# ROBUST CONTROL AND MOTION PLANNING FOR NONLINEAR UNDERACTUATED SYSTEMS USING $H^\infty$ TECHNIQUES

Gregory James Toussaint, Ph.D.
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign, 2000
Tamer Başar, Advisor

This thesis presents new techniques for planning and robustly controlling the motion of nonlinear underactuated vehicles when disturbances are present and only imperfect state measurements are available for feedback. The basic motion planning algorithm uses cubic splines or Pythagorean hodograph curves to connect the initial and final configurations and to generate a feasible trajectory for the system. The feasible trajectory and its control inputs are improved through an iterative $H^\infty$-filter. Techniques are demonstrated for generalizing the motion planning algorithm to address the obstacle avoidance, multiple vehicle, and minimum distance planning problems. To track a desired trajectory, first a state feedback control law is developed for the linearized system using an $H^\infty$-optimal design. The state feedback controller produces a locally exponentially stable closed-loop system and guarantees a precomputable level of disturbance attenuation for the system. Subsequently, an imperfect state measurement feedback controller is developed by combining a state estimate with the state feedback control law. The state estimator exploits a unique structure in the nonlinear equations of motion to decompose the system into two interlaced subsystems, which leads to a direct solution. The estimator is actually an $H^\infty$-filter, and under it the controller achieves a modified form of disturbance attenuation. Simulations included in the thesis illustrate the motion planning, the state feedback control, and the imperfect state measurement control algorithms derived, for selected models of nonlinear underactuated vehicles.

# ROBUST CONTROL AND MOTION PLANNING FOR NONLINEAR UNDERACTUATED SYSTEMS USING $H^\infty$ TECHNIQUES

BY

GREGORY JAMES TOUSSAINT

B.S., Cornell University, 1989
M.S., Air Force Institute of Technology, 1992

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2000

Urbana, Illinois

# ABSTRACT

This thesis presents new techniques for planning and robustly controlling the motion of nonlinear underactuated vehicles when disturbances are present and only imperfect state measurements are available for feedback. The basic motion planning algorithm uses cubic splines or Pythagorean hodograph curves to connect the initial and final configurations and to generate a feasible trajectory for the system. The feasible trajectory and its control inputs are improved through an iterative $H^\infty$-filter. Techniques are demonstrated for generalizing the motion planning algorithm to address the obstacle avoidance, multiple vehicle, and minimum distance planning problems. To track a desired trajectory, first a state feedback control law is developed for the linearized system using an $H^\infty$-optimal design. The state feedback controller produces a locally exponentially stable closed-loop system and guarantees a precomputable level of disturbance attenuation for the system. Subsequently, an imperfect state measurement feedback controller is developed by combining a state estimate with the state feedback control law. The state estimator exploits a unique structure in the nonlinear equations of motion to decompose the system into two interlaced subsystems, which leads to a direct solution. The estimator is actually an $H^\infty$-filter, and under it the controller achieves a modified form of disturbance attenuation. Simulations included in the thesis illustrate the motion planning, the state feedback control, and the imperfect state measurement control algorithms derived, for selected models of nonlinear underactuated vehicles.

# ACKNOWLEDGMENTS

Many people contributed to the results presented in this dissertation and I am sincerely thankful for their help. Foremost among them is my advisor, Professor Tamer Başar, who gave me the opportunity to study at the University of Illinois at Urbana-Champaign and patiently guided me through every aspect of the research process. I benefited greatly from the time Professor Francesco Bullo spent with me as a secondary advisor, and he helped resolve many of the research challenges we encountered. I also appreciate the contributions of Professors Sean Meyn, Seth Hutchinson, and Victoria Coverstone-Carroll, who served on my doctoral committee and positively influenced the direction of this research. I was fortunate to have shared an office and many productive discussions with Jim Melody, who helped me with course work and preparation for the qualifying examination, as well as with computer and software support over the past three years. In addition, I appreciate the work done by Timur Karatas, who skillfully programmed some of the software for parts of the motion planning simulations. I would also like to thank Colonel Alan R. Klayton, the Electrical Engineering Department Head at the United States Air Force Academy, who made it possible for me to return to school, and Lieutenant Colonel Donna E. Peterson, who encouraged me to pursue both my academic and professional goals. My final thanks are reserved for my wife, Tricia, who was my true companion during this research journey.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

**Figure** **Page**

# CHAPTER 1

# INTRODUCTION

The primary objective of this thesis research has been to undertake a comprehensive study of motion planning and tracking control issues that arise in the context of nonlinear underactuated vehicles. Motion planning involves the development of a feasible trajectory by fully accounting for the underactuated structure of the vehicle. Tracking control involves the design of controllers that use noisy partial state measurements to generate inputs that make the vehicle track the feasible trajectory. Usually, underactuated vehicles require manual planning and a skilled operator to perform even basic maneuvers. Automating these tasks will allow us to sever the link between the vehicle and the operator, which can lead to improved performance and expand the range of applications for the vehicle. Before we can cut this vehicle-operator link, however, we must overcome the significant design challenges that arise when we merge nonlinearities and underactuation into a single problem. The well established techniques for planning motions and controlling linear fully actuated systems are inadequate for nonlinear underactuated vehicles. Solving this design problem requires us to fully understand the limits of the current techniques and then advance the theory to develop practical solutions. This study has addressed this challenge, and obtained novel motion planning algorithms and feedback tracking control laws using the framework and tools of $H^\infty$ control and filtering. When combined, these new algorithms allow us to guide the underactuated vehicle between two configurations during a specified time interval. The $H^\infty$ techniques

provide a way to directly calculate practical solutions that are robust to the disturbances the vehicle will likely encounter.

This first chapter will present a detailed description of the basic problem we will be addressing and provide motivation for considering this area of study. We will also describe the original contributions we have made with this research. The chapter concludes with a description of the model of an underactuated vehicle that we will use to demonstrate the new motion planning and control techniques.

## 1.1 Problem Description

The vehicles of interest in this project display four attributes that make the motion planning and control problems challenging. The attributes are that (i) the vehicles are underactuated, (ii) the equations of motion are nonlinear, (iii) there are unknown disturbances entering the equations of motion, and (iv) the state of the system is only available through noisy partial state measurements. Individually, these factors can be difficult to handle, so their combination in a single design problem presents a formidable task. Each of these design challenges merits a brief explanation.

An *underactuated system* is defined to be one where the dimension of the space spanned by the control vector is less than the dimension of the configuration space [1]. Simply stated, this refers to a mechanical system that has fewer control inputs than degrees of freedom. An equivalent characterization of an underactuated system is that it has nonintegrable acceleration relations or dynamics [2]. Another common description for an underactuated system is that it has second-order nonholonomic constraints. First-order nonholonomic constraints refer to restrictions on the velocities of the system, while second-order nonholonomic constraints restrict the accelerations of the system. The example underactuated vehicle we will focus on is a ship moving on a planar surface that

has a forward acceleration input from the propeller thrust and a pure angular acceleration input from the rudder. There is no lateral acceleration input for the ship, which makes the sideways motion the underactuated dimension. Other examples of underactuated vehicles include spacecraft, battle-damaged aircraft, hovercraft, and missiles.

In addition to being underactuated, the vehicles studied in this thesis will have nonlinear equations of motion. Nonlinear systems present their own set of design challenges if we want to plan a set of motions and control the vehicle to follow a specific trajectory. The standard tools used to control nonlinear systems, including feedback linearization and integrator backstepping, do not provide direct solutions for the underactuated systems we will consider, so new techniques must be developed. The new techniques to be presented in this thesis involve $H^\infty$-optimal control design methods and will exploit a unique decomposition of the equations of motion that greatly simplifies the analysis.

The third factor in the problem that contributes to the design challenge is the presence of disturbances in the equations of motion. For the control design we will adopt a game-theoretic approach to solve the $H^\infty$-optimal control problem. Accordingly, we will consider the disturbances as an input to the system selected by an adversary who is trying to disrupt the motion of vehicle as much as possible. Our objective will be to develop a feedback control law that minimizes the maximum negative effect the adversary can have. The result will be a control law that attenuates the effect of the disturbances on the motion of the vehicle and allows it to track a desired trajectory. This minimax approach to control design has documented solutions for both linear and nonlinear systems. The solution for nonlinear systems is expressed in terms of a partial differential equation, which, in general, does not admit a closed-form solution. To develop a closed-form solution, we will linearize the vehicle's equations of motion about a desired trajectory to find the optimal control inputs.

The final design challenge in this thesis is the imperfect state measurement restriction on the control law. In this case, imperfect state measurement means that we can measure only some of the states in the system and there are unknown disturbances entering the measurements. The solution to the $H^{\infty}$-optimal control problem for the imperfect state measurement case is well known and can be computed directly for linear systems. For nonlinear systems, the solution involves two partial differential equations, which usually cannot be solved explicitly. Our approach will use the structure of the underactuated vehicle model to reduce the nonlinear problem to a pair of simpler subproblems. We will demonstrate that we can build on the established results for linear systems to achieve a form of disturbance attenuation for the nonlinear underactuated vehicle.

With the four primary elements of the system in place, we can now make a formal statement of the problems this research effort will address. We assume we are given an accurate, time-invariant model of an underactuated vehicle that we want to control. In addition, we are given the initial and final configurations between which we want to move the vehicle, and a time interval for completing the task. Our first objective is to develop a motion planning algorithm to design a feasible trajectory for the underactuated vehicle to move from one configuration to another during the specified time interval. The outcome of the motion planning algorithm will be a set of time-varying control inputs that can be applied to a model of the underactuated vehicle to generate a desired trajectory. The planning algorithm will focus on finding a direct method for computing the control inputs as opposed to an exhaustive search technique. The second objective is to create a controller design technique that causes an underactuated vehicle to track the desired trajectory while also attenuating the effect of disturbances. The control law will be restricted to use only imperfect state measurements. The control design problem can be subdivided into two parts. The first part will be to design a state feedback controller for the nonlinear underactuated system assuming perfect state information is available

to the controller. The second part will be to modify the state feedback controller to account for the imperfect state measurement restriction. For both the motion planning and the control aspects of the problem, we want to exploit the known dynamics of the underactuated vehicle to develop natural motions for the system to achieve the objectives.

We note that the motion planning and control aspects of the problem can be completely decoupled and addressed separately, but we will attempt to plan motions that make the tracking control problem easier to solve. We also observe that the motion planning algorithm yields a feasible desired trajectory that the underactuated vehicle could track if it had the proper initial conditions and there were no disturbances present. The control law developed here is valuable because it will recover from initial condition errors and successfully track the desired trajectory even in the presence of disturbances.

To formulate a meaningful and realistic thesis research project, there should be limitations on the scope of the research to make the problem manageable. We used the following limitations as guidelines to provide a structure for this research. To keep the research focused, we did not address three areas of control theory that may be relevant to the problem. First, we did not use adaptive methods as a primary technique for vehicle motion control. Adaptive control offers advantages when the true nature of the system is unknown or is changing over time and system parameters need to be identified. Our research assumed we had a well-defined vehicle moving with known properties. The second area of research we excluded was biologically inspired controller design, which includes neural networks, fuzzy logic, and genetic algorithms. These methods offer valid controller design techniques which are suitable for some nonlinear systems, but we avoided these approaches because they do not always offer good insight into the nature of the controller and make it difficult to precisely describe the contributions to control theory. Finally, the research did not focus on the stochastic nature of the problem. There are certainly many aspects of vehicle motion control that could be characterized in probabilistic terms,

but we will not explore this option. These three approaches may offer good results in practice, so they may be viable directions for future research projects. These restrictions to the scope of the project were relatively mild and still offered ample room for significant contributions to motion planning and control for underactuated systems.

This section has outlined the four challenging aspects of the underactuated vehicle control problem and identified some research guidelines to keep the effort manageable. We will now explain why this area of research is important.

## 1.2 Motivation

There are two practical reasons for developing techniques to plan motions for and to control underactuated systems. First, a fully actuated system requires more control inputs than an underactuated system, which means there will have to be more devices to generate the necessary forces. The additional controlling devices add to the cost and weight of the system. Finding a way to control an underactuated version of the system would eliminate some of the controlling devices and could improve the overall performance or reduce the cost. The second practical reason for studying underactuated vehicles is that underactuation provides a backup control technique for a fully actuated system. If a fully actuated system is damaged and we have an underactuated controller available, then we may be able to recover gracefully from the failure. The underactuated controller may be able to salvage a system that would otherwise be uncontrollable. This reason for designing controllers for underactuated systems would be especially useful for aircraft or spacecraft, where actuator failures can be catastrophic to the vehicle or its mission. Section 6.6 will describe in more detail how techniques for underactuated systems can assist with failure recovery for fully actuated systems.

## 1.3 Contributions

This research effort has developed techniques for underactuated vehicles and makes contributions to both the motion planning and the control disciplines. We will highlight the key contributions presented in the following chapters.

There are three significant results from the motion planning research. First, we developed a technique to directly plan motions for planar underactuated vehicles using polynomial curves and numerical methods. The polynomial curves can be cubic splines, Pythagorean hodograph curves, or other curves that can be quickly generated to link two configurations. We then used the underactuated vehicle model to extract a complete feasible trajectory from the simple polynomial curve. Section 3.2 provides more details on this contribution. The second major motion planning result was to demonstrate how an iterative $H^\infty$-filter could be used to improve the feasible trajectory derived from the polynomial curve. The filter updates the trajectory and moves the final configuration closer to the goal configuration to improve the performance of the motion planner; see Subsection 3.2.4 for a complete explanation of the iterative $H^\infty$-filter. The final motion planning result was to show how to incorporate elements from the basic motion planning approach with established motion planning algorithms to solve more challenging planning problems for an underactuated vehicle. Specifically, we identified the path length of the polynomial curve generated in the basic planning algorithm as a valuable part of a metric used in a randomized planning technique. We explain the randomized planning technique, how to improve the metric and how to use the new approach to address the obstacle avoidance and multiple vehicle planning problems in Section 3.3.

We would also like to highlight two primary contributions from the controller design aspects of the problem. The first contribution was the development of a linearized perfect state feedback $H^\infty$ controller for an underactuated system. We proved that the controller

provides local exponential stability for the closed-loop system and offers a solution to the tracking control problem. In addition, our research characterized the disturbance attenuation properties and the region of attraction for the closed-loop system and demonstrated that the technique works for both minimum and nonminimum phase systems. Chapter 4 provides the complete results for this contribution. The other major result from the controller design research was the development of an $H^{\infty}$-filter based on a unique decomposition of the underactuated model and used to estimate the states of the system. The filter design is successful because we can decompose the full nonlinear system into two subsystems, with the property that each is affine if the state of the other subsystem is known. This structure allowed us to develop an approach for estimating the complete state from an imperfect partial state measurement. We then combined the state estimate with the state feedback controller to develop an imperfect state measurement controller for the underactuated system. We were also able to prove a form of disturbance attenuation for the closed-loop system using the imperfect state measurement controller. We present the details behind this second contribution in control design in Chapter 5.

Our research has made some additional contributions in other directions that we will describe as we present the results in Chapters 3, 4, and 5. We highlighted the above five contributions because they stand out as the most significant results of this research effort.

## 1.4 Underactuated Vehicle Model

We present in this section the vehicle model used for this research effort and describe some of its characteristics. The model was used earlier by Pettersen and Nijmeijer [3] and we have made only minor changes to simplify the notation. For a more detailed description of this and similar models, see [4]. The relevant equations of motion for this

system are

$$\dot{u} = m_u\, v\, r - d_u\, u + u_1 + w_1 \tag{1.1}$$

$$\dot{v} = m_v\, u\, r - d_v\, v + w_2 \tag{1.2}$$

$$\dot{r} = m_r\, u\, v - d_r\, r + u_2 + w_3 \tag{1.3}$$

$$\dot{x} = u\, \cos(\psi) - v\, \sin(\psi) \tag{1.4}$$

$$\dot{y} = u\, \sin(\psi) + v\, \cos(\psi) \tag{1.5}$$

$$\dot{\psi} = r \tag{1.6}$$

where $u$, $v$, and $r$ are the body-frame velocities in surge, sway, and yaw, respectively; $x$ and $y$ are the inertial positions; and $\psi$ is the inertial rotation angle. The coefficients $m_i$ and $d_i$ represent combined mass terms, including added mass, and damping coefficients. The two control inputs are $u_1$ and $u_2$. The disturbances $w_i$ only appear in Equations (1.1) through (1.3) because they represent forces or torques that influence the acceleration of the vehicle. Equations (1.4) through (1.6) are just the kinematics of the system relating the inertial reference frame to the body frame, and they do not have any disturbances.

One of our research objectives is to design the control inputs $u_1$ and $u_2$ such that the vehicle tracks a feasible trajectory. A feasible trajectory can be generated by simulating the ship model described above without the disturbances. The following equations describe the desired system, the state of which constitutes a feasible trajectory. In these equations, the subscript $d$ indicates the desired values for the states and control inputs

$$\dot{u}_d = m_u\, v_d\, r_d - d_u\, u_d + u_{1d} \tag{1.7}$$

$$\dot{v}_d = m_v\, u_d\, r_d - d_v\, v_d \tag{1.8}$$

$$\dot{r}_d = m_r\, u_d\, v_d - d_r\, r_d + u_{2d} \tag{1.9}$$

$$\dot{x}_d = u_d \cos(\psi_d) - v_d \sin(\psi_d) \tag{1.10}$$

$$\dot{y}_d = u_d \sin(\psi_d) + v_d \cos(\psi_d) \tag{1.11}$$

$$\dot{\psi}_d = r_d \tag{1.12}$$

where we know the initial conditions of the states and the value of $u_{1d}$ and $u_{2d}$ for all time. We note that the motion planning algorithm will generate the control inputs $u_{1d}$ and $u_{2d}$ so that the feasible trajectory approaches the specified final configuration.

An equivalent way to state the tracking problem is to require the difference between the actual vehicle configuration and the desired configuration to approach zero. Denote the difference between the two configurations as the error. Since the vehicle will not necessarily share the same initial conditions as the desired system, the tracking controller will drive the error to zero and minimize the effect of the disturbances as the vehicle converges to the reference trajectory. To help in the analysis, we write the error differential equations by subtracting the desired equations from the system equations and defining the error variables as $a_e = a - a_d$, where $a$ represents any state or control variable, to get

$$\dot{u}_e = m_u(v_e\, r_d + v_d\, r_e + v_e\, r_e) - d_u\, u_e + u_{1e} + w_1 \tag{1.13}$$

$$\dot{v}_e = m_v(u_e\, r_d + u_d\, r_e + u_e\, r_e) - d_v\, v_e + w_2 \tag{1.14}$$

$$\dot{r}_e = m_r(u_e\, v_d + u_d\, v_e + u_e\, v_e) - d_r\, r_e + u_{2e} + w_3 \tag{1.15}$$

$$\begin{aligned} \dot{x}_e = {} & u_d[\cos(\psi_d + \psi_e) - \cos(\psi_d)] + u_e \cos(\psi_d + \psi_e) \\ & - v_d[\sin(\psi_d + \psi_e) - \sin(\psi_d)] - v_e \sin(\psi_d + \psi_e) \end{aligned} \tag{1.16}$$

$$\begin{aligned} \dot{y}_e = {} & u_d[\sin(\psi_d + \psi_e) - \sin(\psi_d)] + u_e \sin(\psi_d + \psi_e) \\ & + v_d[\cos(\psi_d + \psi_e) - \cos(\psi_d)] + v_e \cos(\psi_d + \psi_e) \end{aligned} \tag{1.17}$$

$$\dot{\psi}_e = r_e. \tag{1.18}$$

We will find it helpful to use an alternative, but equivalent, description for the kinematic equations (1.4) through (1.6) for parts of the analysis. The second description uses

variables $z_1$, $z_2$, and $z_3$ instead of $x$, $y$, and $\psi$, and the transformation is

$$z_1 = x\cos(\psi) + y\sin(\psi) \tag{1.19}$$

$$z_2 = -x\sin(\psi) + y\cos(\psi) \tag{1.20}$$

$$z_3 = \psi. \tag{1.21}$$

This transformation yields the following three kinematic equations after taking time derivatives and simplifying:

$$\dot{z}_1 = u + z_2 r \tag{1.22}$$

$$\dot{z}_2 = v - z_1 r \tag{1.23}$$

$$\dot{z}_3 = r. \tag{1.24}$$

We note that the transformation from $[x,\ y,\ \psi]$ to $[z_1,\ z_2,\ z_3]$ is invertible, and hence given either description, we can recover the other one.

