designs a smooth trajectory even if the prey is traveling chaotically.

Figures 4.26 and 4.27 show the desired positions compared to the actual positions. It can be observed that the actual position matches the reference position closely but with a slight lag as before. Figure 4.21 shows the comparison of the desired velocities with the

actual velocities. Figures 4.28 and 4.29 show the controller parameter gains adapting with passing time. Figure 4.31 shows the $\tau(t)$ values commanded to the motors of the vehicle.

In [4], the vehicle dynamics is not accounted for in the controller design. Therefore, during the experimental testing, at sharp turns the vehicle crossed over to the outside of the reference system's path due to the vehicle's drastically varying dynamics. Moreover, the controller needed extensive parameter tuning based on particular trajectory, curvature and velocity of the vehicle. With the incorporation of the direct MRAC adaptive backstepping controller with the predator-prey model, it is shown both in simulation and experimental results that the controller can now adapt to the changing dynamics and control the vehicle to follow its virtual target inside the reference system's path and allow the vehicle to run at different speeds with the same set parameters.

4.6 Summary for Controller

The table below summarize the direct MRAC backstepping controller.

Table 4.1: Summary of the Direct MRAC Backstepping Controller

System	$\dot{p}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & 0 \\ \sin(\theta(t)) & 0 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}$ $\dot{\theta}(t) = \omega(t)$ $\dot{s} = A(t)s(t) + B(t)\tau(t)$
Virtual Controller	$\alpha(t) = \begin{bmatrix} v_{\mathrm{d}}(t) \\ \omega_{\mathrm{d}}(t) \end{bmatrix} = \Delta^{-1}(t)(K\tanh(e(t) - \delta(t)) + R^{\mathrm{T}}(\theta(t))\dot{p}_{\mathrm{r}}(t) - \dot{\delta}(t))$
Torque Controller	$\hat{\tau}(t) = \hat{\theta}_s(t)s(t) + \hat{\theta}_{\mathrm{r}}(t)\eta(t)$
Error Dynamics	$\dot{e}(t) = -S(\omega(t))(e(t) - \delta(t)) + R^{\mathrm{T}}(\theta(t))\dot{p}_{\mathrm{r}}(t) - \Delta(t)s(t)$ $\dot{e}_1(t) = -S(\omega(t))e_1(t) - K\tanh(e_1(t)) - \Delta(t)e_2(t)$ $\dot{e}_2(t) = -Qe_2(t) + \Delta(t)e_1(t) + B(t)\tilde{\theta}_s(t)s(t) + B(t)\tilde{\theta}_{\mathrm{r}}(t)\eta(t)$
Adaptive Laws	$\dot{\theta}_s(t) = -e_2(t)s^{\mathrm{T}}(t)\Gamma_s$ $\dot{\theta}_{\mathrm{r}}(t) = -e_2(t)\eta^{\mathrm{T}}(t)\Gamma_{\mathrm{r}}$

Chapter 5

$\mathcal{L}_1$ Adaptive Backstepping

In this chapter, we are going to introduce a different control system which uses a combination of $\mathcal{L}_1$ adaptive control and backstepping processes. The chapter demonstrates an alternative to the direct MRAC adaptive backstepping system presented in Chapter 4. The approach taken here is similar to that in [15]. Like the direct MRAC backstepping controller, this $\mathcal{L}_1$ adaptive backstepping controller is developed in two steps. For the first step, we utilize the concepts of backstepping to derive the desired torque signal for the given vehicle assuming that there is no parameter uncertainties. The second step uses the $\mathcal{L}_1$ adaptive control approach to derive the adaptive torque controller to handle the model parameter uncertainties.

5.1 Theoretical Framework

$\mathcal{L}_1$ adaptive control is the design of robust adaptive controllers using fast adaptation schemes. The key feature of the theory is the decoupling of the adaptation loop from the control loop, allowing for fast adaptation without sacrificing robustness. The adaptation rate is limited only by the available control channel bandwidth. In [16], the authors observed that fast adaptation using the MRAC architecture reduces the phase margin of the MRAC architecture to zero, hence the faster adaptation gain introduces a tradeoff between performance and robustness. On the other hand, it was shown that an $\mathcal{L}_1$ controller is beneficial for both performance and robustness as faster adaptation does not force a trade-off. A more detailed explanation of the theory and derivation of $\mathcal{L}_1$ adaptive control can be found in [16, 17].

5.2 $\mathcal{L}_1$ Adaptive for Vehicle

5.2.1 Represented Vehicle Dynamic Model

The dynamic model considered for the controller design was obtained in Chapter 1 and is given by

$$\dot{s}(t) = As(t) + B\tau(t). \tag{5.1}$$

The dynamic model can be broken down to represent the baseline values of vehicle dynamics by A_b and B_b, and the uncertainties of the vehicle dynamics by A_ι and B_ι,

$$\begin{aligned} \dot{s}(t) &= A_\mathrm{b}s(t) + A_\iota s(t) + (B_\mathrm{b} + B_\iota)\tau(t), \\ &= A_\mathrm{b}s(t) + B_\mathrm{b}(\tau(t) + B_\mathrm{b}^{-1}B_\iota\tau(t) + B_\mathrm{b}^{-1}A_\iota s(t)) \\ &= A_\mathrm{b}(t)s(t) + B_\mathrm{b}(\zeta(t)\tau(t) + \sigma(t)), \end{aligned} \tag{5.2}$$

where $\zeta(t) = I + B_\mathrm{b}^{-1}B_\iota$ and $\sigma(t) = B_\mathrm{b}^{-1}A_\iota s(t)$.

5.2.2 Torque Controller with $\mathcal{L}_1$ Augmentation

Baseline Torque Controller. The torque controller is the same as seen in Chapter 2, but represented using the baseline values of vehicle dynamics A_b and B_b

$$\tau_\mathrm{b}(t) = B_\mathrm{b}^{-1}[-A_\mathrm{b}s(t) + \dot{\alpha}(t) - Qe_2(t) + \Delta(t)e_1(t)]. \tag{5.3}$$

Closed-loop Error Dynamics. Based on the virtual controller and newly derived torque controller, we now derive the closed-loop error dynamics. In particular, the dynamics of $\dot{e}_1(t)$ are unchanged,

$$\dot{e}_1(t) = -S(\omega(t))e_1(t) - K\tanh(e_1(t)) - \Delta(t)e_2(t), \tag{5.4}$$

while the dynamics for $\dot{e}_2(t)$ can be rewritten as

$$\begin{aligned} \dot{e}_2(t) &= \dot{s}(t) - \dot{\alpha}(t) \\ &= A_\mathrm{b}s(t) + B_\mathrm{b}\tau(t) - \dot{\alpha}(t) \\ &= A_\mathrm{b}s(t) + B_\mathrm{b}\left(B_\mathrm{b}^{-1}[-A_\mathrm{b}s(t) + \dot{\alpha}(t) - Qe_2(t) + \Delta(t)e_1(t)]\right) - \dot{\alpha}(t) \\ &= -Qe_2(t) + \Delta(t)e_1(t). \end{aligned} \tag{5.5}$$

$\mathcal{L}_1$ Adaptive Control Augmentation. This portion of the torque controller is used to augment the baseline backstepping controller in the presence of uncertainty for the values of A and B. The total torque controller is defined as $\tau(t) = \tau_\mathrm{b}(t) + \tau_\mathrm{a}(t)$, where $\tau_\mathrm{a}(t)$ is defined

as the adaptive augmentation controller, and $\tau_{\rm b}(t)$ is defined as the baseline controller as shown in (5.3). The open-loop dynamics are represented as

$$\dot{s}(t) = A_{\rm b}s(t) + B_{\rm b}(\tau(t) + \sigma_1(t)). \tag{5.6}$$

where $\zeta(t) = I + \zeta_l(t)$ and $\sigma_1(t) = \zeta_l(t)\tau(t) + \sigma(t)$.

State Predictor. The state predictor is defined by following

$$\dot{\hat{s}}(t) = A_{\rm b}s(t) + B_{\rm b}(\tau(t) + \hat{\sigma}_1(t)) + A_{\rm sp}(\hat{s}(t) - s(t)), \tag{5.7}$$

where $A_{\rm sp}$ is a Hurwitx matrix, which defines the desired convergence properties of the prediction error dynamics.

Adaptation Law. If the sampling time of the controller is denoted at T_s, then the adaptive law is as follows

$$\hat{\sigma}_1(t) = B_{\rm b}^{-1}A_{\rm sp}(I - {\rm e}(A_{\rm sp}T_s))^{-1}{\rm e}(A_{\rm sp}T_s)e_3(iT_s), \tag{5.8}$$

where all i=1, 2, 3, ... and $e_3 = \hat{s}(t) - s(t)$.