To help ease notation, it will be convenient to have a simple expression to describe each of the different types of equations of motion. To represent the actual system with the $x$, $y$, and $\psi$ variables in (1.1) through (1.6) we use

$$\dot{\mathbf{q}} = f(t, \mathbf{q}) + B(t)\mathbf{u} + D(t)\mathbf{w} \tag{1.25}$$

where $f(t, \mathbf{q})$ is a vector-valued nonlinear function and the boldface variables represent vectors. Specifically, we have

$$\mathbf{q} = \begin{bmatrix} u & v & r & x & y & \psi \end{bmatrix}^{\mathrm{T}}$$

$$\mathbf{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^{\mathrm{T}}$$

$$\mathbf{w} = \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix}^{\mathrm{T}}.$$

We also have similar expressions for the desired system and the error equations, given by

$$\dot{\mathbf{q}}_d = f(t, \mathbf{q}_d) + B(t)\mathbf{u}_d \tag{1.26}$$

$$\dot{\mathbf{q}}_e = f_e(t, \mathbf{q}_e) + B(t)\mathbf{u}_e + D(t)\mathbf{w}. \tag{1.27}$$

Note that the desired system equation given by (1.26) uses the same nonlinear function as (1.25), but the error equation (1.27) will have, in general, a different nonlinear function. When we use the $z_i$ variables in the system equations, we will include a $z$ subscript to arrive at the following compact notation:

$$\dot{\mathbf{q}}_z = f_z(t, \mathbf{q}_z) + B(t)\mathbf{u}_z + D(t)\mathbf{w}. \tag{1.28}$$

We will introduce additional notation as required when we analyze the problem.

The vehicle model we presented allows us to incorporate into the problem the four design challenges described in Section 1.1. The system is underactuated because there are only two control inputs and three degrees of freedom with the motion. The model is nonlinear because it contains terms where the states are multiplied together, as well as trigonometric functions. The disturbances appear explicitly through the $w_i$ terms. The imperfect state measurement feature arises naturally because, typically, the controller can access only noisy measurements of the states describing the position and orientation of the vehicle. We will use this model to develop a solution to the basic motion planning problem and then expand that solution to address the obstacle avoidance and multiple vehicle planning problems. In addition, we will construct both perfect state feedback and imperfect state measurement tracking controllers based on this model.

The remainder of this thesis is organized to correspond to the sequential design process for planning and controlling the motion of an underactuated vehicle. We start with a comprehensive literature review in Chapter 2. Chapter 3 describes the solution to the basic motion planning problem and how to combine the solution with other planning

techniques to solve more challenging problems. The motion planning algorithm generates a feasible trajectory for the underactuated vehicle to track. With a feasible trajectory in place, Chapter 4 solves the state feedback tracking control problem and proves stability for the closed-loop system. We also explore the disturbance attenuation properties and the region of attraction for the state feedback controller. Chapter 5 modifies the state feedback controller to handle the imperfect state measurement case and proves local disturbance attenuation results. Chapter 6 provides simulations illustrating the results of the motion planning and control techniques applied to the underactuated vehicle model. The final chapter provides a summary of the completed work and proposes additional research areas that came to light during the investigation.

# CHAPTER 2

# LITERATURE REVIEW

The literature associated with motion planning and underactuated vehicle control provides the foundation for the current research effort. The breadth of the existing literature is quite large, so we will present the key works related to the current effort. Our original contributions build directly on some of these recent results. The review below considers literature from the motion planning, underactuated vehicle control, and $H^\infty$-optimal control disciplines. We will describe the literature from each discipline and explain how it relates to our original contributions.

## 2.1 Motion Planning

We will survey the motion planning literature with two goals in mind. The first goal is to provide a summary of the relevant research that is related to the motion planning problem for a nonlinear underactuated vehicle. This summary will extract the key references from the vast collection of work on motion planning and explain how they are relevant to our problem. The second goal for this portion of the literature review is to justify the approach of incorporating randomized planning techniques with the basic motion planning algorithm to solve the obstacle avoidance and multiple vehicle planning problems. There are clearly other alternatives for addressing these problems, but randomized techniques offer a fair balance between being able to handle the underactuated

dynamics of the vehicles in a complex environment and being able to generate a solution quickly. Randomized techniques do not guarantee that a solution will be found, but if a solution is possible, it is very likely they will find it relatively quickly.

### 2.1.1 General motion planning references

A good starting point to understand the motion planning problem and some of the common solution techniques is the recent paper by Latombe [5]. This paper presents a summary of current motion planning research, describes some of the new problems that are emerging, and predicts what may happen with the field in the near future. Latombe emphasizes the fact that applications outside of robotics will significantly influence motion planning research. In addition, he hints at the fact that randomized planning techniques may be a key factor in developing solutions for systems with many degrees of freedom.

For a more complete introduction to the motion planning problem, there are four valuable references that are germane to our problem. The textbook by Latombe [6] is the standard reference for basic motion planning concepts and provides a comprehensive description of the subject. Latombe explains the fundamental techniques, including roadmap, cell decomposition, and potential field planning methods. He also addresses the more advanced problems of handling multiple objects, kinematic constraints, and dealing with uncertainty. Another resource is the survey paper by Hwang and Ahuja [7], which contains an extensive description of the motion planning problem. The authors also provide an excellent list of references to help guide additional research on the subject. The third general reference is a collection edited by Li and Canny [8] which focuses on nonholonomic systems. This book offers information that is more closely associated with motion planning for underactuated vehicles. The techniques addressed in the book

include steering using sinusoids and differential geometric methods. The final reference in this set is the compilation edited by Laumond [9]. Laumond's book contains six chapters that address different aspects of the problem, including planning for nonholonomic systems, probabilistic methods, and collision detection algorithms. Along with [5], these four references provide a thorough overview of the motion planning problem and common solution techniques. However, to develop our motion planning solution, we will have to examine more detailed articles in the literature.

The motion planning literature can be divided into deterministic and randomized methods. Deterministic techniques offer direct approaches for developing solutions and make it easier to prove when a solution will be found, but they can be time consuming for high-dimensional problems. By their nature, the randomized methods make it difficult to prove the results, but they offer promising techniques that can often solve challenging problems quickly. To help structure the motion planning literature review, we will first consider deterministic techniques and then shift our focus to consider randomized planning methods.

### 2.1.2 Deterministic planning techniques

Two papers by Djouani and Hamam [10, 11] consider motion planning for a ship and are directly related to this research. The authors form a nonlinear optimal control problem to find a minimum time-energy trajectory. Their algorithm uses an augmented Lagrangian approach to solve the nonlinear programming problem, which is similar to a penalty function method. The algorithm produces the control inputs for the ship to approach the desired configuration and allows the design to account for obstacles in the environment. The solution uses an iterative numerical approach and the authors do not comment on the computational requirements or how quickly the algorithm converges.

The overall approach is reasonable and offers an alternative to the basic motion planning solution we will present in Chapter 3.

The motion planning problem we are trying to solve can be formulated as a two-point boundary value problem since the initial and terminal conditions are known and the solution must satisfy the equations of motion. The multipoint shooting method offers one numerical approach to solve this challenging problem [12]. With some simplification, the method guesses a portion of the solution, computes the trajectory, uses the resulting terminal error to update the initial guess, and iterates until the final boundary condition is satisfied. Yih and Ro [13] apply the multipoint shooting method to plan motions for nonholonomic systems. Since this is an iterative technique, our basic motion planning algorithm improves on it by offering a way to directly construct the trajectory.

With a slightly different formulation, the motion planning problem can also be cast as a nonlinear optimization problem in an infinite dimensional space. Fernandes, Gurvits, and Li [14, 15] provide a solution to this version of the problem by approximating the infinite-dimensional space with a finite basis. The approximation reduces the task to a finite dimensional problem, and the authors outline how to construct the basis and then efficiently solve the optimization problem. This approach is constructive and offers another viable alternative to our proposed techniques.

The motion planning problem is posed as a different type of optimal control problem by Spangelo and Egeland [16]. The authors achieve optimal trajectory planning and collision avoidance using optimal control techniques. Their innovation is to use a performance index that is a combination of the amount of energy and the time required to execute a motion. The authors use path constraints to prevent collisions. The approach is based on a numerical shooting method to develop a solution and requires a fully actuated system to implement the results.

Arai, Tanie, and Shiroma collaborated on several motion planning articles that are useful for our problem. In [17, 18], they use time-scaling control and a bidirectional motion planning approach to develop trajectories for a robot with one passive joint. Time-scaling allows the authors to control the velocity of the active joints, which in turn provides control over the drift of the passive joint. The bidirectional motion planning algorithm searches forward from the initial condition and backwards from the terminal condition to find a complete path. In this work, the initial and final configuration put the robot at rest, which is not necessarily a feature we want to include for our problem. In related works [19, 20], the same authors develop a motion planning algorithm for an underactuated robot that accounts for the dynamics of the system. They construct a trajectory by combining translational and rotational motion segments to achieve the overall objective. Although we also exploit the dynamics of the vehicle in planning the path, the results from the work of Arai, Tanie, and Shiroma [17, 18] apply to only a specific type of underactuated robot and do not generalize to other systems.

Lynch, Shiroma, Arai, and Tanie [21] also use time-scaling to control an underactuated vehicle. Their motion planning algorithm is a best-first search and allows the authors to avoid obstacles in the environment. The technique will yield a path that approaches the desired configuration, but may not reach it exactly. The search technique relies on the small-time locally controllable property of the robot to advance the path. It does not offer a direct method for constructing a trajectory.

Reister and Lenhart [22] document a method for finding time-optimal paths for constant-speed vehicles. The paths are constructed from arcs of circles and straight line segments to connect the initial and final configurations. The approach contains elements similar to those originally proposed by Reeds and Shepp [23]. Reeds and Shepp catalog the types of paths available to a car-like vehicle that can travel forwards and backwards and describe an approach to identify the shortest length path. Sussmann and

Tang [24] advance the results of Reeds and Shepp by showing how to construct the solution using geometric techniques and nonlinear optimal control. In addition to providing an alternative derivation for the planning results, Sussmann and Tang [24] were able to reduce the catalog of required paths to 46 from the original 48 proposed by Reeds and Shepp. Souères and Laumond [25] further refined the planning approach using Reeds-Shepp curves by showing how to partition the configuration space into regions where the same type of path is optimal to reach the region. This partitioning provides an efficient method for selecting the appropriate trajectory based on the vehicle's initial configuration. We will extract elements of these approaches when we consider the problem of constructing minimum distance trajectories for an underactuated vehicle.

Another approach to motion planning formulates the problem from a game-theoretic perspective and applies numerical techniques to search for an optimal path. LaValle [26] presents a general framework for this approach and applies it to several variations of the motion planning problem, including obstacle avoidance and multiple vehicle motion planning. LaValle and Hutchinson [27] discuss optimal motion planning for multiple robots. Their intent is to solve the collision avoidance problem and simultaneously optimize a performance measure for each robot. In addition, one of their approaches using independent roadmaps for each robot offers a compromise between centralized and decoupled motion planning for multiple robots. Briefly stated, centralized planning for multiple robots concatenates the individual robots into a single system with constraints that prevent collisions. The motion planning problem can then be solved for the combined system. In contrast, the decoupled approach to planning solves the problem for each robot independently and then locally resolves conflicting motions as they arise. By finding a balance between these two common techniques, LaValle and Hutchinson [27] are able to consider a wider range of motions and to find optimal solutions to the problem.

A purely decoupled planning algorithm that might be useful for our situation also poses the planning problem as a noncooperative dynamic game among the vehicles. Tomlin, Pappas, and Sastry [28, 29] developed this approach in the context of air traffic management, but the main principles could be adapted to handle underactuated vehicles. The basic idea with the approach is to create a protective bubble around each vehicle and to replan motions if the existing plans cause a vehicle to trespass into another's protected zone. The approach assumes the vehicles cannot communicate and hence cannot cooperate in replanning their motions. To handle this situation, each vehicle will assume the other will take the most hostile action possible and will plan to maintain separation regardless of the other vehicle's action. This worst-case perspective can be formulated as a dynamic game, and solutions can be developed from the resulting Hamilton-Jacobi equations. The replanned motions involve making turns or adjusting speeds to avoid hazardous situations. To adapt the approach for the underactuated vehicle problem, we would have to account for the new dynamic equations and introduce obstacles into the environment.

Another decoupled approach treats the individual position paths as deformable elastic bands that stretch or contract to accommodate changes in the environment or the motion of other vehicles. The preliminary ideas on this line of research were presented by Quinlan and Khatib [30] and were later extended by Khatib, Jaouni, Chatila, and Laumond [31] to handle nonholonomic vehicles, and by Brock and Khatib [32] to address the multiple vehicle case. In [32], the authors explain how to use velocity tuning or a combination of velocity and trajectory tuning to resolve conflicting motions after the individual motions have been planned. The elastic band approach uses potential fields to determine the new shape of the trajectories in response to the other vehicles or newly discovered obstacles. We could again modify this approach for the underactuated vehicle by incorporating the new dynamic equations to account for the underactuated nature of the motion.

### 2.1.3 Randomized planning techniques

Randomized techniques inject unpredictable motions or search steps into the process to achieve excellent results in practice. Barraquand and Latombe [33] combine randomization with a potential field method to solve the motion planning problem. Their approach uses potential functions to plan the general motions of the robot and the randomization enters the algorithm if the robot enters a local minimum that is not the goal state. The approach uses a random walk from the local minimum to escape its region of attraction. The robot can then follow the potential function again in search of the goal state. The authors showed that the approach works well for robots with many degrees of freedom, for complicated environments, and when multiple robots are navigating the same environment. For the multiple vehicle case, the authors treated the vehicles as a single higher-dimensional system to perform coordinated planning.

Another randomized technique for solving motion planning problems is the probabilistic roadmap planner developed by Kavraki, Švestka, Latombe, and Overmars [34, 35]. This planner solves the problem in two steps for robots with many degrees of freedom operating in static environments. The first step is a learning phase where a roadmap is randomly generated to infiltrate the free configuration space. The learning process relies on a fast local planner to connect the individual configurations. After the learning phase, the second step is the query phase where the algorithm attempts to generate a path between two free configurations. The query phase tries to connect the start and goal configurations to the roadmap and then determine a route along the roadmap to connect the endpoints. After a suitable learning period, the approach can quickly generate solutions for complicated robots and challenging environments. The authors note that the approach should be customized for each application to improve the results. Since the approach contains random aspects, it is not possible to guarantee the performance,

but Kavraki, Kolountzakis, and Latombe [36, 37] have made efforts to quantify the approach and determine bounds on the number of points that should be examined to find a solution with a given probability. This approach has been extended to handle robots with nonholonomic constraints [38] and when multiple robots are moving in the environment [39]. See [40] for a detailed analysis of the randomized roadmap planner and the approach developed by Barraquand and Latombe.

LaValle collaborates with Kuffner [41] to develop a randomized search technique that accounts for the dynamics of the vehicle to identify feasible trajectories. The randomized technique is called rapidly-exploring random trees (RRTs). The approach builds a tree between the initial and goal states by sampling the free configuration space and planning motions between the samples. The type of tree and sampling scheme can be tailored to improve the performance of the algorithm. The approach easily incorporates kinodynamic constraints for the vehicle and obstacles in the environment. In addition, the approach offers attractive advantages in computation speed and flexibility that may be useful when we address the obstacle avoidance problem and planning for multiple vehicles.

The paper by Hsu, Latombe, and Motwani [42] uses a technique similar to rapidly-exploring random trees to generate paths between initial and goal states. The authors expand from the initial and goal states by choosing points that have relatively few neighbors and sampling the free space around these points. The approach then tries to connect the existing tree to the new points. The algorithm also tries to connect the trees growing from the initial and goal states together to complete the path. Related works by Kindel et al. [43] and Hsu et al. [44] apply the technique to systems with kinodynamic constraints to achieve excellent results. The authors demonstrate that the approach performs well when the environment includes multiple moving obstacles. We note that the main difference between this approach and RRTs is in how the existing path is extended into

the unexplored regions. This difference is relatively minor and both approaches produce similar results.

Hu et al. [45] developed another planning technique that layers random motion on top of a deterministic plan to develop collision-free motions. Their decoupled planning approach was designed to resolve potential conflicts between aircraft. The algorithm efficiently estimates the probability of a conflict and then adapts the motion of each aircraft to reduce the likelihood of a collision. This randomized approach offers a potential solution to the multiple vehicle planning problem.

The recent paper by Leven and Hutchinson [46] presents another variation on the probabilistic roadmap planning approach. Their contribution solves the problem by first finding a roadmap to represent an environment with no obstacles. The authors then determine a mapping between regions in the workspace and segments of the roadmap. With these two preprocessing steps completed, the algorithm can then plan motions in real time when obstacles are present. To do so, the real-time planner first uses the mapping to eliminate the portions of the roadmap that are blocked by obstacles. Then the planner searches the remaining roadmap to connect the initial and goal configurations. This approach is extremely successful at developing plans for high-dimensional systems, but it requires the extensive preprocessing stage to develop the solutions.

For our motion planning problem, we are seeking a technique that can quickly generate paths for nonlinear underactuated vehicles, that can handle obstacles in the environment, and that can accommodate planning for multiple vehicles. Chapter 3 presents our original solution to the basic motion planning problem, but this approach alone cannot handle the obstacle avoidance and multiple vehicle planning problems. The recent trends in the literature indicate that randomized techniques offer the best chances for quickly generating solutions to these more demanding problems. Other approaches may be able to guarantee finding solutions when they exist, but the computational requirements make

these algorithms impractical with current processing speeds. From the assortment of randomized techniques available, we selected RRTs [41] as a tool to help address the motion planning problem. The approach appears to be sufficiently fast and is flexible enough to handle the variations of the underactuated vehicle problem. We will use our solution to the basic motion planning problem to correct some of the deficiencies in the RRTs algorithm and report on the results in Chapter 3.