$\mathcal{L}_1$ Adaptive Control Law and Error Dynamics. In order to derive the adaptive augmentation control law, we substitute the expression for $\tau_{\rm b}(t)$ into (5.6)

$$\begin{aligned}
\dot{s}(t) &= A_{\rm b}s(t) + B_{\rm b}(\tau_{\rm a}(t) + \tau_{\rm b}(t) + \sigma_1(t)) \\
&= A_{\rm b}s(t) + B_{\rm b}(B_{\rm b}^{-1}[-A_{\rm b}s(t) + \dot{\alpha}(t) - Qe_2(t) + \Delta(t)e_1(t)] \\
&\quad + B_{\rm b}(\tau_{\rm a}(t) + \sigma_1(t)) \\
&= A_{\rm b}s(t) - A_{\rm b}s(t) + \dot{\alpha}(t) - Qe_2(t) + \Delta(t)e_1(t) + B_{\rm b}(\tau_{\rm a}(t) + \sigma_1(t)).
\end{aligned} \tag{5.9}$$

By rewriting the error dynamics, we obtain

$$\dot{e_2}(t) = -Qe_2(t) + \Delta(t)e_1(t) + B_{\rm b}(\tau_{\rm a}(t) + \sigma_1(t) + B_{\rm b}^{-1}\Delta(t)e_1(t)), \tag{5.10}$$

$$\dot{e_2}(t) = -Qe_2(t) + \Delta(t)e_1(t) + B_{\rm b}(\mu(t)), \tag{5.11}$$

where $\mu(t) = (\tau_{\rm a}(t) + \hat{\sigma}_1(t) + B_{\rm b}^{-1}\Delta(t)e_1(t))$.

As the control objective of $\tau_{\rm a}(t)$ is to drive $e_2(t)$ to 0, the adaptive torque controller is chosen as

$$\tau_{\rm a}(s) = -K_1D(s)\mu(s), \tag{5.12}$$

where $D(s)$ is a strictly proper transfer function, and K_1 is a positive gain which yields a strictly proper stable $C(s)$, with the DC gain $C(0) = I$, defined as

$$C(s) = \zeta K_1D(s)(I + \zeta K_1D(s))^{-1}. \tag{5.13}$$

The stability of the $\mathcal{L}_1$ system can be ensured by the $\mathcal{L}_1$ norm stability condition similar to one in Section 3.1 in [18].

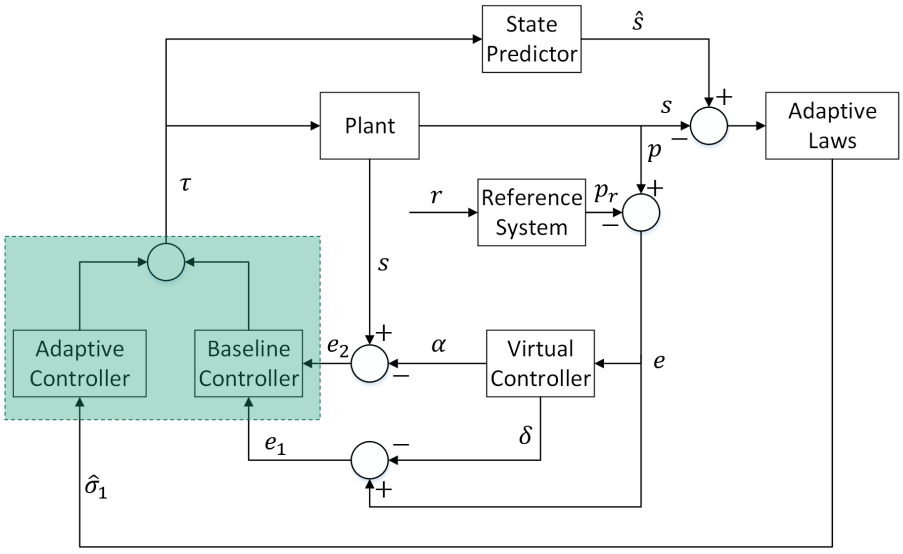

Figure 5.1: $\mathcal{L}_1$ Algorithm Block Diagram

5.3 Simulation Implementation

In this section, the effectiveness of the $\mathcal{L}_1$ adaptive backstepping controller is investigated through the use of computer simulation. This section includes the description of the simulation, results, and analysis. All simulations presented were carried out with either a trajectory dictated by a sine wave or a trajectory dictated by GPS waypoints as a reference system.

5.3.1 Computer Simulation and Results

Computer simulation using the software Simulink was used to test the controller. The details of the simulation for the $\mathcal{L}_1$ adaptive backstepping controller is presented via a block diagram shown in Figure 5.1. Each block of the diagram will be discussed individually.

(1) **Plant.** This block is the plant, which simulates the vehicle using the unicycle and dynamic model defined in Chapter 1. It takes in torque control commands as inputs and outputs the trajectory and the longitudinal and angular velocities of the vehicle.

(2) **Reference system.** This block is for the reference plant, based on the model we are using. It is either, the sine-wave dictated trajectory or the waypoint dictated trajectory. This block is responsible for simulating a reference trajectory that the vehicle should follow.