## 2.2 Underactuated Vehicle Control

A fair amount of research effort has been applied to study the control problems associated with underactuated vehicles in air, land, sea, and space environments. For our contributions, we used a model of an underactuated ship and posed a tracking problem inspired by the work of Pettersen and Nijmeijer [3, 47]. Their controller is based on a variation of the backstepping technique and allows the ship to recover from initial errors to track a reference trajectory. Their model does not include disturbances and they have not proven stability for the output feedback controller used for the experiments described in [47]. The theoretical results presented by Pettersen and Nijmeijer rely on a set of transformations that put their nonlinear system in chained form so that they can apply the approach Jiang and Nijmeijer presented in [48]. Pettersen and Nijmeijer developed a different way to perform tracking for an underactuated vehicle by combining backstepping with averaging as reported in [49]. The model for this problem did not include disturbances and the resulting motion displayed oscillations in the orientation and lateral velocity of the vehicle. These researchers also teamed to study the output feedback tracking problem for ships in [50], but this effort did not consider underactuated vehicles.

Godhavn developed a more traditional type of backstepping controller to make a ship track a desired trajectory [51]. This work was recently extended by Toussaint, Başar, and Bullo [52] to account for vehicles with generalized forces. Godhavn's approach required the ship to move along straight lines or arcs of circles to guarantee stable zero dynamics for the nonlinear system. Although our approach also restricts the class of reference trajectories the vehicle can track, we allow for a much wider range of motion than just straight lines and arcs of circles. In addition, under certain initial conditions, Godhavn's approach would allow the ship to rotate 180 degrees and track the reference path backwards. Our approach corrects this deficiency, and the simulations show that the vehicle makes natural maneuvers to recover from poor initial conditions.

Other results in underactuated vehicle control come from Leonard [53–55]. Leonard's research focuses on underwater vehicle control, but the system models and control theories are similar to those for surface vessels. Her results rely on energy-based methods and exploit the natural dynamics of the vehicles to stabilize a steady motion, which is also known as a relative equilibrium.

Bullo has adopted a geometrical approach for controlling the motion of a vehicle along relative equilibria. His work with Leonard and Lewis, as reported in [56–58], considers underactuated vehicles. In addition, Bullo and Murray [59] have recently used the geometric description to develop results for a fully actuated vehicle, and Bullo has provided in [60] a general description of the geometrical version of the problem.

Although the above references focus on nautical vessels, underactuated control techniques have been developed for other vehicles as well. Reyhanoglu, van der Schaft, McClamroch, and Kolmanovsky [2] have presented a general framework for studying underactuated systems, which offers a compact introduction to the subject and includes applications to different types of vehicles. Hauser, Sastry, and Meyer [61] used exact linearization to control the motion of an underactuated air vehicle. Martin, Devasia, and

Paden [62] considered the same type of air vehicle as Hauser, but approached the problem using flat outputs to find an appropriate controller. Reyhanoglu [63,64] and Reyhanoglu, van der Schaft, McClamroch, and Kolmanovsky [65] addressed the underactuated spacecraft control problem. Underactuated robots have been discussed in Spong [66, 67] and the references therein.

## 2.3 $H^\infty$-Optimal Control

The final set of literature related to our problem addresses the solution to the $H^\infty$ -optimal control problem. To help explain the rationale for selecting the combination of linearization and $H^\infty$ design for our approach, we will start by reviewing a variety of standard nonlinear control techniques and describe why they are not appropriate for our problem. We will then describe the literature related to the tracking control problem with an emphasis on $H^\infty$ techniques.

### 2.3.1 Alternative nonlinear control techniques

The first nonlinear control approach we considered for this problem was exact linearization. Isidori [68] provides a complete description of the theory behind this approach for single-input single-output systems, as well as for multiple-input multiple-output systems. The essence of the approach is to find a coordinate transformation, which performs a change of variables, and a state feedback for the nonlinear system, such that in the new coordinates and with the feedback the resulting system is linear with respect to a new input. Once we know the exact linearization of the system, we can use any of the well established linear design techniques to develop a controller for the system. We can then reverse the coordinate transformation and the state feedback rule to find the corresponding controller for the original nonlinear system. We cannot use exact linearization for the

underactuated vehicle control problem because the system does not satisfy the sufficient conditions to use the approach. Specifically, the equations of motion for the vehicle do not have a well-defined vector relative degree, which prevents us from performing the exact linearization.

A recent paper by Devasia, Chen, and Paden [69] is closely related to the exact linearization approach and deserves a few comments. The authors developed a nonlinear operator that allows them to invert a nonlinear system to reconstruct the input from the desired output trajectory. This problem is very similar to the one we are addressing here and the technique relies on the same tools used for exact linearization. The significant drawback with the approach that makes it inappropriate for our problem, however, is that the control law does not work very well when there are unmodeled dynamics or disturbances.

A related alternative to exact linearization is input-output linearization, which is also known as partial linearization [68]. In this variation, we select a set of outputs and then find the coordinate transformation and state feedback to make the new system linear with respect to the outputs. This approach is especially useful if we are interested in tracking just a portion of the state vector and do not need to control the remaining states. Input-output linearization has the by-product of generating zero dynamics, which describe the internal behavior of the transformed system. To develop well-defined results, the zero dynamics must be stable. Relating this approach to our problem, we have two control inputs available in the design and could perform input-output linearization if we select appropriate outputs. The most appropriate outputs to influence the motion of the vehicle are the $x$-$y$ position coordinates. This choice for outputs does not lead to a well-defined relative degree, so the approach will not work. In addition, using the $z_i$ position variables described in Section 1.4 does not resolve the problem. We could input-output linearize the system if we chose $x$ and $\psi$ as the output variables, but having the ability to exactly

track these two elements of the state vector leads to trajectories that do not follow the $y$ position variable. Not being able to track the $y$ position translates into exceptionally poor performance. Input-output linearization is feasible for our system, but it produces designs that do not necessarily track the entire configuration, which is one of our goals.

Another common nonlinear control design technique is integrator backstepping. The standard reference for backstepping is the text by Krstić, Kanellakopoulos, and Kokotović [70], which also describes adaptive control techniques. The basic idea behind backstepping is that if we can write the system equations with a certain structure (parametric strict-feedback form), then there is an iterative procedure to build a controller by considering one layer of the system at a time. As mentioned in Section 2.2, backstepping has been used to address the nonlinear underactuated vehicle problem we are considering [3, 51]. The backstepping approaches offer alternatives to our design, but they do not explicitly account for the disturbances that enter the equations of motion. The tracking controller we will present includes the disturbances in the design process, and we will demonstrate how it performs better than the two backstepping approaches when we present the simulation results in Chapter 6.

Two other techniques that could be used to solve this type of nonlinear control problem are gain scheduling and pseudolinearization. See Khalil's text [71] for a general description of gain scheduling, and the paper by Reboulet and Champetier [72] for an introduction to pseudolinearization. Briefly, gain scheduling linearizes the system equations about a set of operating points and then determines a switching schedule among the points to maintain control over the system. The key challenge in this approach is to design the switching schedule to achieve the desired performance. Pseudolinearization is a related approach, where a coordinate transformation and a feedback are used to make the system such that when the new system is linearized, the results do not depend on the operating point. This approach allows the designers to develop a single controller based

on a linear system, and it does not require scheduling to account for different operating points. Both of these techniques could be applied to nonlinear underactuated systems, but we chose to study $H^\infty$ techniques instead. We will leave the application of gain scheduling and pseudolinearization to underactuated systems as open research problems.

We have outlined a variety of nonlinear control techniques that are standard approaches for the type of problem we are considering. We have decided to apply linearization along with $H^\infty$ design techniques to address the problem because combining these methods offers a tractable approach with reasonable performance. The ability of this approach to solve the nonlinear underactuated vehicle tracking problem has not been documented and we are contributing to fill that vacancy. The linearization simplifies the design, but restricts our analysis so that we can only prove local results. The $H^\infty$ techniques partially compensate for the linearization because they are robust with respect to disturbances. Since our problem statement explicitly includes disturbances, we must develop an approach to attenuate their effect on the system. The approach also allows us to account for the linearization errors by incorporating the remaining nonlinear terms into the disturbances. This approach is more conservative, but allows us to make precise statements about the closed-loop performance.

To complete this section, we will now review the literature directly related to tracking control and the disturbance attenuation problem.

### 2.3.2 Tracking control literature

There is a great deal of literature that addresses the tracking control and the output feedback disturbance attenuation problems. In this summary, we will attempt to highlight some of the key contributions to these fields. We will mention the papers that provide

the fundamental results and make more detailed comments on the references that are directly related to our solution of the problem.

Three papers by van der Schaft lay the ground work for studying nonlinear $H^\infty$ control. In [73], van der Schaft shows how to relate the disturbance attenuation performance of a nonlinear system to the performance of the linearization of the system. The paper captures the basic approach we used in the controller design for the underactuated vehicles and explains how the Riccati equations for the linearized system are related to the Hamilton-Jacobi equation for the nonlinear system. This first paper considered time-invariant systems and used a performance index based on an infinite time horizon. In contrast, our problem has a time-varying nonlinear system (and linearization) and a finite horizon performance index. Van der Schaft expands on these results in [74], where he accounts for the finite horizon performance index and provides a more detailed description of the results. In this paper, Remark 24 is relevant to our problem. In that remark, van der Schaft suggests an approach to prove that the solution to the linearized problem maintains the same performance level as the solution to the nonlinear problem. In addition, the remark indicates the region of attraction for the linearized solution will be contained in the region of attraction for the nonlinear solution and the latter region is the largest one possible. In the third paper [75], van der Schaft summarizes the previous results and includes the solution to the $H^\infty$ dynamic output feedback problem. The solution to this problem is based on two Riccati equations and uses the game-theoretic approach from [76] to develop the results. The three papers by van der Schaft provide a basic introduction to the nonlinear $H^\infty$ control problem we are trying to solve for an underactuated system.

Isidori provides a comprehensive review of disturbance attenuation for nonlinear systems in [77]. The paper develops the results in the linear setting and then highlights the similarities in finding a solution to the nonlinear problem.

Our underactuated vehicle control problem involves tracking a desired trajectory, and we have formulated the problem in terms of the error dynamics, so it is essentially a regulation problem. Isidori and Byrnes [78] address the output regulation problem for nonlinear systems. Their results reiterate the fact that the solution requires solving a nonlinear partial differential equation. Although this problem is generally difficult to solve, Krener [79] describes approaches for iteratively constructing the solution developed by Isidori and Byrnes. The nonlinear system in [78] is time-invariant and the authors allow the time horizon to extend to infinity, so we cannot apply the results directly to our problem.

To overcome one of these limitations, Ball, Kachroo, and Krener [80] recently addressed the nonlinear $H^\infty$ tracking problem and provided a solution for the finite time horizon case. As expected, the results depend on finding solutions to Hamilton-Jacobi inequalities. In addition, the authors have developed numerical techniques for estimating solutions to the inequalities and promise to provide more details about their methods in the future.

Since our controller designs also depend on measurement feedback, the papers by Isidori and Astolfi [81], and Ball, Helton, and Walker [82] are of interest. Both papers consider the infinite time horizon versions of the $H^\infty$-control problem for a nonlinear system with measurement feedback. In addition, both papers show that the solution depends on solving a pair of partial differential equations, but the approaches are slightly different. In [81], the authors prove that the solution satisfies the required dissipation inequalities and relate the results to the more familiar linear case. In [82], the authors rely on a game-theoretic approach to the problem to describe the solution. As described below, we will also adopt the game-theoretic approach, since it provides a natural setting to analyze the time-varying finite horizon problem.

James and Baras [83] present a slightly different solution to the $H^\infty$ output feedback control problem for nonlinear systems. Their more general results show that the required $H^\infty$ controller is expressed in terms of the information state of the system. The information state is infinite dimensional and captures observable information that is important for the controller. The information state may not provide the best estimate for the state of the system, but it does carry the history which allows us to design the optimal disturbance attenuating controller. The infinite-dimensional nature of the information state introduces a significant obstacle to computing the solution. James and Baras [83] developed their results using discrete time systems and briefly commented on how to adapt the results for continuous time systems. Although this approach is theoretically sound, we will not focus on it because it does not readily lead to a practical solution to our problem.

The one paper that brings together all of the keys aspects of the problem we are considering is by Lu [84]. Lu addresses the $H^\infty$ output feedback control problem for nonlinear time-varying systems with a finite time horizon. The author presents the sufficient conditions to find a solution in terms of the familiar Hamilton-Jacobi inequalities. As we have mentioned before, this technically accurate solution relying on partial differential inequalities is difficult to implement for nonlinear systems. The controller design approach presented in Chapter 4 uses linearization and the system decomposition to avoid the problems associated with the full nonlinear solution.

We have adopted a game-theoretic approach to address the control problems, and Başar and Bernhard [76] provide a complete account of the solution from this perspective for both linear and nonlinear systems. Since this book is a single source that contains most of the results required for our disturbance attenuation analysis, we will use its notation to present our solution. As a secondary reference, Başar and Olsder [85] provide additional material on noncooperative dynamic games. We followed the results in this book to

help derive the optimal solutions for affine-quadratic disturbance attenuation problems. The paper by Pan and Başar [86] supplies some key results on how the $H^\infty$ controllers developed for a nominal linear system are still optimal when nonlinear perturbations are introduced. We rely on these results to complete the disturbance attenuation proofs for our controller designs.

The $H^\infty$ controller design uses linearization about a trajectory to generate the solution and prove stability. Our results resemble those originally developed by Cheng [87] and Walsh, Tilbury, Sastry, Murray, and Laumond [88], but we extend them to account for the disturbances in the system.

The final set of papers in this literature review describe four different techniques for obtaining disturbance attenuation results for specific nonlinear systems. Elgersma, Stein, Jackson, and Yeichner [89, 90] use linearized system equations to study $H^\infty$ controller designs for a rigid spacecraft. Even though they used $H^\infty$ techniques to analyze the system, the authors reverted to a simple linear control law for the final design. In the second technique, Kang [91] uses the unique geometric structure in the rigid spacecraft problem to show how a simple proportional-derivative control law can be used to satisfy the Hamilton-Jacobi inequalities associated with the problem. The geometry of the problem enters the solution by providing convenient bounds on some of the terms that help reduce the complexity of the inequalities. The results by Dalsmo and Egeland [92] and Dalsmo [93] address the same problem and are extremely similar to those by Kang, except they use quaternions to express the system coordinates. The new coordinates allow the authors to extend some of the local results in [91] to be global in nature. The final disturbance attenuation technique comes from the paper by Zasadzinski, Richard, Khelfi, and Darouach [94], and applies to robot manipulators. In this case, the authors show how to rearrange the equations of motion for the system and incorporate a linear control law to satisfy the Hamilton-Jacobi inequalities. The solution uses output feedback

to perform tracking and imposes bounds on the type of trajectories that can be followed to complete the proofs. These four techniques provide a cross section of the options available to develop control laws, but they all rely on fully actuated controllers, and there does not seem to be any way to generalize them to accommodate underactuated systems. In view of this, there is a need to develop new techniques to address the problem of controlling underactuated systems.

This section described the standard control techniques that should be considered first when dealing with nonlinear systems. We concluded that most of the standard approaches were not suitable for the underactuated vehicle tracking problem and we selected the combination of linearization and $H^\infty$ techniques to develop a relatively simple solution with reasonable performance characteristics. We also included papers describing significant results with $H^\infty$ control techniques as well as papers addressing the tracking problem. The literature review indicates that our approach to the problem is novel and shows how our contributions relate to the established results.

## 2.4 Literature Review Summary

We presented a sample of the relevant literature from the motion planning, underactuated systems and $H^\infty$-optimal control fields to identify how the current work contributes to these disciplines. This review helps establish the originality of our results and also points to additional related research areas that remain as open problems. Having described this background material, we are prepared to present our solution to the motion planning and control problems for a nonlinear underactuated vehicle.

# CHAPTER 3

# MOTION PLANNING FOR UNDERACTUATED VEHICLES

The first step in the design process is to plan the motion for the underactuated vehicle. Our objective is to generate a feasible trajectory that joins an initial configuration with a goal configuration in the alloted time frame. A feasible trajectory is one that the underactuated vehicle could follow if there were no disturbances and the vehicle's initial configuration aligned with the trajectory. There are many ways to address this problem, so we will present a direct approach for constructing a trajectory using numerical methods and $H^\infty$ techniques. In addition to solving the basic motion planning problem, we will present solutions to the obstacle avoidance and multiple vehicle planning problems and we will comment briefly on the minimum distance planning problem. Before addressing these items, we will provide a precise formulation of the general motion planning problem.

## 3.1 Problem Formulation

To formulate the basic motion planning problem, we assume we are given the nonlinear dynamics describing the vehicle we want to control. We are also given the initial and final configurations for the vehicle as well as a time interval for the motion. Denote the initial and final configurations as $\mathbf{q}_0$ and $\mathbf{q}_f$, respectively, and the time interval as $[t_0, t_f]$. Our goal is to find the control inputs to move the vehicle from $\mathbf{q}_0$ to $\mathbf{q}_f$ during the

time interval $[t_0, t_f]$. We can then apply these control inputs to the equations of motion for the vehicle to generate a feasible trajectory.

For the basic motion planning problem, there are no obstacles in the environment and no other requirements for the vehicle, except that its motion must satisfy the dynamic equations. We will introduce obstacles and additional vehicles after solving the basic problem. Even though the vehicle is underactuated, there are still an infinite number of control laws that will move the vehicle from one configuration to the other in the given time. We would like to find a technique for directly generating the required controls, which is efficient in terms of the required computations, the type of path followed, and the amount of control effort expended.

We note that we have selected a fixed time interval for the motion planning problem. The problem can also be formulated with the time interval as a variable, which adds an additional degree of freedom to the design. For example, with a flexible time interval the design could minimize the time required to complete a motion. We selected the fixed time interval approach because it is a realistic requirement for many applications. Situations with underactuated vehicles that require completing a motion at a specified time include docking a ship at a busy port, maneuvering an aircraft arriving at a gate, and guiding a missile to a target in a coordinated attack. In each of these cases, specifying the time interval in advance is an important part of the design and it should not be left as a variable.

Having formulated the basic motion planning problem, Section 3.2 presents a solution to the problem for underactuated nonlinear vehicles using numerical methods and $H^\infty$ techniques. Section 3.3 describes how to build upon the basic solution to address the obstacle avoidance, multiple vehicle, and minimum distance planning problems. We conclude the chapter with a brief summary in Section 3.4.

## 3.2 Basic Motion Planning

The basic motion planning algorithm consists of four main steps to generate a feasible trajectory that moves the vehicle from $\mathbf{q}_0$ to $\mathbf{q}_f$. The first step generates a polynomial curve in the $x$-$y$ plane that connects the initial and final positions and has the proper orientation at each endpoint. The second step uses the curve to estimate the complete trajectory by assuming the vehicle's position aligns with the curve. We call this set of configurations the *candidate* trajectory. Along with the estimate for the trajectory, the second step also finds approximate values for the control inputs for the system, which we denote as the *candidate* inputs. At the end of the second step, we have a complete set of configurations that take the vehicle from $\mathbf{q}_0$ to $\mathbf{q}_f$ during $[t_0, t_f]$. The third step uses the candidate inputs to generate a feasible trajectory for the vehicle. If this feasible trajectory approaches the desired final configuration, then the fourth step is not required. If the feasible trajectory is not satisfactory, the fourth step implements an $H^\infty$-filter which uses the underactuated dynamics of the vehicle to update the estimate for the trajectory. With an updated trajectory estimate we can also improve the estimate for the control inputs. We will iterate the $H^\infty$-filter until the resulting feasible trajectory approaches the final boundary condition or the iteration does not improve the solution. We now provide a detailed description of each step as it applies to the underactuated ship model presented in Section 1.4.

### 3.2.1 Polynomial position curve generation

The first step in the motion planning process uses one of two established techniques to generate a polynomial position curve connecting the initial and final positions for the vehicle with the correct orientation at each endpoint. The first technique uses cubic splines to generate the curve and is sufficient for most motion planning problems. The

second technique uses Pythagorean hodograph curves to connect the points and is slightly more complicated than the cubic spline approach. The additional complexity associated with the second approach may be justified if we want to quickly calculate the curve length or the curvature along the curve. Appendix A provides detailed information about both techniques for generating the polynomial position curves.

Both curve generation techniques require knowledge of the initial and final configurations for the vehicle, which are given as

$$\begin{aligned} \mathbf{q}_0 &= \begin{bmatrix} u(t_0) & v(t_0) & r(t_0) & x(t_0) & y(t_0) & \psi(t_0) \end{bmatrix}^{\mathrm{T}} \\ \mathbf{q}_f &= \begin{bmatrix} u(t_f) & v(t_f) & r(t_f) & x(t_f) & y(t_f) & \psi(t_f) \end{bmatrix}^{\mathrm{T}}. \end{aligned}$$

We compute the initial and final velocities in the inertial frame, $\dot{x}(t)$ and $\dot{y}(t)$, by using $\mathbf{q}_0$ and $\mathbf{q}_f$ to evaluate (1.4) and (1.5) at $t_0$ and $t_f$, respectively. When combined with the orientation information, the initial and final velocities determine the direction of motion for the vehicle at the endpoints. We use this information along with the $x$-$y$ coordinates of the endpoints to find an appropriate polynomial position curve. Once we have the polynomial expressions for $x(t)$ and $y(t)$, we can use them to estimate the other states of the system.

### 3.2.2 Candidate trajectory estimation

The second step in the algorithm uses the polynomial position curve to estimate the complete trajectory between the endpoints and to estimate a pair of control inputs corresponding to the trajectory. To develop accurate estimates, we will use the equations of motion for the system and make some reasonable assumptions about the motion of the vehicle. We start with a few comments about notation.

We used the 'dot' notation to indicate a time derivative in Equations (1.1)–(1.6), because these equations describe how the variables evolve over time. We will use the 'prime' notation to denote the derivative of a polynomial or an estimated state, so that $x'(t)$ is the derivative of the polynomial $x(t)$. We will use the 'hat' notation to denote a state or configuration estimate, so that $\hat{\mathbf{q}}$ represents the estimate of $\mathbf{q}$.

We note that we will use the Mathematica software to perform the calculations for the trajectory estimation procedure. This software allows us to take exact symbolic derivatives of complicated expressions, which simplifies the approach and improves the accuracy of the results. We could use other software packages, such as MATLAB, to get similar results, but we would have to rely on numerical estimates of the derivatives. If the time step for the numerical derivatives is small, the differences between the two approaches should be insignificant.

For the first step of the trajectory estimation, we use the polynomials for $x(t)$ and $y(t)$ as the estimates for the position of the vehicle, so that $\hat{x}(t) = x(t)$ and $\hat{y}(t) = y(t)$. These position estimates may not represent a feasible motion for the underactuated vehicle, but they do represent the type of curve we would like the vehicle to follow to satisfy the boundary conditions. We will treat the difference between the $x$-$y$ curve and a similar feasible path as a disturbance for the motion planning algorithm to attenuate. The iterative $H^{\infty}$-filter will attenuate this disturbance as it improves the estimate for the trajectory.

We will perform a local analysis of the vehicle's motion to find estimates for the other states. At any fixed time $t_1$, the instantaneous motion of the vehicle is in the direction given by the vector combination of $x'(t_1)$ and $y'(t_1)$ and is tangent to the $x$-$y$ curve. If the vehicle were moving with a steady motion along a straight line, the orientation would align with the direction of motion. As a first approximation for the orientation of the vehicle, we assume it is oriented tangent to the $x$-$y$ curve. The initial estimate for the

orientation is given by $\bar{\psi}(t)$ as

$$\bar{\psi}(t) \;=\; \arctan\left[x'(t), y'(t)\right]. \tag{3.1}$$

The function $\arctan(a, b)$ with two arguments gives the argument of the complex number $a + ib$, so that the value of $\bar{\psi}(t)$ can range from $-\pi$ to $\pi$. Equations (1.6) and (3.1) allow us to make the following initial estimate for the angular velocity $\bar{r}(t)$ as

$$\begin{aligned} \bar{r}(t) &= \bar{\psi}'(t) \\ &= \frac{x'(t)y''(t) - x''(t)y'(t)}{x'^2(t) + y'^2(t)}. \end{aligned} \tag{3.2}$$

Since the vehicle is underactuated, it cannot maintain an orientation that is tangent to the $x$-$y$ curve if the motion contains any turns. With this in mind, we can use these initial estimates for $\bar{\psi}$ and $\bar{r}$ along with the unactuated dynamic equation (1.2) to develop better estimates.

The improved estimate for $\psi$ assumes the vehicle is turning in a circle at a constant velocity and corrects for the offset angle in a turn. This approach is an improvement over the initial estimate because it assumes the vehicle is moving along an arc of a circle instead of in a straight line. The following analysis is based on one presented by Godhavn [51] for a similar type of underactuated vehicle.

If the vehicle is moving along an arc of a circle, we can use the following expressions to describe its position:

$$\begin{aligned} x(t) &= x_0 + \frac{K}{\bar{r}}\sin(\bar{r}t) \\ y(t) &= y_0 + \frac{K}{\bar{r}}[1 - \cos(\bar{r}t)] \end{aligned}$$

where $[x_0, y_0]$ is the initial position and $K$ is the constant speed of the vehicle. We can view this approximation as a first-order series expansion describing the path of the vehicle. To fully describe the motion, we would have to include higher-order terms which

would improve the accuracy, but complicate the analysis. Since we are performing a local analysis for a short time interval, the first-order approximation is sufficient. Taking derivatives of the expressions for the positions we get

$$x'(t) = K\cos(\bar{r}t) \tag{3.3}$$

$$y'(t) = K\sin(\bar{r}t). \tag{3.4}$$

We can now solve Equations (1.4) and (1.5) for $u$ and $v$ and substitute $x'(t)$ for $\dot{x}(t)$ and $y'(t)$ for $\dot{y}(t)$ using (3.3) and (3.4) to get

$$\begin{aligned} u(t) &\approx x'(t)\cos\left[\hat{\psi}(t)\right] + y'(t)\sin\left[\hat{\psi}(t)\right] \qquad (3.5)\\ &= K\left\{\cos\left[\bar{r}t\right]\,\cos\left[\hat{\psi}(t)\right] + \sin\left[\bar{r}t\right]\,\sin\left[\hat{\psi}(t)\right]\right\} \\ &= K\cos\left[\bar{r}t - \hat{\psi}(t)\right] \\ v(t) &\approx -x'(t)\sin\left[\hat{\psi}(t)\right] + y'(t)\cos\left[\hat{\psi}(t)\right] \qquad (3.6)\\ &= K\left\{\sin\left[\bar{r}t\right]\,\cos\left[\hat{\psi}(t)\right] - \cos\left[\bar{r}t\right]\,\sin\left[\hat{\psi}(t)\right]\right\} \\ &= K\sin\left[\bar{r}t - \hat{\psi}(t)\right] \end{aligned}$$

where we have used $\hat{\psi}(t)$ as the improved estimate for the orientation of the vehicle. We will write $\hat{\psi}(t)$ as

$$\hat{\psi}(t) = \bar{\psi}(t) + \psi_0(t)$$

where $\psi_0(t)$ is the offset angle for the underactuated vehicle. This offset is similar to a side-slip angle for an aircraft. We also note that $\bar{\psi}(t) = \bar{r}t$ for a constant rate turn. We now have

$$u(t) = K\cos\left[\bar{\psi}(t) - \hat{\psi}(t)\right] \tag{3.7}$$

$$v(t) = K\sin\left[\bar{\psi}(t) - \hat{\psi}(t)\right]. \tag{3.8}$$

If the vehicle is making a constant turn, the lateral velocity is not changing and we can make the approximation $\dot{v} = 0$. From a local perspective, over the short time interval

$[t_1, t_1 + \delta]$, where $\delta$ is small, the lateral velocity will not change significantly, which further supports the approximation that the lateral acceleration is zero. Using this approximation, substituting (3.7) and (3.8) into the unactuated dynamic equation (1.2), we get

$$\begin{aligned} 0 &= m_v u \bar{r} - d_v v \\ &= m_v K \bar{r} \cos(\bar{\psi} - \hat{\psi}) - d_v K \sin(\bar{\psi} - \hat{\psi}). \end{aligned}$$

Rearranging, we arrive at

$$\tan(\bar{\psi} - \hat{\psi}) = \frac{m_v \bar{r}}{d_v}$$

which implies

$$\hat{\psi}(t) = \bar{\psi}(t) - \arctan\left[\frac{m_v \bar{r}(t)}{d_v}\right]. \tag{3.9}$$

Equation (3.9) represents the improved estimate for the orientation of the vehicle, and accounts for the typical underactuated behavior as the vehicle turns. Note that if the vehicle is not turning, then $\bar{r}(t) = 0$ and we recover the initial estimate for the orientation given by $\bar{\psi}(t)$. We use the new estimate for $\hat{\psi}(t)$ to improve the estimate for $r(t)$ as follows:

$$\hat{r}(t) = \hat{\psi}'(t). \tag{3.10}$$

We now have expressions for $\hat{x}$, $\hat{y}$, $\hat{\psi}$, and $\hat{r}$. To estimate the remaining two states, we can use (3.5) and (3.6) to arrive at the following expressions for $\hat{u}(t)$ and $\hat{v}(t)$:

$$\hat{u}(t) = \hat{x}'(t) \cos\left[\hat{\psi}(t)\right] + \hat{y}'(t) \sin\left[\hat{\psi}(t)\right] \tag{3.11}$$

$$\hat{v}(t) = -\hat{x}'(t) \sin\left[\hat{\psi}(t)\right] + \hat{y}'(t) \cos\left[\hat{\psi}(t)\right]. \tag{3.12}$$

Now that we have estimates for the complete trajectory $\hat{\mathbf{q}}$, we can make estimates for the two control inputs for the system by using Equations (1.1) and (1.3) and the

derivatives $\hat{u}'(t)$ and $\hat{r}'(t)$:

$$\hat{u}_1(t) = \hat{u}'(t) - m_u\hat{v}(t)\hat{r}(t) + d_u\hat{u}(t) \tag{3.13}$$

$$\hat{u}_2(t) = \hat{r}'(t) - m_r\hat{u}(t)\hat{v}(t) + d_r\hat{r}(t). \tag{3.14}$$

The state estimation procedure fully exploits the information in the equations of motion to provide an initial trajectory estimate from a polynomial $x$-$y$ position curve. The procedure has two notable features. First, the approach develops an accurate estimate for the orientation of the vehicle in a turn by using the unactuated dynamic equation (1.2). As a result, the approach can exactly characterize motion along straight lines and arcs of circles. The second significant feature is that the approach provides an accurate estimate for the lateral velocity of the vehicle. The estimation technique finds nonzero values for the lateral velocity because the vehicle is not oriented tangent to the position curve. These two features make the trajectory estimation process very accurate and ultimately help produce feasible trajectories with smaller final configuration errors.

We also emphasize that this approach can be generalized to other types of underactuated systems. Although we have focused on a specific model to illustrate the method, we can use the same basic techniques with other vehicles. The simulations and discussion in Chapter 6 will demonstrate how easily the approach can be adapted to other vehicles.

The computations presented above outline the steps required to find the entire candidate trajectory between $\mathbf{q}_0$ and $\mathbf{q}_f$. We denote this candidate trajectory by $\hat{\mathbf{q}}(t)$. We also have made an initial estimate for the two control inputs, $\hat{u}_1(t)$ and $\hat{u}_2(t)$, which we will denote as $\hat{\mathbf{u}}(t)$. Finding these estimates marks the end of the second step in the motion planning algorithm. We now use the candidate trajectory and inputs to construct a feasible trajectory.

### 3.2.3 Generating the feasible trajectory

The next step in the motion planning process is to use the inputs $\hat{\mathbf{u}} = [\hat{u}_1, \hat{u}_2]^{\mathrm{T}}$ to numerically integrate the equations of motion given by (1.1) through (1.6) starting at $\mathbf{q}_0$. This numerical integration will generate a feasible trajectory $\mathbf{q}(t)$ for the underactuated vehicle. If the feasible trajectory approaches the desired final configuration, then we have found a solution to the basic motion planning problem and can move on to the tracking control aspects of the problem. It is possible that the feasible trajectory will not align with the estimated trajectory $\hat{\mathbf{q}}$. Differences between the two trajectories could arise because the position curve is significantly different from a straight line or a circular arc. In addition, there will be differences because we are using the fixed initial condition $\mathbf{q}_0$ to generate the feasible trajectory and it may not match the estimated initial condition.

One way to reduce the differences between the feasible trajectory and the estimated trajectory is to use the estimated initial configuration $\hat{\mathbf{q}}(t_0)$ to initiate the numerical integration. This adjustment will introduce a small error between the initial configurations for the actual vehicle and the feasible trajectory. If the configuration error is within the region of attraction for the tracking controllers, then this approach is acceptable and will yield a suitable feasible trajectory. To better understand when we can apply this correction, we will examine the region of attraction for the state feedback tracking controllers in Section 4.6.

There is another way to adjust the initial conditions that actually eliminates the final position and orientation errors, but can introduce a potentially large initial error to the system. Suppose we use the desired initial conditions and the estimated control inputs to generate a feasible trajectory that has final position and orientation errors $x_e$, $y_e$, and $\psi_e$. We can correct these errors in two steps by rotating and translating the initial conditions. We rotate the initial condition to eliminate the orientation error and then translate the

rotated system to remove the position error. We can perform this adjustment because the variables $x$, $y$, and $\psi$ do not enter into the equations of motion for the dynamics involving $\dot{u}$, $\dot{v}$, and $\dot{r}$. When we apply the control inputs to the system equations with the modified initial conditions we will get a feasible trajectory that exactly satisfies the intended final position and orientation requirements and closely matches the final velocity requirements. Unlike the previous correction technique, this approach is model specific and does not always generalize to other systems.

If either approach for adjusting the initial condition introduces a large configuration error, then the tracking controllers may not be able to recover from the initial error to reach the desired configuration. We must have another approach to reduce the potential differences between the feasible trajectory and the desired final configuration. Subsection 3.2.4 introduces an iterative $H^{\infty}$-filter that allows us to use the vehicle's dynamics to improve the estimate for the trajectory, the estimate for the control inputs, and the resulting feasible trajectory.

### 3.2.4 Iterative $H^{\infty}$-filtering

If the feasible trajectory generated by the first three steps in the basic motion planning algorithm does not approach the desired final configuration, we can use an $H^{\infty}$-filter to iteratively improve the results. The filtering approach relies on the fact that we can decompose the system equations into two subsystems that are each affine if the state of the other subsystem is known. The interlaced subsystems allow us to compute improved estimates for the state of the system that account for the nonlinear underactuated dynamics of the vehicle. We can then use the new state estimates to update the estimates for the control inputs used to generate a feasible trajectory. The general approach used

in this algorithm is similar to the interlaced $H^\infty$-filter developed for the imperfect state measurement tracking controller and described in Chapter 5.

We will derive the iterative $H^\infty$-filter starting with the equations of motion for the vehicle and assuming that the disturbances are not present. We have polynomials representing the $x$-$y$ positions we want the vehicle to track, but since it is an underactuated vehicle, it may not be able to do so exactly because the motion may be infeasible. Therefore, we will treat the $x$-$y$ curve as an imperfect measurement of the position of the vehicle. We will now use this measurement of the vehicle's position and the equations of motion to estimate the remaining four states of the system that are consistent with the measurements. We let $\mathbf{y}$ represent the measurement vector as follows:

$$\mathbf{y} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_x \\ w_y \end{bmatrix} = C\mathbf{q} + E\mathbf{w}_{xy} \tag{3.15}$$

where

$$\begin{aligned} \mathbf{q} &= \begin{bmatrix} u & v & x & y & r & \psi \end{bmatrix}^{\mathrm{T}} \\ C &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{aligned}$$

and $w_x$ and $w_y$ represent disturbance terms that capture the error in our measurements. We note that the vector $\mathbf{q}$ now has a special ordering for the states of the system that will be useful when we decompose the equations.

To develop the $H^\infty$-filter, we will first decompose the nonlinear system into two subsystems with the property that each one is affine if the state of the other subsystem

is known. We let

$$\begin{aligned}
\mathbf{q}_1 &= \begin{bmatrix} u & v & x & y \end{bmatrix}^{\mathrm{T}} \\
\mathbf{q}_2 &= \begin{bmatrix} r & \psi \end{bmatrix}^{\mathrm{T}} \\
\mathbf{q} &= \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix}.
\end{aligned}$$

Subsystem 1 is given by the four equations

$$\begin{aligned}
\dot{u} &= m_u\, v\, r - d_u\, u + u_1 + w_1 \\
\dot{v} &= m_v\, u\, r - d_v\, v + w_2 \\
\dot{x} &= \cos(\psi)\, u - \sin(\psi)\, v \\
\dot{y} &= \sin(\psi)\, u + \cos(\psi)\, v
\end{aligned}$$

which can be written compactly as

$$\dot{\mathbf{q}}_1 = A_{11}(\mathbf{q}_2)\mathbf{q}_1 + \alpha_1(\mathbf{q}_2) + B_1\mathbf{u}_1 + D_1\mathbf{w}_1. \tag{3.16}$$

Likewise, subsystem 2 consists of two equations

$$\begin{aligned}
\dot{r} &= m_r\, u\, v - d_r\, r + u_2 + w_3 \\
\dot{\psi} &= r
\end{aligned}$$

and can be written as

$$\dot{\mathbf{q}}_2 = A_{22}(\mathbf{q}_1)\mathbf{q}_2 + \alpha_2(\mathbf{q}_1) + B_2\mathbf{u}_2 + D_2\mathbf{w}_2. \tag{3.17}$$

We want to extract as much information about the nonlinear system as possible, so we calculate the linear portion of the affine terms $\alpha_1$ and $\alpha_2$ as follows:

$$A_{12} = \frac{\partial \alpha_1(\mathbf{q}_2)}{\partial \mathbf{q}_2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \qquad A_{21} = \frac{\partial \alpha_2(\mathbf{q}_1)}{\partial \mathbf{q}_1} = \begin{bmatrix} m_r v & m_r u & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

We can now rewrite Equations (3.16) and (3.17) as

$$\dot{\mathbf{q}}_1 = A_{11}(\mathbf{q}_2)\mathbf{q}_1 + A_{12}\mathbf{q}_2 + [\alpha_1(\mathbf{q}_2) - A_{12}\mathbf{q}_2] + B_1\mathbf{u}_1 + D_1\mathbf{w}_1 \tag{3.18}$$

$$\dot{\mathbf{q}}_2 = A_{22}(\mathbf{q}_1)\mathbf{q}_2 + A_{21}\mathbf{q}_1 + [\alpha_2(\mathbf{q}_1) - A_{21}\mathbf{q}_1] + B_2\mathbf{u}_2 + D_2\mathbf{w}_2. \tag{3.19}$$

We can combine (3.18) and (3.19) into a single set of equations as

$$\dot{\mathbf{q}} = \begin{bmatrix} A_{11}(\mathbf{q}_2) & A_{12} \\ A_{21} & A_{22}(\mathbf{q}_1) \end{bmatrix} \mathbf{q} + \begin{bmatrix} \alpha_1(\mathbf{q}_2) - A_{12}\mathbf{q}_2 \\ \alpha_2(\mathbf{q}_1) - A_{21}\mathbf{q}_1 \end{bmatrix} + \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \mathbf{u} + \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \mathbf{w}$$

where

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \qquad \mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}.$$

We can then write the combined subsystem equations compactly as

$$\dot{\mathbf{q}} = A(\mathbf{q})\mathbf{q} + \alpha(\mathbf{q}) + B\mathbf{u} + D\mathbf{w}. \tag{3.20}$$

We do not know the value of $\mathbf{q}$ in advance, so we will use the estimate $\hat{\mathbf{q}}$, to be defined shortly, in the terms $A(\mathbf{q})$ and $\alpha(\mathbf{q})$ to get

$$\dot{\mathbf{q}} = A(\hat{\mathbf{q}})\mathbf{q} + \alpha(\hat{\mathbf{q}}) + B\mathbf{u} + D\mathbf{w} + \left[A(\mathbf{q})\mathbf{q} - A(\hat{\mathbf{q}})\mathbf{q} + \alpha(\mathbf{q}) - \alpha(\hat{\mathbf{q}})\right]. \tag{3.21}$$

Using a change of variables, we can rewrite (3.21) as

$$\dot{\mathbf{q}} = A(\hat{\mathbf{q}})\mathbf{q} + \alpha(\hat{\mathbf{q}}) + B\mathbf{u} + D\mathbf{w} + \varepsilon\left[a(t,\mathbf{q}) + b(t,\mathbf{q})\mathbf{u} + d(t,\mathbf{q})\mathbf{w}\right] \tag{3.22}$$

where $\varepsilon > 0$ is small. We can also consider the term in square brackets in (3.21) [or equivalently in (3.22)] to be part of the disturbance term so that we can rewrite (3.21) as

$$\dot{\mathbf{q}} = A(\hat{\mathbf{q}})\mathbf{q} + \alpha(\hat{\mathbf{q}}) + B\mathbf{u} + D\tilde{\mathbf{w}} \tag{3.23}$$

where $\tilde{\mathbf{w}}$ accounts for $\mathbf{w}$ and the term in square brackets.