(3) **State Predictor.** This block is responsible for predicting the uncertain vehicle model. It takes in torque control commands as inputs and outputs the predicted longitudinal and angular velocities of the vehicle.

(4) **Virtual Controller.** This block is responsible for computing the desired longitudinal and angular velocities based on the error computed between the position of the vehicle and the position of the reference system.

(5) **Adaptive Law.** This block is responsible for updating the adaptive gains based on the error computed between the actual velocities and the predicted velocities from the state predictor.

(6) **Baseline Controller.** This block is responsible for computing the baseline torque commands for the motors of the vehicle using the error of the desired position and velocities with the actual position and velocities.

(7) **Adaptive Controller.** This block is responsible for handling model parameter uncertainties by computing the adaptive torque commands for the motors of the vehicle based on the adaptive gain $\hat{\sigma}_1(t)$.

Unsaturated Sine Wave Dictated Trajectory. For this category, the sine wave dictated trajectory is shown below in Figure 5.2 and the actual and desired velocities compared are shown in Figure 5.3. The following model parameters were used: $A^* = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$, $B^* = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $A_\text{b} = \left[\begin{smallmatrix} 6 & 0 \\ 0 & 6 \end{smallmatrix}\right]$, and $B_\text{b} = \left[\begin{smallmatrix} 1.5 & 0 \\ 0 & 1.5 \end{smallmatrix}\right]$. The following initialization values were used: $x_\text{r}(0) = 0$, $y_\text{r}(0) = 0$, $x(0) = -0.1$, $y(0) = 0$, $\theta(0) = 0$, $d(0) = 0.1$, $v(0) = 0$, $\omega(0) = 0$, $v_\text{d}(0) = 0$, and $\omega_\text{d}(0) = 0$. And finally, the following tuning parameters were used: $k_v = 1$, $k_w = 1$, $\alpha = 0.5$, $\beta = 0.1$, $\lambda = 1$, $T_s = 0.01$secs, $K_1 = 2$, $D(s) = \frac{1}{0.0001s^4+0.0036s^3+0.0993s^2+s}$, $A_{sp} = \left[\begin{smallmatrix} 40 & 0 \\ 0 & 40 \end{smallmatrix}\right]$, and $Q = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$.

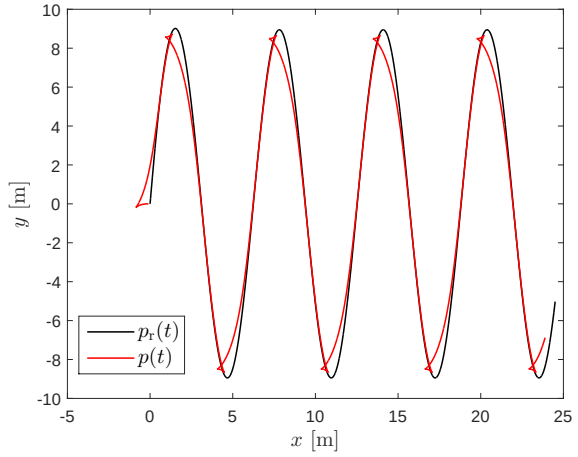

Figure 5.2: Sinusoidal trajectory without saturation

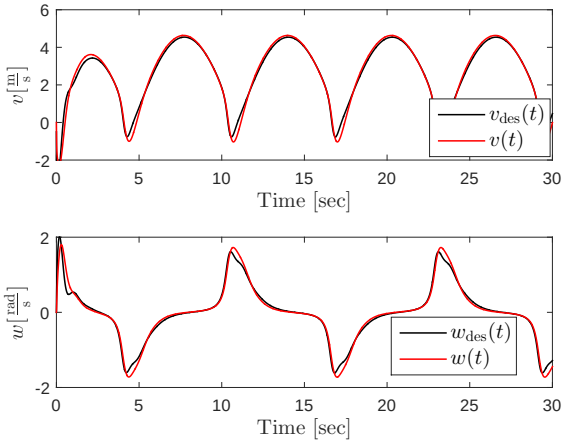

Figure 5.3: Velocity without saturation

Saturated Sine Wave Dictated Trajectory. For this category, the sine wave dictated trajectory is shown, with velocity saturation in Figure 5.4 and the actual and desired velocities are compared in Figure 5.5. The following model parameters were used: $A^* = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$, $B^* = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $A_{\mathrm{b}} = \left[\begin{smallmatrix} 6 & 0 \\ 0 & 6 \end{smallmatrix}\right]$, and $B_{\mathrm{b}} = \left[\begin{smallmatrix} 1.5 & 0 \\ 0 & 1.5 \end{smallmatrix}\right]$. The following initialization values were used: $x_{\mathrm{r}}(0) = 0$, $y_{\mathrm{r}}(0) = 0$, $x(0) = -0.1$, $y(0) = 0$, $\theta(0) = 0$, $d(0) = 0.1$, $v(0) = 0$, $\omega(0) = 0$, $v_{\mathrm{d}}(0) = 0$, and $\omega_{\mathrm{d}}(0) = 0$. The following tuning parameters were used: $k_v = 1$, $k_w = 1$, $\alpha = 0.5$, $\beta = 0.1$, $\lambda = 1$, $T_s = 0.01$secs, $K_1 = 2$, $D(s) = \frac{1}{0.0001s^4+0.0036s^3+0.0993s^2+s}$, $A_{sp} = \left[\begin{smallmatrix} 40 & 0 \\ 0 & 40 \end{smallmatrix}\right]$, and $Q = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$. Finally, the following tuning parameters for the velocity saturation were used: $d^* = 0.1$, $\dot{d}^* = 0$, $strip = 0.1$, $w_d = 2$, $\zeta_d = 0.85$, $L = 0.3556$m, $\gamma_{\max} = 25^o$, $v_{\max} = 10\frac{m}{s}$, and $v_{\min} = 1\frac{m}{s}$.

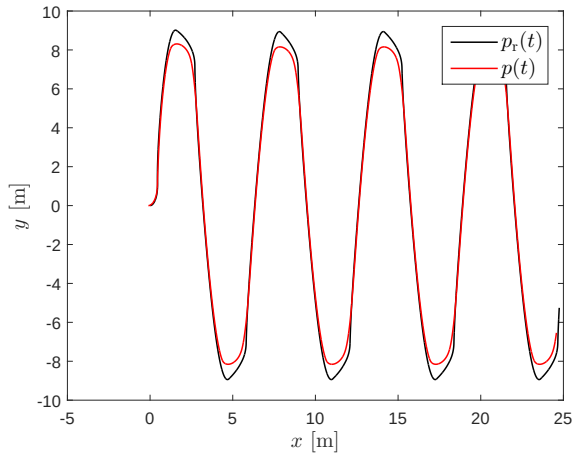

Figure 5.4: Sinusoidal trajectory with saturation

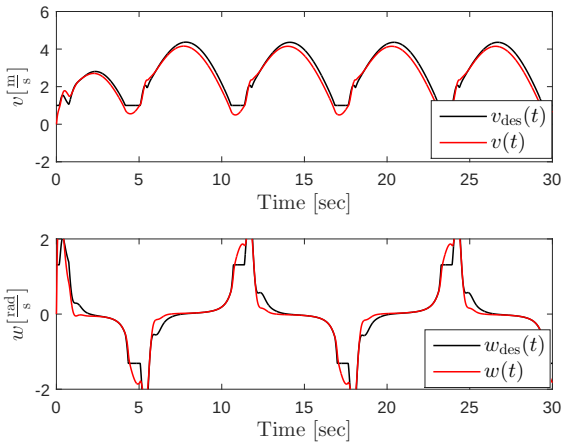

Figure 5.5: Velocity with saturation

Unsaturated Waypoint Dictated Trajectory. For this category, the way-point dictated trajectory is shown below in Figure 5.6 and the actual and desired velocities compared are shown in Figure 5.7. The following model parameters were used: $A^* = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$, $B^* = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $A_\mathrm{b} = \left[\begin{smallmatrix} 6 & 0 \\ 0 & 6 \end{smallmatrix}\right]$, and $B_\mathrm{b} = \left[\begin{smallmatrix} 1.5 & 0 \\ 0 & 1.5 \end{smallmatrix}\right]$. The following initialization values were used: $x_\mathrm{r}(0) = 0$, $y_\mathrm{r}(0) = 0$, $x(0) = 4$, $y(0) = 5$, $\theta(0) = 0$, $d(0) = 0.1$, $v(0) = 0$, $\omega(0) = 0$, $v_\mathrm{d}(0) = 0$, and $\omega_\mathrm{d}(0) = 0$. The following tuning parameters were used: $k_v = 1$, $k_w = 1$, $\alpha = 0.5$, $\beta = 0.1$, $\lambda = 1$, $T_s = 0.01$secs, $K_1 = 2$, $D(s) = \frac{1}{0.0001s^4+0.0036s^3+0.0993s^2+s}$, $A_{sp} = \left[\begin{smallmatrix} 40 & 0 \\ 0 & 40 \end{smallmatrix}\right]$, and $Q = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$. Finally, the following tuning parameters for the potential field algorithm were used: $F_\mathrm{ac} = 10$, $F_\mathrm{rc} = 8$, $W = 2.5$, $n = 2$, and $E = 8$.