We will take (3.23) to be the affine differential equation that describes the motion of the vehicle. We can measure the states $x$ and $y$ and want to use those measurements to estimate the remaining states. We will use the following $H^\infty$-filter differential equations to estimate the states and compute the corresponding covariance matrix [76, pp. 298-300]:

$$\dot{\hat{\mathbf{q}}} = A(\hat{\mathbf{q}})\hat{\mathbf{q}} + \alpha(\hat{\mathbf{q}}) + B\mathbf{u} + \Sigma C^{\mathrm{T}} N^{-1}(\mathbf{y} - C\hat{\mathbf{q}}), \quad \hat{\mathbf{q}}(t_0) = \mathbf{q}_d(t_0) \tag{3.24}$$

$$\dot{\Sigma} = A(\hat{\mathbf{q}})\Sigma + \Sigma A^{\mathrm{T}}(\hat{\mathbf{q}}) - \Sigma(C^{\mathrm{T}} N^{-1} C - \gamma^{-2} Q)\Sigma + DD^{\mathrm{T}}, \quad \Sigma(t_0) = Q_0^{-1}. \tag{3.25}$$

In (3.24), we start the state estimate at the given desired state for the system. We could start the estimate at the origin, but including the known initial condition should improve the results. Also in (3.24), the $A(\hat{\mathbf{q}})$ matrix and the affine term $\alpha(\hat{\mathbf{q}})$ depend on the state estimate, which is available information and allows us to capture the nonlinear dynamics in an affine differential equation.

In Equation (3.24), we need to know $\mathbf{u}$ to compute the state estimates, but $\mathbf{u}$ represents the control inputs we are trying to determine to generate a feasible trajectory. One way to resolve this circular problem is to use the estimate $\hat{\mathbf{u}}$ in place of $\mathbf{u}$ in (3.24). We can then implement the $H^\infty$-filter to develop a new estimate for the configuration vector for the vehicle. With the new configuration estimates, we can use (3.13) and (3.14) to update the estimates for the control inputs. With revised estimates for the control inputs, we can repeat the $H^\infty$-filtering to again improve the results. We will run the iterative $H^\infty$-filter until the final configuration error no longer improves or a fixed number of

iterations has occurred. We limit the number of iterations to guarantee that the iteration will terminate. These new estimates for the control inputs may be an improvement over the previous estimates that did not use the iterative $H^\infty$-filter and could generate a more accurate feasible trajectory. We will use the final results of the $H^\infty$-filter to generate a feasible trajectory for the vehicle. We will call this feasible trajectory the *desired* trajectory for the tracking controllers and use a $d$ subscript to denote the desired configurations and inputs.

The motion planning algorithms use four steps to construct control inputs for the underactuated vehicle. The inputs generate a feasible trajectory to make the vehicle move from the initial configuration toward the final configuration in the allotted time. We will present examples of the basic motion planning algorithm in Section 6.1. Before we shift our focus to the tracking control problem, there are a few reasonable extensions to the basic motion planning problem that we should address.

## 3.3 Extensions to Basic Motion Planning

We will consider three extensions to the basic motion planning problem that rely on some of the fundamental techniques from the solution presented in Section 3.2. These extensions are to solve planning problems when obstacles are present and when other vehicles are in the environment, as well as to handle the minimum distance planning problem. As mentioned in the literature review in Chapter 2, randomized techniques offer promising solutions for difficult planning problems and we will rely on one randomized technique to find solutions to two of the extensions to the basic problem. We will describe the randomized technique of rapidly-exploring random trees before we explore the solutions to the modified motion planning problems.

### 3.3.1 Rapidly-exploring random trees

From the many techniques available to address the obstacle avoidance and multiple vehicle planning problems, we selected the rapidly-exploring random trees (RRTs) approach developed by LaValle [41,95] as the one to use for our application. This approach offers the advantages of easily handling the underactuated nature of our problem, automatically accounting for obstacles, and, with minor modifications, handling the multiple vehicle case. As mentioned in Subsection 2.1.3, this randomized technique can provide practical answers to challenging planning problems, but its random nature makes it more difficult to *prove* that the approach will generate an appropriate solution. We will focus here on how we can use elements of the basic motion planning algorithm to improve the standard RRT approach and how the new technique can plan motions for underactuated vehicles. We will leave the detailed proofs describing the ability of the new approach to generate a solution as an item for future research.

There are many variations of the RRT algorithm, so we will briefly describe the one that we implemented for our problem. The algorithm builds two search trees, with one starting at the initial configuration and the other starting from the the goal configuration. The trees consist of nodes that represent configurations the vehicle could achieve after applying an input for an incremental time step. The objective is to build the two trees towards each other so that they can eventually be linked to form a complete trajectory between the initial and goal states. The key innovation with the approach is the manner in which the trees are extended. At each time step, a random configuration in the space of all possible configurations is selected. Each state variable has an upper and lower limit, so there are bounds on the values for the configuration variables. The algorithm then searches one of the trees to find the closest node to the new configuration. The distance between two configurations is determined by a metric, which is a key element in the

design process. Once the algorithm identifies the closest node in the tree, it searches the list of possible inputs to find values that, in one time step, will move the configuration as close as possible to the new node without colliding with an obstacle. The algorithm then adds the resulting configuration as a new node in the tree. In addition, the algorithm tries to extend the tree with the new node towards the other tree. The process is then repeated with the roles of the trees swapped and the whole algorithm is iterated until the number of nodes is exhausted or the trees are connected. For additional details on the algorithm, see [41] or the documentation with the software.

As mentioned above, the metric used to compute the distance between two configurations is a critical aspect of the algorithm. The metric must provide a reasonable measure of the separation between two configurations and we must be able to compute it quickly, since it is evaluated many times in the algorithm. The standard metrics in the example models in the software can be calculated very quickly, but do not always provide a meaningful result for an underactuated vehicle. Likewise, the motion planning metrics suggested in the recent paper by Amato, Bayazit, Dale, Jones, and Vallejo [96] are not appropriate for an underactuated vehicle. We will describe how we changed the metric in the RRT software to improve the results of the algorithm and how we adapted the software to accommodate the extensions to our basic motion planning problem.

### 3.3.2 Obstacle avoidance motion planning

The first enhancement to the basic motion planning problem is to introduce obstacles into the environment. Accounting for obstacles will make the situation more realistic and add a significant challenge to the planning problem. For the planar underactuated vehicle, we will designate obstacles as regions that the vehicle is not allowed to occupy. We will assume the obstacles are fixed in the environment and the vehicle is not allowed

to contact them. The objective will be to find a feasible trajectory for the underactuated vehicle to track that avoids collisions with the obstacles. Our ability to control the vehicle will determine how close the feasible trajectory can come to an obstacle.

We solved the problem of planning motions around obstacles by modifying the RRT software in the following ways. First we developed a model for the underactuated vehicle to capture the dynamics of the system. The equations of motion for this model match those presented in Section 1.4. We set upper and lower bounds on the states for the system by choosing extreme values from previous motion planning trials. We then developed a set of inputs for the system that are representative of the inputs encountered in past experiments. There are no disturbances in the motion planning simulations, so we did not have to set values for them.

The most significant change we made to the RRT software was to the metric used to evaluate the separation between configurations. The standard metrics in the software are based on a Euclidean-type distance between the configurations. These metrics are not appropriate for an underactuated vehicle, because it is possible that two configurations could be separated by a small Euclidean distance and the vehicle would have to travel a relatively long trajectory to move between the two configurations. The new metric tries to compensate for this deficiency by including a measure of the path length for the trajectory the vehicle would actually follow. Given two configurations, the new metric computes the length of a Pythagorean hodograph curve connecting the two positions, with the curve having the proper tangent at each endpoint. The tangents are determined by the vehicle's velocity and orientation at the endpoints. The nature of the Pythagorean hodograph curve is such that we do not have to compute the actual polynomials for $x(t)$ and $y(t)$ to determine the path length, which helps reduce the processing time. The new metric sums the square of the path length with the squares of the errors in the other four state variables and then takes the square root of the result. This metric attempts

to provide a more reasonable estimate of the separation between two configurations than the simple Euclidean metrics, and our simulations indicate it can improve the results of the RRT algorithm.

We solved an obstacle avoidance problem for the underactuated vehicle using the modified RRT software with both the standard metric and the new Pythagorean hodograph curve metric. Since the approach is random, we computed the solution 20 times for each approach to get a sample of the results. In each case, we allowed the algorithm to examine up to 10 000 nodes in each tree before stopping the simulation. If the simulation did not find a solution within 10 000 steps, we counted the trial as a failure. We note that, given enough steps, both approaches have a high probability of finding a solution, so allowing the simulations to continue would likely have generated feasible trajectories. We terminated the simulations at 10 000 steps to help limit the time required to collect the data. Table 3.1 summarizes some of the key statistics from the 20 trials and example solutions are shown in Chapter 6. Note that the parameter $G_{\text{dist}}$ that determines the maximum separation between the two trees before they can be connected is different for the two metrics. We originally used the same value for both metrics, but the algorithm

**Table 3.1** Statistics from the solutions to the obstacle avoidance motion planning problem using the RRT software. We used each metric to attempt to generate 20 solutions to the problem. We classified the trial as a failure if there was no solution after 10 000 steps.

| | Standard Metric | PH Curve Metric |
|---|---|---|
| Incremental time step ($\Delta T$) | 0.2 | 0.2 |
| Maximum separation to connect ($G_{\text{dist}}$) | 0.2 | 0.5 |
| Failures | 7 | 5 |
| Average nodes for 20 trials | 5686.15 | 4284.3 |
| Average nodes in successful trials | 4745.4 | 3065.3 |
| Average path length | 15.83 | 11.15 |
| Average maximum jump | 0.383 | 0.303 |

with the standard metric and $G_{\text{dist}} = 0.5$ would produce unreasonable solutions in which the vehicle would make a large position jump between the trees. In some cases, the jump would even pass through obstacles, so we lowered the value to improve the performance. The resulting algorithms had roughly the same number of failures, so the settings appear to be comparable.

There are three advantages to using the Pythagorean hodograph curve metric compared to the standard metric. First, the algorithm could find solutions with fewer nodes in the trees, which translates into fewer exploration steps. Second, the resulting trajectories are usually shorter than with the standard metric, so the motion is more efficient. Finally, the maximum jump between any two nodes is usually smaller with the new metric, so the paths are smoother and it will be easier for the tracking controller to follow the trajectories. Note that the maximum jump is usually smaller for the Pythagorean hodograph curve method, even though we allowed a larger value for $G_{\text{dist}}$ than with the standard metric.

The 20 trials provide a fair sample of the two approaches and indicate that the metric based on a Pythagorean hodograph curve offers some improvements over the standard approaches. The new metric requires more computational time because it is more involved than the Euclidean metric, but the improved planning solutions may be worth the additional time. In addition, we have not optimized the code to compute the Pythagorean hodograph curve length, and enhanced programming may be able to narrow the gap between the two approaches.

As a final remark for this version of the motion planning problem, we note that we can smooth the trajectories generated by the RRT software to improve the motion of the vehicle. The trajectories generated by the RRT algorithm satisfy the motion planning requirements, but the resulting motion may be very rough because the inputs are randomly selected at each time step. There are two simple methods for smoothing

that are based on the basic motion planning algorithm. The first smoothing method applies the iterative $H^\infty$-filter to the feasible trajectory and we will refer to it as the *filtering* approach. The resulting trajectory is smoother than the original and retains the overall shape, so it will likely avoid the obstacles and other vehicles, as originally planned. In addition, the filtered trajectory does not have a discontinuity at the point where the two trees in the RRT algorithm are connected. Finally, the new trajectory is still feasible, so we can use the tracking control algorithms discussed in Chapters 4 and 5 to follow it.

The second approach for smoothing the original trajectory also uses elements of the basic motion planning algorithm. In this case, we sample the original trajectory and then use the basic planning algorithm to design motions between the sampled configurations. We will refer to this technique as the *sampling* approach. With this approach, the new segments are feasible paths and we can apply the tracking control algorithms to make the underactuated vehicle follow the new trajectories. It is possible that the smoothed trajectory could collide with the obstacles or other vehicles, but this problem could be corrected by sampling more points in the trajectory to maintain its shape in critical areas. At each transition point, we initialize the tracking controller with the final configuration from the previous segment to maintain continuity. The resulting trajectories are again very smooth and maintain the overall shapes developed in the original designs.

The sampling approach to smoothing offers a couple of advantages over the filtering technique. First, with the sampling approach the resulting control inputs will be smooth functions of time because they are generated by the tracking control algorithm. With the filtering technique, the inputs were based on the randomly selected inputs generated by the RRT algorithm and may be difficult to implement if there is fast switching in the input signal. The other key advantage with the sampling approach is that we can use the built-in features of the basic motion planning algorithm to adjust the time interval

for the motion. In our implementation of the RRT software, we did not include any constraints to enforce a time interval on the motion. Using the sampling approach to smooth the motion gives us the opportunity to correct this problem, which we cannot easily accomplish with the filtering approach.

### 3.3.3 Multiple vehicle motion planning

We now consider the problem of planning motions for multiple vehicles in the presence of obstacles. The vehicles may have conflicting requirements that cause their planned trajectories to intersect. Our goal is to identify feasible trajectories for each vehicle that satisfy the motion requirements and do not cause collisions. There are two basic strategies for addressing the multiple vehicle problem [6]. The first approach is called centralized planning and works by combining the individual vehicles into a single higher-dimensional system and then performing planning for the larger system. This approach is relatively simple to implement and can use standard motion planning techniques on the composite system, but increasing the dimensionality of the system can drastically increase the computational requirements. The second main approach is to independently plan trajectories for the individual vehicles and then identify and resolve any possible conflicts. This type of planning is known as decoupled planning. The method operates with lower dimensional systems, so it may be able to generate solutions faster than the centralized approach, but there are situations where this approach may not be able to resolve the conflicts and fails to finds a solution.

We considered both approaches for the multiple underactuated vehicle planning problem and selected the centralized approach to demonstrate our results. The centralized approach is easier to implement than a decoupled, decentralized one, and requires fewer modifications to the control algorithms presented in Chapters 4 and 5. More specifically,

decoupled planning techniques often rely on time scaling to resolve conflicting motions. Making a time scale adjustment would disrupt the feedback control laws that are based on a solution to a generalized Riccati differential equation with a fixed time scale.

We changed the RRT software to develop a centralized planner for multiple vehicles with obstacles in the environment. We implemented the approach for the two-vehicle case only, simply because it depicts all features of the multiple vehicle scenario, and including additional vehicles does not bring in anything conceptually new. We emphasize, however, that the time required to solve the motion planning problem grows exponentially with the dimension of the composite configuration space for the system [6, p. 377], so adding more vehicles will significantly increase the processing time.

To modify the RRT software, we doubled the number of state variables and inputs to account for the second vehicle. We then replicated the dynamic equations for the second vehicle and established another set of initial and goal configurations. We modified the metric computation to include the path length for each vehicle and the sum of the squared errors for the velocity and orientation states. We still used Pythagorean hodograph curves to compute the path lengths. The two-vehicle problem required one significant change to the tree-building algorithm to help limit the processing time. The algorithm is the same up through the point where the nearest node in a tree is selected. Once the nearest node is identified, the algorithm selects an input at random and extends the tree using this input. This random selection eliminates the need to check every input to find the best one, which is a significant savings in processing time. Since we have doubled the number of inputs, we have squared the number of input combinations, and checking each one is a substantial processing burden. The random choice for the input does not penalize the ability of the tree to search for a solution, since finding the closest node is more important than the motion after the node is found. The final change to the software was to include collision checking between the two vehicles. Since the state of the system

includes the position of each vehicle, we can easily eliminate configurations where the distance between the two vehicles drops below the minimum required separation.

We executed the modified RRT motion planning algorithm and generated viable solutions for the two-vehicle problem when obstacles were present. The vehicles avoided each other and the obstacles as they successfully moved from the initial states to the goal states. The simulation results are presented in Chapter 6 and include examples of smoothed trajectories using the techniques described in Subsection 3.3.2. For a comparison, we also attempted to solve this problem using the standard Euclidean-type metric. After several extensive experiments with trees that contained over 30 000 nodes, we were not able to find a solution to the two vehicle problem using the Euclidean metric. These results again indicate that the Pythagorean hodograph curve length is a valuable addition to the metric used in the RRT algorithm.

### 3.3.4 Minimum distance path planning

The final extension to the motion planning algorithms is to find the shortest trajectories that connect the initial and final configurations. Using the model presented in Section 1.4, we can construct a trivial solution to the minimum distance planning problem by manipulating the two inputs to bring the vehicle to rest, orient it towards the goal position, move the vehicle along a straight line to the goal position, bring the vehicle to rest again, and then adjust its orientation. Since this trivial solution is not very practical, we will formulate a more meaningful minimum distance problem by introducing two assumptions. We assume the vehicle is always moving forward with a constant velocity, and that is has an upper bound on how quickly it can turn. The literature review in Section 2.1 has discussed additional formulations of this problem. For a planar vehicle, this problem has a known solution where the path consists of straight line segments and

arcs of circles [97]. We will describe how to incorporate this solution with our basic motion planning algorithm.