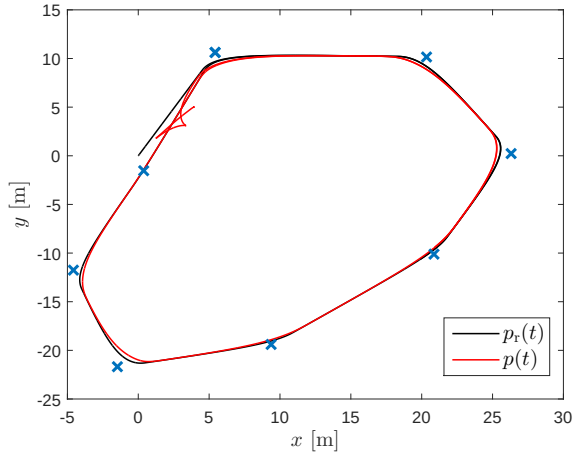

Figure 5.6: Waypoint trajectory without saturation

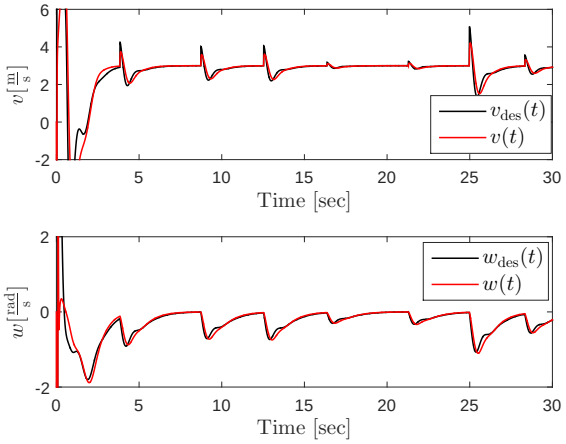

Figure 5.7: Velocity without saturation

Saturated Waypoint Dictated Trajectory. For this category, the way-point dictated trajectory is shown, with velocity saturation in Figure 5.8 and the actual and desired velocities are compared in Figure 5.9. The following model parameters were used: $A^* = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$, $B^* = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $A_{\rm b} = \left[\begin{smallmatrix} 6 & 0 \\ 0 & 6 \end{smallmatrix}\right]$, and $B_{\rm b} = \left[\begin{smallmatrix} 1.5 & 0 \\ 0 & 1.5 \end{smallmatrix}\right]$. The following initialization values were used: $x_{\rm r}(0) = 0$, $y_{\rm r}(0) = 0$, $x(0) = 4$, $y(0) = 5$, $\theta(0) = 0$, $d(0) = 0.1$, $v(0) = 0$, $\omega(0) = 0$, $v_{\rm d}(0) = 0$, and $\omega_{\rm d}(0) = 0$. The following tuning parameters were used: $k_v = 1$, $k_w = 1$, $\alpha = 0.5$, $\beta = 0.1$, $\lambda = 1$, $T_s = 0.01$secs, $K_1 = 2$, $D(s) = \frac{1}{0.0001s^4+0.0036s^3+0.0993s^2+s}$, $A_{sp} = \left[\begin{smallmatrix} 40 & 0 \\ 0 & 40 \end{smallmatrix}\right]$, and $Q = \left[\begin{smallmatrix} 5 & 0 \\ 0 & 5 \end{smallmatrix}\right]$. The following tuning parameters for the velocity saturation were used: $d^* = 0.1$, $\dot{d}^* = 0$, $strip = 0.1$, $w_d = 2$, $\zeta_d = 0.85$, $L = 0.3556$m, $\gamma_{\max} = 25^o$, $v_{\max} = 10\frac{m}{s}$, and $v_{\min} = 1\frac{m}{s}$. Finally, the following tuning parameters for the potential field algorithm were used: $F_{\rm ac} = 10$, $F_{\rm rc} = 8$, $W = 2.5$, $n = 2$, and $E = 8$.