We can solve the minimum distance planning problem by making a slight modification to the basic planning algorithm. The modification is to develop a candidate position path composed of straight lines and arcs of circles and then approximate the path with a polynomial to develop explicit expressions for $x(t)$ and $y(t)$. The path segments would use seventh-order polynomials to maintain smooth derivatives at the transition points between line segments and circular arcs. The standard mathematical software packages, such as Mathematica and MATLAB, can readily compute these polynomials with built-in functions. This approach was originally proposed by Godhavn [51] and is a simple extension to the basic algorithm, which already requires polynomial expressions for $x(t)$ and $y(t)$. We can then proceed with the remainder of the basic motion planning process as described in Section 3.2. This solution to the minimal distance planning problem provides a final example of how the basic solution to the motion planning problem is extremely versatile and can be modified to address several common planning situations.

## 3.4 Motion Planning Summary

We have presented in this chapter a set of motion planning algorithms that generate feasible trajectories for an underactuated vehicle. The solutions address the basic motion planning algorithm, the situations where obstacles or other vehicles are present in the environment, and the minimum distance planning problem. The solution to the basic problem uses a polynomial position curve to estimate the trajectory the vehicle follows and incorporates an iterative $H^{\infty}$-filter to improve the trajectory estimate. The obstacle avoidance and multiple vehicle problems rely on rapidly-exploring random trees to generate an initial solution, which can later be smoothed to find a more realistic trajectory.

The solution to the minimum distance planning problem uses an established technique to modify the basic algorithm. Now that we have techniques to create a feasible trajectory, we want to make the underactuated vehicle follow it. The next chapter will explain how we can make the vehicle track a trajectory even when there are disturbances affecting the system and the initial configuration does not match the expected initial configuration.

# CHAPTER 4

# PERFECT STATE FEEDBACK TRACKING CONTROL

The perfect state feedback tracking control law for the underactuated vehicle is the second step in the overall planning and controlling design process. The controller is based on $H^\infty$-optimal design techniques applied to a linearized version of the system model and forces the vehicle to follow the feasible desired trajectory while attenuating the effect of disturbances on the system. We prove in this chapter that the solution to the linearized version of the problem provides a locally exponentially stabilizing control law for the nonlinear system, and establish conditions under which the solution is valid. In addition, we examine how the disturbance attenuation parameter $\gamma$ affects the performance of the controller. We will also discuss how the controller design technique applies equally to minimum phase and nonminimum phase systems. Finally, we will use two methods to estimate the region of attraction for the state feedback control law.

## 4.1 Problem Formulation

Our current objective is to design a control law to track a feasible trajectory and to reject the effect of disturbances. We will assume that the feasible trajectory has already been created and that we have full knowledge of the required control inputs. At this stage, we also assume we can use the full state vector in the feedback control law, which

corresponds to the perfect state measurement case. Chapter 5 will discuss how to develop a controller for the imperfect state measurement case.

Using the compact notation, we recall the relevant equations for the vehicle from Chapter 1. The equations of motion for the vehicle are given by

$$\dot{\mathbf{q}}(t) = f(t, \mathbf{q}) + B(t)\mathbf{u}(t) + D(t)\mathbf{w}(t), \qquad \mathbf{q}(t_0) = \mathbf{q}_0 \tag{4.1}$$

where $\mathbf{q} \in I\!R^n$ is the configuration of the vehicle, $\mathbf{u} \in I\!R^m$ is the control input vector, $\mathbf{w} \in I\!R^p$ is the disturbance vector, and $f(t, \mathbf{q})$ is a vector-valued nonlinear function. The matrices $B(t)$ and $D(t)$ are (possibly time-varying) matrices with the appropriate dimensions. The initial condition $\mathbf{q}_0$ does not necessarily agree with the desired initial configuration for the vehicle. (To keep the results as general as possible, we do not specialize the description of the vectors and matrices to match the model presented in Chapter 1. This more general description will make it easier to apply the results to other systems in the future.)

The desired system uses the control inputs $\mathbf{u}_d(t)$ from the motion planning algorithm to generate the feasible trajectory we want to track. The desired trajectory is represented by the differential equation

$$\dot{\mathbf{q}}_d(t) = f(t, \mathbf{q}_d) + B(t)\mathbf{u}_d(t), \qquad \mathbf{q}_d(t_0) = \mathbf{q}_{d0} \tag{4.2}$$

where we assume that the initial condition $\mathbf{q}_{d0}$ is known. We want the actual trajectory to approach the desired trajectory, so it makes sense to consider the error dynamics. If we subtract (4.2) from (4.1) we arrive at the equations for the error system

$$\dot{\mathbf{q}}_e(t) = f_e(t, \mathbf{q}_e) + B(t)\mathbf{u}_e(t) + D(t)\mathbf{w}(t), \qquad \mathbf{q}_e(t_0) = \mathbf{q}_{e0} \tag{4.3}$$

where $\mathbf{q}_e = \mathbf{q} - \mathbf{q}_d$ and $\mathbf{u}_e = \mathbf{u} - \mathbf{u}_d$. Note that in (4.3) the matrices $B(t)$ and $D(t)$ are the same as in (4.1) because the control inputs and disturbances enter the equations linearly.

Tracking the desired trajectory is equivalent to finding a set of control inputs $\mathbf{u}_e(t)$ for the error system to drive the error state to the origin and keep it there. If we can find these control inputs, then we can calculate the tracking control inputs for the underactuated vehicle as $\mathbf{u} = \mathbf{u}_d + \mathbf{u}_e$. Our control design approach uses linearization about the desired trajectory to find an $H^\infty$-optimal controller.

## 4.2 Linearized $H^\infty$ Controller Design

Our objective is to develop a control law that drives the error state $\mathbf{q}_e$ to the origin even when disturbances are encountered. The tracking controller uses a relatively simple linearization approach to achieve surprisingly good results. To our knowledge, this technique for underactuated vehicle tracking has not been reported in the literature. We start with error differential equations in (4.3) and linearize the nonlinear system about the origin $\mathbf{q}_e = 0$ to get the new system matrices

$$A_e(t) = \left.\frac{\partial f_e(t, \mathbf{q}_e)}{\partial \mathbf{q}_e}\right|_{\mathbf{q}_e = 0}, \qquad B_e(t) = B(t), \qquad D_e(t) = D(t). \tag{4.4}$$

Note that $B_e(t) = B(t)$ and $D_e(t) = D(t)$ because we assumed in the problem formulation that the control and disturbance enter the system dynamics linearly. Also note that linearizing the error equations about the origin is equivalent to linearizing the actual system equations about the desired trajectory. We can now write (4.3) as

$$\dot{\mathbf{q}}_e = A_e(t)\mathbf{q}_e + B_e(t)\mathbf{u}_e + D_e(t)\mathbf{w} + \left[f_e(t, \mathbf{q}_e) - A_e(t)\mathbf{q}_e\right], \qquad \mathbf{q}_e(t_0) = \mathbf{q}_{e0} \tag{4.5}$$

where the term in square brackets is the remaining nonlinear portion of the system and is $o(|\mathbf{q}_e|)$, so that we have

$$\lim_{|\mathbf{q}_e| \to 0} \frac{o(|\mathbf{q}_e|)}{|\mathbf{q}_e|} = 0.$$

We will start the controller design process by formulating an optimal disturbance attenuation problem and then modify the results to prove local exponential stability. For the initial design, we will ignore the nonlinear term in square brackets in (4.5), but we will account for it in our stability analysis.

If we associate a quadratic performance index with the linear portion of the system given in (4.5), we can compute a solution to the $H^\infty$ control problem and find the optimal disturbance attenuating controller for the *linear* problem. Accordingly, we select the performance index to be

$$L(\mathbf{u}_e, \mathbf{w}) = |\mathbf{q}_e(t_f)|^2_{Q_f} + \int_{t_0}^{t_f} \left\{ |\mathbf{q}_e(t)|^2_{Q(t)} + |\mathbf{u}_e(t)|^2 \right\} \, dt \tag{4.6}$$

where $Q_f > 0$, $Q(t) \geq 0$, and $t \in [t_0, t_f]$. Then, the related parametrized soft-constrained performance index is

$$L_\gamma(\mathbf{u}_e, \mathbf{w}) = |\mathbf{q}_e(t_f)|^2_{Q_f} + \int_{t_0}^{t_f} \left\{ |\mathbf{q}_e(t)|^2_{Q(t)} + |\mathbf{u}_e(t)|^2 - \gamma^2 |\mathbf{w}(t)|^2 \right\} \, dt \tag{4.7}$$

where $\gamma > 0$. The solution to the $H^\infty$ control problem with full state measurements depends on finding a unique positive definite solution $Z_\gamma(t)$ to the generalized Riccati differential equation (GRDE)

$$\dot{Z} + A_e^{\mathrm{T}} Z + Z A_e + Q - Z \left( B_e B_e^{\mathrm{T}} - \gamma^{-2} D_e D_e^{\mathrm{T}} \right) Z = 0, \qquad Z(t_f) = Q_f. \tag{4.8}$$

Following the development in [76], we define the infimum of values for $\gamma$ that allow a solution to (4.8) as

$$\gamma^* := \inf\{\gamma > 0 \ : \ \text{The GRDE (4.8) does not have a conjugate point on } [0, t_f]\}. \tag{4.9}$$

As we mentioned earlier, we have assumed in this development that the entire state is available for the feedback controller so that there is a closed-loop perfect-state information structure for the corresponding linear-quadratic zero-sum differential game on the time

interval $[t_0, t_f]$. As described in [76, Appendix A], there always exists a $\gamma^* > 0$ such that for any $\gamma > \gamma^*$ we can find a positive definite solution to (4.8). The solution to the GRDE leads to a unique feedback saddle-point solution to the differential game, with

$$\mathbf{u}^*(t, \mathbf{q}_e(t)) = -B^{\mathrm{T}}(t) Z_\gamma(t) \mathbf{q}_e(t) \tag{4.10}$$

$$\mathbf{w}^*(t, \mathbf{q}_e(t)) = \gamma^{-2} D^{\mathrm{T}}(t) Z_\gamma(t) \mathbf{q}_e(t), \qquad t \geq t_0 \tag{4.11}$$

where $\mathbf{u}^*$ and $\mathbf{w}^*$ are the optimal control and worst-case disturbance, respectively.

We can solve (4.8) in reverse time for $Z_\gamma(t)$ because $A_e$ depends on only the desired trajectory, which we assumed can be computed in advance, and we can pick $\gamma > \gamma^*$ and $Q_f > 0$ in our design. The control input $\mathbf{u}^*$ provides an optimal controller for the linearized error equations (4.5), so we will set $\mathbf{u}_e = \mathbf{u}^*$. We construct the locally optimal solution for (4.1) by summing $\mathbf{u}_e(t) + \mathbf{u}_d(t) =: \mathbf{u}(t)$, since we know $\mathbf{u}_d(t)$ in advance. We can then apply

$$\mathbf{u}(t) = \mathbf{u}_d(t) - B^{\mathrm{T}}(t) Z_\gamma(t) \mathbf{q}_e(t) \tag{4.12}$$

as the state feedback controller for the full nonlinear system.

Note that we are using the linearized controller for the nonlinear system. This controller could cause the region of attraction to shrink significantly compared to the nonlinear state feedback controller that solves the Hamilton-Jacobi-Isaacs (HJI) inequality associated with the problem [76]. Using the above formulation, the corresponding HJI is a partial differential inequality given by

$$\begin{aligned} -\frac{\partial V(t, \mathbf{q}_e)}{\partial t} &\geq \min_{\mathbf{u} \in U} \max_{\mathbf{w} \in W} \left\{ \frac{\partial V(t, \mathbf{q}_e)}{\partial \mathbf{q}_e} \left[ f_e + B_e \mathbf{u}_e + D_e \mathbf{w} \right] + |\mathbf{q}_e|_Q^2 + |\mathbf{u}_e|^2 - \gamma^2 |\mathbf{w}|^2 \right\} \\ V(t_f; \mathbf{q}_e) &\equiv q(\mathbf{q}_e) \end{aligned} \tag{4.13}$$

where $V(t, \mathbf{q}_e)$ is a continuously differentiable value function. If we could solve this inequality, the optimal state feedback control law would be

$$\mathbf{u}^*(\mathbf{q}_e) = -\frac{1}{2} B^{\mathrm{T}}(t) \left( \frac{\partial V(t, \mathbf{q}_e)}{\partial \mathbf{q}_e} \right)^{\mathrm{T}}. \tag{4.14}$$

In general, for the time-varying problems we will consider, Equation (4.13) cannot be solved analytically, but a numerical solution is possible. We opted to use a linearization approach to find a reasonable control law instead of the approximate numerical solution.

The above controller design is a relatively straightforward approach to the problem and the simulations in Chapter 6 will demonstrate that it leads to a closed-loop system with reasonable tracking performance. One deficiency with this approach is that it is difficult to prove global closed-loop stability for the nonlinear system. We will focus on the local analysis, so there is a direct proof for stability.

Before we present the stability proof, we make one simple adjustment to the control law, which involves introducing an additional design parameter $\kappa(t)$, which is taken to be positive for all $t \in [t_0, t_f]$:

$$\mathbf{u}_e(t) = -\kappa(t) B^{\mathrm{T}} Z_\gamma(t) \mathbf{q}_e(t). \tag{4.15}$$

This control law is just a scaled version of the $H^\infty$ control law and offers some additional flexibility in the design. The modified control law also offers reasonable tracking performance. In Section 4.3 we will prove that (4.15) leads to a locally exponentially stable closed-loop system under mild restrictions on $\kappa(t)$.

## 4.3 Local Exponential Stability

We have already outlined the controller design process and now we want to formally analyze how well the proposed controller performs. To prove that the controller provides

reasonable tracking performance, we examine the stability of the closed-loop system without any disturbances.

We will work directly with the error equations for the system and we rewrite (4.5) without the disturbances to get

$$\dot{\mathbf{q}}_e = A_e \mathbf{q}_e + B_e \mathbf{u}_e + o(|\mathbf{q}_e|). \tag{4.16}$$

Assume we have picked $Q = q\,I$, $Q_f = q_f\,I$, where $q$ and $q_f$ are positive scalar constants, and $\gamma > \gamma^*$ to solve (4.8) for $Z_\gamma(t)$. If we apply control law (4.15) and suppress the $\gamma$ subscript on $Z_\gamma$, we get the closed-loop system

$$\dot{\mathbf{q}}_e = \left(A_e - \kappa B_e B_e^{\mathrm{T}} Z\right) \mathbf{q}_e + o(|\mathbf{q}_e|). \tag{4.17}$$

We choose a candidate Lyapunov function

$$V(t, \mathbf{q}_e) = \mathbf{q}_e^{\mathrm{T}} Z(t) \mathbf{q}_e \tag{4.18}$$

and note that for all $t \in [t_0, t_f]$ we have that $V(t, \mathbf{q}_e)$ is positive definite and radially unbounded because $Z(t) > 0$ for $t \in [t_0, t_f]$. After solving for $Z(t)$, we can also find real-valued functions $m(t)$ and $M(t)$ and constants $\underline{m}$ and $\bar{M}$ such that

$$0 < \underline{m}\,I \leq m(t)\,I \leq Z(t) \leq M(t)\,I \leq \bar{M}\,I. \tag{4.19}$$

Since $A_e(t)$ is time-varying and depends on the desired trajectory, the functions $m(t)$ and $M(t)$ will have to be determined numerically after solving for $Z(t)$. To show that the closed-loop system is locally exponentially stable, we want to show that the time derivative of the candidate Lyapunov function (4.18) is negative definite in a neighborhood of

the origin. We compute the time derivative as

$$\begin{aligned}
\dot{V} &= \mathbf{q}_e^{\mathrm{T}} \dot{Z} \mathbf{q}_e + \dot{\mathbf{q}}_e^{\mathrm{T}} Z \mathbf{q}_e + \mathbf{q}_e^{\mathrm{T}} Z \dot{\mathbf{q}}_e \\
&= -\mathbf{q}_e^{\mathrm{T}} \left[ A_e^{\mathrm{T}} Z + Z A_e + Q - Z \left( B_e B_e^{\mathrm{T}} - \gamma^{-2} D_e D_e^{\mathrm{T}} \right) Z \right] \mathbf{q}_e \\
&\quad + \mathbf{q}_e^{\mathrm{T}} \left( A_e - \kappa B_e B_e^{\mathrm{T}} Z \right)^{\mathrm{T}} Z \mathbf{q}_e + \mathbf{q}_e^{\mathrm{T}} Z \left( A_e - \kappa B_e B_e^{\mathrm{T}} Z \right) \mathbf{q}_e + 2 \mathbf{q}_e^{\mathrm{T}} Z \, o(|\mathbf{q}_e|) \\
&= -\mathbf{q}_e^{\mathrm{T}} \left[ Q + (2\kappa - 1) Z B_e B_e^{\mathrm{T}} Z + \gamma^{-2} Z D_e D_e^{\mathrm{T}} Z \right] \mathbf{q}_e + 2 \mathbf{q}_e^{\mathrm{T}} Z \, o(|\mathbf{q}_e|). \qquad (4.20)
\end{aligned}$$

If we choose $\kappa(t) \geq \frac{1}{2}$ on $[t_0, t_f]$, then for all $t \in [t_0, t_f]$ we have

$$\left[ Q + (2\kappa - 1) Z B_e B_e^{\mathrm{T}} Z + \gamma^{-2} Z D_e D_e^{\mathrm{T}} Z \right] > 0.$$

The remaining term in (4.20) is sign indefinite, but vanishes as $|\mathbf{q}_e|$ approaches the origin, so we can use the bounds on $Z$ and our knowledge of $Q$ to find a neighborhood of the origin where $\dot{V} < 0$. Following the approach presented by [87, 88], and using routine arguments, we can find some $\varepsilon > 0$ such that

$$[f_e(t, \mathbf{q}_e) - A_e(t) \mathbf{q}_e] \leq \frac{1}{4} \frac{q}{\bar{M}} |\mathbf{q}_e|, \qquad \forall \, \mathbf{q}_e \text{ such that } |\mathbf{q}_e| < \varepsilon. \qquad (4.21)$$

This implies that

$$\left| 2 \mathbf{q}_e^{\mathrm{T}} Z \, o(|\mathbf{q}_e|) \right| \leq \frac{1}{2} q |\mathbf{q}_e|^2. \qquad (4.22)$$

Combining (4.22) with (4.20), we get

$$\begin{aligned}
\dot{V}(t, \mathbf{q}_e) &\leq -\mathbf{q}_e^{\mathrm{T}} \left[ Q + (2\kappa - 1) Z B_e B_e^{\mathrm{T}} Z + \gamma^{-2} Z D_e D_e^{\mathrm{T}} Z \right] \mathbf{q}_e + \frac{1}{2} q |\mathbf{q}_e|^2 \\
&\leq -q |\mathbf{q}_e|^2 + \frac{1}{2} q |\mathbf{q}_e|^2 = -\frac{1}{2} q |\mathbf{q}_e|^2 \\
&\leq -\frac{1}{2} \frac{q}{\bar{M}} V(t, \mathbf{q}_e). \qquad (4.23)
\end{aligned}$$

Equation (4.23) leads to the following bounds on the Lyapunov function and the magnitude of the state vector:

$$V(t, \mathbf{q}_e) \leq V(t_0; \mathbf{q}_{e0}) \exp \left[ -\frac{1}{2} \frac{q}{\bar{M}} t \right] \qquad (4.24)$$

$$|\mathbf{q}_e(t)| \leq |\mathbf{q}_{e0}| \sqrt{\frac{\bar{M}}{\underline{m}}} \exp \left[ -\frac{1}{4} \frac{q}{\bar{M}} t \right] \qquad (4.25)$$

so that in an $\varepsilon$ neighborhood of the origin, we have exponential stability for the closed-loop system. Simulation results show that these convergence bounds for the system are relatively conservative.