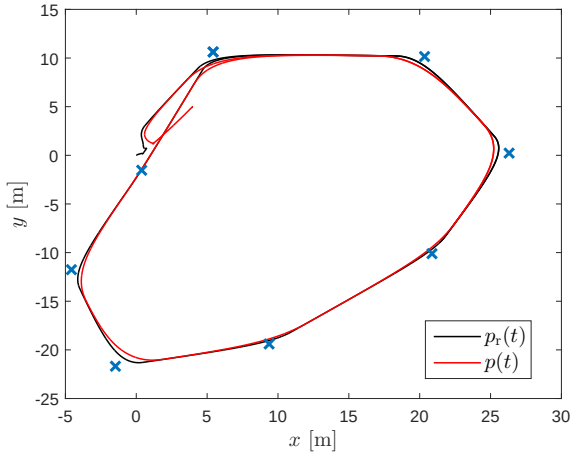

Figure 5.8: Waypoint trajectory with saturation

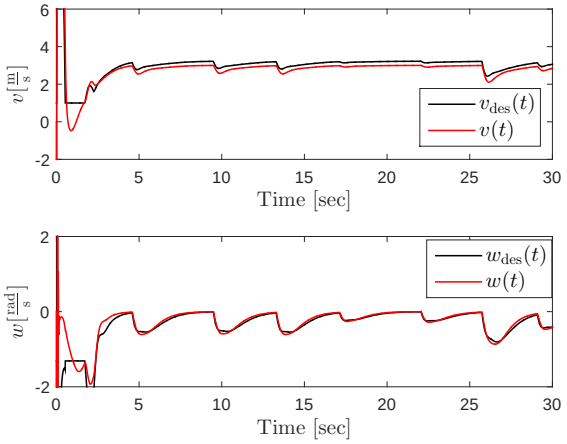

Figure 5.9: Velocity with saturation

Analysis of Results. As can be seen in Figures 5.2 and 5.6, the trajectories for the vehicle in the sine dictated and waypoint dictated reference without velocity saturation are similar to the behavior of the responses in [4] and in Chapter 4 as expected. The velocity profiles in Figures 5.3 and 5.7 are also similar to previous results.

With the addition of velocity saturation, the trajectories are similar with the saturated trajectories in [4] and in Chapter 4, avoiding all negative velocities as shown in Figures 5.4 and 5.8. By comparing the velocity profiles in Chapter 4 and Figures 5.5 and 5.9, it can be seen that the $\mathcal{L}_1$ adaptive backstepping controller enables the velocity profiles to be smoother with no sudden spikes. The spikes in the MRAC backstepping velocity profile is a result of the velocity saturation implemented on the predator-prey model.

Therefore, it can be concluded that the $\mathcal{L}_1$ adaptive backstepping controller could be a feasible alternative to the direct MRAC backstepping and could potentially allow for more robust experimental solution.

5.4 Summary for Controller

The table below summarize the $\mathcal{L}_1$ Adaptive Backstepping Controller.

Table 5.1: Summary of the $\mathcal{L}_1$ Adaptive Backstepping Controller

System	$\dot{p}(t) = \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} \cos(\theta(t)) & 0 \\ \sin(\theta(t)) & 0 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}$ $\dot{\theta}(t) = \omega(t)$ $\dot{s} = As(t) + B\tau(t)$
State Predictor	$\dot{\hat{s}}(t) = A_{\mathrm{b}} s(t) + B_{\mathrm{b}}(\tau(t) + \hat{\sigma}_1(t)) + A_{\mathrm{sp}}(\hat{s}(t) - s(t))$
Virtual Controller	$\alpha(t) = \begin{bmatrix} v_{\mathrm{d}}(t) \\ \omega_{\mathrm{d}}(t) \end{bmatrix} = \Delta^{-1}(t)(K\tanh(e(t) - \delta(t)) + R^{\mathrm{T}}(\theta(t))\dot{p}_{\mathrm{r}}(t) - \dot{\delta}(t))$
Baseline Controller	$\tau_{\mathrm{b}}(t) = B_{\mathrm{b}}^{-1}[-A_{\mathrm{b}} s(t) + \dot{\alpha}(t) - Qe_2(t) + \Delta(t)e_1(t)]$
Adaptive Controller	$\tau_{\mathrm{a}}(s) = -K_1 D(s)\mu(s)$ $\mu(t) = (\tau_{\mathrm{a}}(t) + \hat{\sigma}_1(t) + B_{\mathrm{b}}^{-1}\Delta(t)e_1(t))$
Adaptive Law	$\hat{\sigma}_1(t) = B_{\mathrm{b}}^{-1} A_{\mathrm{sp}}(I - \mathrm{e}(A_{\mathrm{sp}}T_s))^{-1}\mathrm{e}(A_{\mathrm{sp}}T_s)e_3(iT_s)$

Chapter 6

Conclusions and Future Research

In [4], the authors presented a predator-prey model for obstacle avoidance and tracking. This model allows the vehicle to perform in unstructured driving environments and track its virtual target in a smooth path even if the target is traveling in a chaotic fashion. However, there is a disconnect between the desired velocities requested by the predator-prey model and the torque signals provided to the vehicle. The vehicle's velocity-dependent dynamics were not taken into account, therefore causing the vehicle to derail from the ideal trajectory at sharp turns. We developed an adaptive backstepping architecture to overcome this shortcoming, allowing the vehicle to track its planned trajectory more accurately. The development will also allow for the predator-prey model to be cross-platform compatible.