Several remarks about the control law are now in order. First, we have not taken advantage of the term $[(2\kappa - 1)ZB_eB_e^{\mathrm{T}}Z + \gamma^{-2}ZD_eD_e^{\mathrm{T}}Z]$ in our stability analysis. This term is nonnegative definite for $\kappa \geq \frac{1}{2}$ and will improve the stability of the design as can be seen in the inequalities preceding (4.23). A more detailed analysis of this part of the Lyapunov function should lead to a tighter convergence bound on the error. In addition, since we do not rely on this term, our stability analysis would hold even if the term $\gamma^{-2}ZD_eD_e^{\mathrm{T}}Z$ was not included in the GRDE (4.8) when we solved for $Z$. Removing the $\gamma^{-2}ZD_eD_e^{\mathrm{T}}Z$ term from the GRDE is equivalent to setting $\gamma = \infty$, in which case there is always a solution to (4.8) since we require $Q \geq 0$ and $Q_f > 0$ in the design.

The second remark is that even though we have ignored the disturbances in the stability analysis, we expect the state feedback control law to exhibit reasonable disturbance attenuation properties. The reason for this expectation is that the control law closely resembles the optimal disturbance attenuating control law for the linearized system and uses the same $Z(t)$ matrix. If we choose $\kappa(t) \equiv 1$, we recover the optimal disturbance attenuating controller. The time-varying $Z(t)$ matrix is the key element of the design that provides the disturbance attenuation properties.

We have demonstrated a form of local exponential stability for the nonlinear system. Our approach does not match the standard definition for exponential stability because we consider a finite time interval and the standard approach considers all time greater than a given initial time [71]. The analysis is still helpful because if the time interval is long enough and the vehicle approaches the desired trajectory exponentially over the interval, we will have reasonable tracking in finite time. One related concern is that (4.25) indicates that extremely large values for $\bar{M}$ could lead to large changes in $\mathbf{q}_e$ as the vehicle

approaches the desired trajectory. This point will be important when we consider the role of parameter $\gamma$ in the control law.

The acceptable range for $\kappa(t)$ is the interval $[\frac{1}{2}, \infty)$, which also appears as the allowable range for a design parameter used by Walsh, Tilbury, Sastry, Murray, and Laumond [88]. The similarity is not a coincidence and requires a brief comment. Walsh, Tilbury, Sastry, Murray, and Laumond find a control law for a time-varying nonlinear system through linearization and by exploiting known bounds on the controllability Gramian. The bounds on the controllability Gramian are equivalent to the existence of a solution to a Lyapunov equation. If we remove the $[(2\kappa - 1)ZB_eB_e^{\mathrm{T}}Z + \gamma^{-2}ZD_eD_e^{\mathrm{T}}Z]$ term from (4.8), as we did in the stability proof, we have a Lyapunov equation for the linearized system. Including the term containing $B_e$ and $D_e$ in (4.8), however, allows the control law to account for how the inputs and disturbances enter the system, which provides additional robustness properties.

The final remark is that since we are using a control law based on the linearization of the system, we cannot draw any conclusions about the disturbance attenuation properties of the closed-loop nonlinear system on a global scale. To make a statement along these lines, we would have to find a value function to satisfy the appropriate Hamilton-Jacobi-Isaacs inequality, given in (4.13). The system will maintain some disturbance rejection capabilities based on the choice for $\gamma$, and we explore the role of this parameter in more detail next.

## 4.4 Disturbance Attenuation Analysis

We are interested in how well the control law attenuates the effects of disturbances in the system, and the parameter $\gamma$ provides one measure of this capability. As long as we maintain $\gamma > \gamma^*$, smaller values for $\gamma$ indicate better disturbance attenuation

properties. Pan and Başar [86] prove that the controller designed using the linearization of the nonlinear system can achieve a performance level $\gamma$ if the difference between the nonlinear system and its linearization is sufficiently small. This is exactly the situation in our case if we assume that the initial configuration errors are small. Instead of relying on this valuable result to characterize the disturbance attenuation properties of the system, this section describes how variations in $\gamma$ influence the trajectory of the vehicle.

We originally thought that we would be able to use the special structure in the equations of motion to decompose the system and develop more precise results, but this approach is not possible for the state feedback control law. We can easily perform the decomposition, but the resulting subsystems are coupled and each depends on the state of the other subsystem. Since we are using state feedback, the associated Riccati equations must be solved in reverse time, starting at the final time. If we could solve the Riccati equations, we could compute the control inputs for the system and calculate the states by solving the system differential equations in forward time. We cannot execute this approach because the Riccati equations depend on the future system matrices, which in turn depend on the future error states. We cannot accurately compute the future error states without the controller, so we cannot use this design approach.

We note, however, that we will use the above decomposition technique to successfully analyze the imperfect state measurement controller design presented in Chapter 5. The approach works for the imperfect state case because the system matrices depend on the *estimated* states. The differential equations to calculate the estimated states are computed in forward time, so we can use the state estimate in the control law and do not need to know the future state estimates to find a solution.

To provide a more complete description of the disturbance attenuation properties of the state feedback controller, we will take a different approach. Recall that the differential equations for the error system are given by (4.5), where the term in square brackets is

nonlinear and $o(|\mathbf{q}_e|)$. Equation (4.5) is not an approximation, because it accounts for the nonlinear term. We make an approximation only when we drop the nonlinear term and use the linear portion to solve the associated GRDE. Suppose we rewrite (4.5) as

$$\dot{\mathbf{q}}_e = A_e(t)\mathbf{q}_e + B_e(t)\mathbf{u}_e + \tilde{D}_e(t)\tilde{\mathbf{w}}, \qquad \mathbf{q}_e(t_0) = \mathbf{q}_{e0} \tag{4.26}$$

where we define

$$\tilde{D}_e(t)\tilde{\mathbf{w}} := D_e(t)\mathbf{w} + [f_e(t, \mathbf{q}_e) - A_e(t)\mathbf{q}_e] .$$

The term $\tilde{D}_e(t)$ represents a $6 \times 5$ matrix where the upper five rows are an identity matrix and the bottom row is all zeros. We would then select $\tilde{\mathbf{w}}$ as a vector in $\mathbb{R}^5$ that accounts for the original disturbances $\mathbf{w}$ and the nonlinear terms in (4.5). We note that this representation is again an exact description of the nonlinear system. We have only changed the nature of the disturbance term to make the system appear linear.

We can now design a state feedback control law for this new representation of the system by following the approach in Section 4.2. The only adjustment we will have to make to the existing design is to change the $D_e$ matrix to $\tilde{D}_e$ in GRDE (4.8). The new design will be more conservative than the old design because we are accounting for artificial disturbances that appear because of the nonlinear terms. Relabeling the disturbances is similar to a step used by Kang [91] to develop disturbance attenuation results for a fully actuated system. This new approach allows us to precisely state the disturbance attenuation results for the state feedback design in terms of $\tilde{D}_e$ and $\tilde{\mathbf{w}}$.

Consider the linear-quadratic zero-sum differential game given by (4.26) with performance index (4.6) and with the closed-loop perfect state information structure. Let the system be defined on the time interval $[t_0, t_f]$. Replace $D_e$ by $\tilde{D}_e$ in the GRDE (4.8) and in (4.11) throughout the problem. Let $\gamma^*$ be defined by (4.9). Then, following [76], for $\gamma > \gamma^*$, the differential game admits a unique feedback saddle-point solution which

is given by (4.10) and (4.11), where $Z_\gamma(\cdot)$ is the unique solution to (4.8). In addition, the minimax attenuation level is equal to $\gamma^*$. For a particular $\bar{\gamma} > \gamma^*$, the suboptimal controller that achieves attenuation level $\bar{\gamma}$ is given by

$$\mathbf{u}^*_{\bar{\gamma}}(t, \mathbf{q}_e(t)) = -B^{\mathrm{T}}(t) Z_{\bar{\gamma}}(t) \mathbf{q}_e(t). \tag{4.27}$$

These results provide a mathematical characterization of the performance of the perfect state feedback tracking controller. Along with this rigorous description of the results, we are also interested in the qualitative disturbance attenuation properties of the system.

As we allow $\gamma$ to decrease to $\gamma^*$, the GRDE (4.8) approaches an equation with a conjugate point and the maximum eigenvalue of $Z(t)$ increases without bound. The large eigenvalue in $Z(t)$ can increase the control effort in the direction associated with the eigenvalue, which has two impacts on the system. First, the control law becomes very sensitive to errors aligned with the eigenvalue's associated eigenvector. This sensitivity means that small errors in this direction generate large control inputs. The large control inputs may overcompensate for the errors and cause oscillations in the response as these errors are eliminated. Second, in terms of the trajectory of the vehicle, the larger eigenvalue can cause the vehicle to approach the desired path more quickly because of the increased control effort.

As mentioned in the previous subsection, increasing the value of $\gamma$ decreases the role of the $\gamma^{-2} Z D_e D_e^{\mathrm{T}} Z$ term in (4.8). This term is improving the convergence rate for the system, so in general, increasing $\gamma$ will cause the vehicle to converge to the desired trajectory more slowly. From another perspective, increasing $\gamma$ moves the GRDE away from a conjugate point condition, which, for a fixed value of $t$, tends to decrease the maximum eigenvalue of $Z(t)$. The reduced maximum eigenvalues indicate that the vehicle will require more time to track the desired trajectory.

Ideally, we would like to determine the value of $\gamma^*$ for a given desired trajectory, so that we can always pick $\gamma > \gamma^*$. For the problem we have developed, there is no analytical approach to find $\gamma^*$ because of the time-varying nature of the system equations and the matrix $A_e(t)$. Even though we are working with the linearized system, the only way to find $\gamma^*$ is to select a desired trajectory and numerically simulate (4.8) for different values of $\gamma$ to search for the smallest value that provides a solution.

By recharacterizing the disturbance, we were able to make a precise statement about the disturbance attenuation properties of the linearized $H^\infty$ controller. We also made several qualitative observations about the relationship between the disturbance attenuation parameter $\gamma$ and the performance of the controller in the closed-loop system. We mentioned the possibility of proving additional disturbance attenuation properties by using the results in [86], but we will defer a more complete look at this approach until we consider the imperfect state measurement controller described in Chapter 5.

## 4.5 Nonminimum Phase Vehicle Control

The nonlinear model of an underactuated vehicle we have analyzed heretofore is minimum phase with respect to the $x$-$y$ position outputs. We will demonstrate that the motion planning and state feedback tracking control techniques developed with this model will also work when applied to a nonminimum phase system. We recall that a nonlinear system becomes nonminimum phase when the zero dynamics associated with some output are not asymptotically stable [61]. To create a nonlinear model with the nonminimum phase property, we replace Equations (1.1) through (1.3) with the following

dynamic equations:

$$\dot{u} = m_u\, v\, r - d_u\, u + u_1 + w_1 \tag{4.28}$$

$$\dot{v} = m_v\, u\, r - d_v\, v + k_v u_2 + w_2 \tag{4.29}$$

$$\dot{r} = m_r\, u\, v - d_r\, r + u_2 + w_3. \tag{4.30}$$

The kinematic equations for the new model are the same as (1.4) through (1.6). In (4.29), the parameter $k_v$ is a coupling coefficient which relates the pure torque input $u_2$ and the lateral acceleration of the vehicle. This lateral force is a natural addition to a ship model when the thrust and rudder are located at the trailing end of the vessel. Turning the rudder causes a rotation as well as a slight lateral force on the ship. We should note that the dynamic equations for the original model described in Section 1.4 can be recovered from this nonminimum phase model by setting $k_v$ to zero.

If we write the nonminimum phase equations of motion in compact form, we get

$$\dot{\mathbf{q}} = f(t, \mathbf{q}) + B_n \mathbf{u} + D\mathbf{w} \tag{4.31}$$

where $B_n$ is the new input matrix that accounts for the coupling between the rotational and lateral acceleration inputs. Note that the nonlinear function $f(t, \mathbf{q})$ and the disturbance terms do not require any changes for the new system. Since there are no changes to the general structure of the system, we can directly apply the controller design developed in Section 4.2 to the new system.

A close inspection of the basic motion planning algorithm in Section 3.2 shows that we do not need to make any changes to it to handle the nonminimum phase case. We do have the option of changing how we estimate the control input $u_2$, because it now appears in two of the equations, but it makes sense to maintain the existing approach. The change would be to use Equation (4.29) to estimate $u_2$ instead of using Equation (4.30). Since typically both $v$ and $k_v$ are small, errors in estimating $\dot{v}$ would translate into larger errors

in the estimate of $u_2$ using (4.29) than if we use (4.30). If the coupling coefficient $k_v$ was large, then we could use either (4.29) or (4.30) to estimate $u_2$ with approximately the same amount of error. The motion planning algorithm will account for the nonminimum phase behavior in the system through the iterated $H^\infty$-filter, which relies on $B_n$. In summary, to adapt the motion planning algorithm for this nonminimum phase nonlinear system, we just need to adjust the input matrix in the $H^\infty$-filtering equations.

The nonlinear function $f(t, \mathbf{q})$ has not changed, so there is also no change in the linearization of the error system, except for the input matrix $B_n$. The system has maintained the same general structure and we can use the linearized system to find the $H^\infty$-optimal controller. We solved for the linearized controller, applied it to the nonlinear system, and simulated the closed-loop behavior of the model. Chapter 6 reports on the positive results of using the state feedback tracking control algorithm with a nonminimum phase system.

The tracking controller works for both the minimum and nonminimum phase nonlinear systems because the $H^\infty$ controller is inherently robust and the differences between the two systems are relatively small. In addition, the designs fully account for the new input matrix, which captures the nonminimum phase behavior. If the coupling between the rotational and lateral accelerations was very large or if there was a different type of nonminimum phase behavior, then we might expect a decrease in the performance of the tracking controller.

From this brief analysis of a single system, we can project that the motion planning and state feedback tracking controller algorithms can accommodate nonminimum phase systems. The details of how we implement each algorithm will have to be tailored for each new system, but the general approach and the overall results should remain the same.

## 4.6 Region of Attraction Analysis

Our final investigation in this chapter considers the region of attraction for the closed-loop system with state feedback tracking control. The region of attraction is the set of initial conditions from which the tracking controller can cause the vehicle to approach the desired final configuration within the specified time interval. We have two methods for estimating the region of attraction for the nonlinear model of an underactuated vehicle. The first method uses the proof of local exponential stability presented in Section 4.3 to develop an analytical description of the region of attraction. The second approach uses numerical simulation to search a range of initial conditions to estimate the region. The following two subsections describe each approach and the results.

### 4.6.1 Analytical description

One way to make a definitive statement about the size of the region of attraction for a nonlinear system is to find a Lyapunov function for the system and a corresponding region where the derivative of the Lyapunov function is negative definite. We can then estimate the region of attraction as the largest level set of the Lyapunov function that is within the region where the function has a negative derivative. This estimate for the region of attraction is a lower bound for the actual region and may be extremely conservative. This approach may not fully describe the shape of the region, but it does provide a direct method for developing analytical results. We will present the key steps required to find a lower bound on the region of attraction for the underactuated system.

We will first identify the set of configurations that guarantee that the Lyapunov function has a negative derivative. From Section 4.3, the key inequality that must be satisfied to obtain local exponential stability is given by Equation (4.22), which we rewrite

here with slightly different notation:

$$\left|2\mathbf{q}_e^{\mathrm{T}} Z\left[f_e(t,\mathbf{q}_e) - A_e\mathbf{q}_e\right]\right| \;\leq\; \frac{1}{2}q\left|\mathbf{q}_e\right| \tag{4.32}$$

where $Z$ is the solution of the generalized Riccati differential equation for the linearized system and the matrix $Q = qI$ is part of the performance index. Recall that we are working with the error equations, which are denoted with an $e$ subscript. Our goal is to find a value of $\varepsilon$ such that $\mathbf{q}_e$ satisfies (4.32) whenever $|\mathbf{q}_e| < \varepsilon$.

To perform the analysis, we require bounds on the positive definite matrix $Z(t)$, in the form

$$0 < mI \leq Z(t) \leq MI. \tag{4.33}$$

The generalized Riccati differential equation used to find $Z(t)$ depends on the matrix $A_e(t)$, which in turn depends on the desired trajectory the system is trying to track. We cannot place a meaningful bound on $A_e(t)$ without restricting the desired trajectories for the system. If we did restrict the set of desired trajectories to get a bound on $A_e(t)$, the set would likely be too small to account for many reasonable motions. As a compromise, to complete the analytical description of the region of attraction, we will have to resort to a numerical estimate for the bounds on $Z(t)$ given by (4.33). Given a desired trajectory, we can compute $Z(t)$ and then solve for $M$ and $m$ as, respectively, the largest and smallest eigenvalues of $Z(t)$ for all $t \in [t_0, t_f]$. We will use the estimated bounds on $Z(t)$ to describe the region of attraction for the specified desired trajectory. We will have to find new values for $M$ and $m$ for each desired trajectory of interest.

To convert the inequality (4.32) into a bound on the initial condition for the system, we used the $z_i$ variables to represent the position and orientation of the vehicle instead of the $x$, $y$, and $\psi$ variables. The $z_i$ variables simplify the equations of motion and allow us to perform some cancellations to reach a compact expression for the region of attraction.