In Chapters 3, 4 and 5, newly developed indirect MRAC backstepping, direct MRAC backstepping, and $\mathcal{L}_1$ adaptive backstepping controllers are presented. For our vehicle, it is shown that the indirect MRAC backstepping is not suitable. To overcome this difficulty, the direct MRAC backstepping is developed. Through the use of simulation and experimental results, it is shown that the direct MRAC backstepping controller with velocity and acceleration saturation, allows the vehicle to handle varying dynamics ideally and is suitable for a velocity range of $1\frac{m}{s}$ to $9\frac{m}{s}$ without any parameter tuning required for different trajectories, curvatures and velocities. Hence, concluding that the direct MRAC backstepping controller is able to fix the disconnect between the desired velocities and the torque provided to the motors. The result can be seen in Figure 4.18 and 4.25, where the vehicle is able to adapt and follow the trajectory in the desired manner even in the presence of sharp turns.

To investigate an alternative method to the direct MRAC backstepping, we developed a $\mathcal{L}_1$ adaptive backstepping controller to allow for faster adaptation than that of direct MRAC backstepping. Through simulation, we concluded that it is a feasible controller to investigate experimentally. In future endeavors, the work presented here could be extended in the following ways: First, the $\mathcal{L}_1$ controller will be implemented on the vehicle. Second, the $\mathcal{L}_1$ controller will be analyzed and compared with respect to the MRAC Backstepping controller. Finally, the controllers would be implemented on different vehicles to investigate

robustness.

Bibliography

[1] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouveliere, "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, pp. 589–606, Sept 2010.

[2] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, pp. 1599–1616, Dec 2012.

[3] V. Medina-Garciadiego and A. Leonessa, "Tracking control of a nonholonomic ground vehicle," in *American Control Conference (ACC), 2011*, pp. 1710–1713, June 2011.

[4] A. Shoemaker and A. Leonessa, "Bioinspired tracking control of high speed nonholonomic ground vehicles," *Journal of Robotics*, 2015.

[5] G. Indiveri, "Kinematic time-invariant control of a 2d nonholonomic vehicle," in *Decision and Control, 1999. Proceedings of the 38th IEEE Conference on*, vol. 3, pp. 2112–2117 vol.3, 1999.

[6] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ: Prentice Hall, third ed., 2002.

[7] M. Krstić, I. Kanellakopoullos, and P. Kokotović, *Nonlinear and Adaptive Control Design*. New York, NY: John Wiley and Sons, Inc, 1995.

[8] N. Fischer, R. Kamalapurkar, and W. E. Dixon, "Lasalle-yoshizawa corollaries for non-smooth systems," *IEEE Transactions on Automatic Control*, vol. 9, no. 58, pp. 2333–2338, 2013.

[9] J. Borenstein and Y. Koren, "High-speed obstacle avoidance for mobile robots," in *Intelligent Control, 1988. Proceedings., IEEE International Symposium on*, pp. 382–384, IEEE, 1988.

[10] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 19, no. 5, pp. 1179–1187, 1989.

[11] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pp. 1398–1404, IEEE, 1991.

[12] D. Lawrence, W. Sethares, and W. Ren, "Parameter drift instability in adaptive systems," in *Decision and Control, 1990., Proceedings of the 29th IEEE Conference on*, pp. 3230–3235 vol.6, Dec 1990.

[13] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Prentice-Hall, Upper Saddle River, NJ, 1996.

[14] J.-B. Pomet and L. Praly, "Adaptive nonlinear regulation: estimation from the lyapunov equation," *Automatic Control, IEEE Transactions on*, vol. 37, pp. 729–740, Jun 1992.

[15] S. Mallikarjunan, B. Nesbitt, E. Kharisov, E. Xargay, N. Hovakimyan, C. Cao, *et al.*, "L1 adaptive controller for attitude control of multirotors," 2012.

[16] E. Kharisov, N. Hovakimyan, and K. J. Astrom, "Comparison of several adaptive controllers according to their robustness metrics," *AIAA Paper No. AIAA-2010-8047*, 2010.

[17] C. Cao and N. Hovakimyan, "Design and analysis of a novel adaptive control architecture with guaranteed transient performance," *Automatic Control, IEEE Transactions on*, vol. 53, no. 2, pp. 586–591, 2008.

[18] N. Hovakimyan and C. Cao, *L1 adaptive control theory: guaranteed robustness with fast adaptation*, vol. 21. Siam, 2010.