Using (4.33), we can show that (4.32) is equivalent to

$$\left|\mathbf{q}_e^{\mathrm{T}}\left[f_e(t,\mathbf{q}_e) - A_e\mathbf{q}_e\right]\right| \;\leq\; \frac{q}{4M}\left|\mathbf{q}_e\right|^2. \tag{4.34}$$

With the $z_i$ variables, we have

$$\left[f_e(t,\mathbf{q}_e) - A_e\mathbf{q}_e\right] = \begin{bmatrix} m_u v_e r_e & m_v u_e r_e & m_r u_e v_e & z_{2e} r_e & -z_{1e} r_e & 0 \end{bmatrix}^{\mathrm{T}}$$

where $m_u$, $m_v$, and $m_r$ are the mass parameters in the equations of motion. Expanding both sides of (4.34), we get

$$\left|(m_u + m_v + m_r) u_e v_e r_e\right| \;\leq\; \frac{q}{4M}\left|u_e^2 + v_e^2 + r_e^2 + z_{1e}^2 + z_{2e}^2 + z_{3e}^2\right|$$

which leads to

$$\left|u_e v_e r_e\right| \;\leq\; \frac{q}{4M\left|m_u + m_v + m_r\right|}\left|u_e^2 + v_e^2 + r_e^2 + z_{1e}^2 + z_{2e}^2 + z_{3e}^2\right|. \tag{4.35}$$

Suppose we restrict $|\mathbf{q}_e| \leq \varepsilon < 1$, for some $\varepsilon > 0$, and then we maximize the left-hand side of (4.35) subject to $|\mathbf{q}_e| \leq \varepsilon$. The maximum occurs when $u_e = v_e = r_e = \frac{\varepsilon}{\sqrt{3}}$. Using the constraint on $\mathbf{q}_e$, we also have $|\mathbf{q}_e|^2 \leq \varepsilon^2$, so we can rewrite (4.35) as

$$\frac{\varepsilon^3}{3\sqrt{3}} \;\leq\; \frac{q\varepsilon^2}{4M\left|m_u + m_v + m_r\right|}$$

which simplifies to

$$\varepsilon \;\leq\; \frac{3q\sqrt{3}}{4M\left|m_u + m_v + m_r\right|}. \tag{4.36}$$

We now set $\varepsilon$ equal to the right-hand side of (4.36), because it is the largest value that allows us to claim that $\mathbf{q}_e$ satisfies (4.34) whenever $|\mathbf{q}_e| \leq \varepsilon$. Therefore, the region defined by

$$|\mathbf{q}_e| \;\leq\; \frac{3q\sqrt{3}}{4M\left|m_u + m_v + m_r\right|} \;=:\; q_{\max} \tag{4.37}$$

is a conservative estimate for the region of initial conditions that guarantee that the Lyapunov function will maintain a negative derivative along the associated trajectory. We will now find a level set for the Lyapunov function within this region.

A level set is the collection of configurations such that we can place an upper bound on the value of the Lyapunov function. We recall that the Lyapunov function is $V(\mathbf{q}_e) = \mathbf{q}_e^{\mathrm{T}} Z \mathbf{q}_e$ and define the level set as

$$\Omega_c := \{\mathbf{q}_e \in I\!R^6 \mid V(\mathbf{q}_e) < c\} \tag{4.38}$$

where $c$ is a parameter describing the level. Using the lower bound in (4.33) and (4.37) and following the standard arguments in [71], we can select

$$c \;<\; \min_{|\mathbf{q}_e| = q_{\max}} V(\mathbf{q}_e) \;=\; \frac{27mq^2}{16M^2\,|m_u + m_v + m_r|}.$$

With this value for $c$, the region defined by $\Omega_c$ provides an estimate for the region of attraction for the system and describes the set of initial conditions that will generate trajectories approaching the goal configuration at the required time.

We performed three simulations of typical motions to determine bounds on the matrix $Z(t)$ and found that choosing $m = 0.2$ and $M = 20$ would be sufficient in all three cases. We also chose $q = 1$ and picked values for $m_u$, $m_v$, and $m_r$ such that $|m_u + m_v + m_r| = 1$. These desired trajectories and parameter choices result in a numerical value of $\varepsilon = 0.065$ and $c = 0.0008$. These values for $\varepsilon$ and $c$ indicate that the initial error magnitude must be very small to guarantee that the tracking controller will cause the vehicle to approach the desired final configuration. The estimate is extremely conservative because we are using the worst-case bound on the matrix $Z(t)$. As the vehicle moves, the upper bound on $Z(t)$ will often be less than $MI$, so using the maximum upper bound contributes to the conservative nature of the estimate. To find a more accurate estimate for the region of attraction, we will use a numerical approach.

### 4.6.2 Numerical description

The numerical approach for estimating the region of attraction is a brute force technique that scans a set of initial conditions and determines which of them lead to trajectories that approach the desired final configuration. To reduce the dimension of the search space, we assumed the desired trajectory generated by the motion planning algorithm had no initial position error. This assumption is valid when we use the version of the motion planning algorithm described in Subsection 3.2.3 that generates the desired trajectory starting from the *estimated* initial conditions. With no error in the position variables, we only have to vary the remaining four states to characterize the region of attraction, which significantly reduces the computational requirements.

We applied the motion planning algorithm starting at $\mathbf{q} = [1,\ 0,\ 0,\ 0,\ 0,\ 0]^{\mathrm{T}}$ and moving to $[1,\ 0,\ 0,\ 8,\ 8,\ 0]^{\mathrm{T}}$, to generate the desired trajectory for the first set of simulations. Using this desired trajectory, we computed $Z(t)$ for the entire time interval. We then selected an initial condition and applied the state feedback control law to simulate the motion of the vehicle. With this desired trajectory, we performed three simulations with different sets of initial conditions. The initial conditions were

1. $u_e(t_0),\ v_e(t_0),\ r_e(t_0) \in [-2,\ -1.5,\ -1.0,\ -0.5,\ 0.0,\ 0.5,\ 1.0,\ 1.5,\ 2.0]$,
   $\psi_e(t_0) \in [-\pi,\ -3\pi/4,\ -\pi/2,\ -\pi/4,\ 0,\ \pi/4,\ \pi/2,\ 3\pi/4,\ \pi]$.

2. $u_e(t_0),\ v_e(t_0),\ r_e(t_0) \in [-1,\ -0.8,\ -0.6,\ -0.4,\ -0.2,\ 0.0,\ 0.2,\ 0.4,\ 0.6,\ 0.8,\ 1.0]$,
   $\psi_e(t_0) \in [-\pi,\ -4\pi/5,\ -3\pi/5,\ -2\pi/5,\ -\pi/5,\ 0,\ \pi/5,\ 2\pi/5,\ 3\pi/5,\ 4\pi/5,\ \pi]$.

3. $u_e(t_0),\ v_e(t_0),\ r_e(t_0) \in [-0.5,\ -0.4,\ -0.3,\ -0.2,\ -0.1,\ 0.0,\ 0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5]$,
   $\psi_e(t_0) \in [-\pi/2,\ -2\pi/5,\ -3\pi/10,\ -\pi/5,\ -\pi/10,\ 0,\ \pi/10,\ \pi/5,\ 3\pi/10,\ 2\pi/5,\ \pi/2]$.

The first two sets of initial conditions led to trajectories that did not converge to the desired final configuration. However, in the third set, all of the initial conditions generated

trajectories that approached the desired configuration. We can thus conclude that for this desired trajectory and within the resolution of this numerical computation, the region of attraction is between the second and third sets of initial conditions. We can therefore use the third set of initial conditions to form a conservative estimate for the region of attraction. Although this estimate is conservative, it is an order of magnitude larger than the analytical estimate for the region of attraction developed in Subsection 4.6.1.

To further support these results, we checked the third set of initial conditions given above with the two other desired trajectories considered in the analytical estimate for the region of attraction. In both cases, all of the initial conditions in the third set generated trajectories that converged to the desired final configuration.

The numerical approach for estimating the region of attraction is inherently limited because we can only survey a small number of initial conditions and desired trajectories within a reasonable amount of time. Improving the resolution of the description of the region of attraction or finding its exact boundary are computationally expensive tasks that may not be worth the effort. The region of attraction described by the third set of initial conditions is reasonably large for the types of simulations we have considered and indicates that the state feedback tracking controller provides a practical solution to the nonlinear underactuated vehicle control problem in many situations.

## 4.7 Perfect State Feedback Summary

The perfect state feedback controller design provides the initial approach for making an underactuated vehicle track a feasible desired trajectory while rejecting the effect of disturbances. The approach relies on linearization and $H^\infty$ design techniques to develop the controller. We have demonstrated that the closed-loop system achieves local exponential stability, and we have characterized the disturbance attenuation properties of the

system. In addition, we have explained how the approach can apply to nonminimum phase systems and used two methods to estimate the region of attraction for the closed-loop system. As will be illustrated in Chapter 6, the approach provides exceptionally good tracking performance when the full state is available for feedback. When the model does not allow us to access the entire state for feedback, we need another approach to develop the controller. We will address this imperfect state measurement case in the next chapter (Chapter 5) by using a state estimate in the state feedback control law.

# CHAPTER 5

# IMPERFECT STATE MEASUREMENT TRACKING CONTROL

The controller developed in Chapter 4 relies on perfect state measurements for the feedback law. In a realistic situation, the entire state for the system cannot generally be measured. In our example of an underactuated ship, we can expect to directly measure the position and orientation of the vehicle, but not the velocities. In addition, the position and orientation measurements will likely be corrupted by disturbances, which we will have to account for in the analysis. We will now modify the controller designed in the previous chapter to handle the imperfect state measurement case. We still rely on the $H^\infty$ design tools to obtain a controller, and will exploit an existing unique structure in the model of the underactuated vehicle to achieve the desired results. The approach will allow us to prove a modified version of disturbance attenuation for the closed-loop system.

## 5.1 Problem Formulation

The problem formulation for this chapter is similar to that for Chapter 4. As before, we want to determine the control inputs to cause the underactuated vehicle to track a feasible trajectory and attenuate the effect of disturbances. The control law will have access only to imperfect measurements of the position and orientation states. The feasible

trajectory will be constructed in advance and the controller will be able to use the inputs that generated the trajectory.

We will use the same equations for the actual, desired, and error systems as presented in the previous chapter in (4.1) through (4.3). We note that the error equation (4.3) has an equilibrium point at the origin when there are no disturbances present. We introduce the measurement equation

$$\mathbf{y}(t) = C(t)\mathbf{q}(t) + E(t)\mathbf{w}(t) \tag{5.1}$$

where $\mathbf{y} \in I\!R^q$ and $C(t)$ and $E(t)$ are matrices of the appropriate dimensions. Our approach will be to use the measurements $\mathbf{y}(t)$ to estimate the full state of the system $\mathbf{q}(t)$ and then substitute the estimate for the actual state in the feedback controller developed in Equation (4.12). We will denote the estimate for the state as $\hat{\mathbf{q}}$.

## 5.2 $H^\infty$-Filter Design Using System Decomposition

The first step in this design process is to use the structure of the model for the underactuated vehicle to write the equations of motion in a more convenient format. The nonlinear error equations for the vehicle can be decomposed into two sets of equations, where each equation is affine if the state of the other subsystem is known. For this analysis, we will suppress the $e$ subscript on the error equations. The equations for the two subsystems are given as

$$\dot{\mathbf{q}}_1 = A_{11}(\mathbf{q}_2)\mathbf{q}_1 + \alpha_1(\mathbf{q}_2) + B_1\mathbf{u}_1 + D_1\mathbf{w}_1 \tag{5.2}$$

$$\dot{\mathbf{q}}_2 = A_{22}(\mathbf{q}_1)\mathbf{q}_2 + \alpha_2(\mathbf{q}_1) + B_2\mathbf{u}_2 + D_2\mathbf{w}_2 \tag{5.3}$$

where

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix}, \qquad \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}, \qquad \mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}. \tag{5.4}$$

For our model, we have

$$\mathbf{q}_1 = \begin{bmatrix} u & v & x & y \end{bmatrix}^{\mathrm{T}}, \qquad \mathbf{q}_2 = \begin{bmatrix} r & \psi \end{bmatrix}^{\mathrm{T}}$$

$$\mathbf{u}_1 = u_1, \qquad \mathbf{u}_2 = u_2$$

$$\mathbf{w}_1 = \begin{bmatrix} w_1 & w_2 \end{bmatrix}^{\mathrm{T}}, \qquad \mathbf{w}_2 = w_3.$$

The $\alpha_i(\mathbf{q}_j)$ terms are nonlinear and we would like to identify a first-order approximation for each term to further simplify (5.2) and (5.3). To do so, we take the Jacobian of the $\alpha_i(\mathbf{q}_j)$ terms with respect to $\mathbf{q}_j$ to identify the linear portions

$$A_{12} = \left.\frac{\partial \alpha_1(\mathbf{q}_2)}{\partial \mathbf{q}_2}\right|_{\mathbf{q}_2=0}, \qquad A_{21} = \left.\frac{\partial \alpha_2(\mathbf{q}_1)}{\partial \mathbf{q}_1}\right|_{\mathbf{q}_1=0}.$$

We now rewrite the subsystem equations as

$$\dot{\mathbf{q}}_1 = A_{11}(\mathbf{q}_2)\mathbf{q}_1 + A_{12}\mathbf{q}_2 + [\alpha_1(\mathbf{q}_2) - A_{12}\mathbf{q}_2] + B_1\mathbf{u}_1 + D_1\mathbf{w}_1 \tag{5.5}$$

$$\dot{\mathbf{q}}_2 = A_{22}(\mathbf{q}_1)\mathbf{q}_2 + A_{21}\mathbf{q}_1 + [\alpha_2(\mathbf{q}_1) - A_{21}\mathbf{q}_1] + B_2\mathbf{u}_2 + D_2\mathbf{w}_2. \tag{5.6}$$

Combining the subsystem equations, we arrive at

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \end{bmatrix} = \begin{bmatrix} A_{11}(\mathbf{q}_2) & A_{12} \\ A_{21} & A_{22}(\mathbf{q}_1) \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \end{bmatrix} + \begin{bmatrix} \alpha_1(\mathbf{q}_2) - A_{12}\mathbf{q}_2 \\ \alpha_2(\mathbf{q}_1) - A_{21}\mathbf{q}_1 \end{bmatrix} + \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} + \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix}$$

which we write compactly as

$$\dot{\mathbf{q}} = A(\mathbf{q})\mathbf{q} + \alpha(\mathbf{q}) + B\mathbf{u} + D\mathbf{w}. \tag{5.7}$$

We note for future reference that $\alpha(0) = 0$ in the systems under consideration. This property indicates that the transformations have not disturbed the equilibrium point at

the origin for the error equations. We are going to estimate the state $\mathbf{q}$ using $\mathbf{y}$ so we rewrite (5.7) as

$$\dot{\mathbf{q}} = A(\hat{\mathbf{q}})\mathbf{q} + \alpha(\hat{\mathbf{q}}) + B\mathbf{u} + D\mathbf{w} + [A(\mathbf{q})\mathbf{q} - A(\hat{\mathbf{q}})\mathbf{q} + \alpha(\mathbf{q}) - \alpha(\hat{\mathbf{q}})] \tag{5.8}$$

where the estimate $\hat{\mathbf{q}}$ depends on past measurements $y_{[t_0,t]} := \{y(\tau) \ : \ \tau \in [t_0, t]\}$.

We can consider (5.8) as an affine system with time-varying elements and a nonlinear perturbation term in square brackets. We would like to derive results similar to those by Pan and Başar [86] to establish attenuation of disturbances for this system. To mimic the problem presented in [86], we rewrite (5.8) as

$$\dot{\mathbf{q}} = A(\hat{\mathbf{q}})\mathbf{q} + \alpha(\hat{\mathbf{q}}) + B\mathbf{u} + D\mathbf{w} + \varepsilon\left[a(t, \mathbf{q}) + b(t, \mathbf{q})\mathbf{u} + d(t, \mathbf{q})\mathbf{w}\right]. \tag{5.9}$$

Equation (5.9) is similar to Equation (1a) in [86], with the difference being that (5.9) contains the affine term $\alpha(\hat{\mathbf{q}})$. We can develop a disturbance attenuating controller with the affine term in the equations if we make one assumption and minor modifications to the derivation in [86]. Section 5.3 presents the controller design and outlines a proof of disturbance attenuation for the closed-loop system.

## 5.3 Disturbance Attenuating Controller

Our objective is to find a control law based on the imperfect state measurements in (5.1) that achieves disturbance attenuation level $\gamma$ for the nonlinear system (5.7). If (5.9) did not have the affine term, we could apply the results in [86] almost directly to get the desired performance level. The affine term in (5.9) creates one difficulty that must be addressed to reach a meaningful conclusion about disturbance attenuation. Due to the presence of the affine term, we cannot prove the standard disturbance attenuation results, but we can find a finite bound for the upper value of the associated soft-constrained

dynamic game. This upper bound allows us to prove a modified form of disturbance attenuation. We now present the controller equations that solve this problem and describe how they achieve a specified level of disturbance attenuation.

We form the candidate control law by replacing the actual state in Equation (4.10) with the estimate for the state $\hat{\mathbf{q}}$ to get

$$\mu(t, \mathbf{y}_{[t_0,t]}) = -B^{\mathrm{T}}(t)Z(t)\hat{\mathbf{q}} \tag{5.10}$$

where $\hat{\mathbf{q}}$ is generated by

$$\dot{\hat{\mathbf{q}}} = [A - (BB^{\mathrm{T}} - \gamma^{-2}DD^{\mathrm{T}})Z]\hat{\mathbf{q}} + \alpha(\hat{\mathbf{q}}) + [I - \gamma^{-2}\Sigma Z]^{-1}\Sigma C^{\mathrm{T}}N^{-1}(\mathbf{y} - C\hat{\mathbf{q}}), \;\; \hat{\mathbf{q}}(t_0) = 0 \tag{5.11}$$

with $\Sigma$ satisfying the following $H^\infty$-filter error covariance equation:

$$\dot{\Sigma} = A\Sigma + \Sigma A^{\mathrm{T}} - \Sigma(C^{\mathrm{T}}N^{-1}C - \gamma^{-2}Q)\Sigma + DD^{\mathrm{T}}, \quad \Sigma(t_0) = [Q_0 - \eta I]^{-1} \tag{5.12}$$

where $Q(t) \geq 0$, $Q_0 - \eta I > 0$, $\eta > 0$, and $N(t) := E(t)E^{\mathrm{T}}(t) > 0$. The performance index associated with the imperfect state measurement case is

$$\begin{aligned} L(\mathbf{u}, \mathbf{w}, \mathbf{q}_0; \varepsilon) \;\; = \;\; & \int_{t_0}^{t_f} \left[ |\mathbf{q}(t)|^2_Q + |\mathbf{u}(t)|^2 + \varepsilon \left( q(t, \mathbf{q}(t)) + \mathbf{u}^{\mathrm{T}}(t) r(t, \mathbf{q}(t)) \mathbf{u}(t) \right) \right] dt \\ & + |\mathbf{q}(t_f)|^2_{Q_f}. \end{aligned} \tag{5.13}$$

We will assume that the initial state is unknown and treat it as part of the disturbance. The associated zero-sum differential game has the following soft-constrained cost function:

$$L_\gamma(\mathbf{u}, \mathbf{w}, \mathbf{q}_0; \varepsilon) = L(\mathbf{u}, \mathbf{w}, \mathbf{q}_0; \varepsilon) - \gamma^2 \left( |\mathbf{q}_0|^2_{Q_0} + \|\mathbf{w}\|^2 \right). \tag{5.14}$$

We apply control law (5.10) to (5.9) and to the performance index (5.14), and obtain equations similar to Equations (37) and (38) in [86]. For brevity, we repeat the short

versions of these two equations here:

$$\dot{\bar{\mathbf{q}}} = F(t)\bar{\mathbf{q}} + \beta(t) + G(t)\mathbf{w} + \varepsilon\left[f(t,\bar{\mathbf{q}}) + g(t,\bar{\mathbf{q}})\mathbf{w}\right], \quad \bar{\mathbf{q}}(t_0) = \begin{bmatrix} x_0 \\ 0 \end{bmatrix}, \tag{5.15}$$

$$\begin{aligned} L_\gamma^* &= -\gamma^2 \bar{\mathbf{q}}_0^{\mathrm{T}} \bar{Q}_0 \bar{\mathbf{q}}_0 + \bar{\mathbf{q}}^{\mathrm{T}}(t_f)\bar{Q}_f \bar{\mathbf{q}}(t_f) \\ &\quad + \int_{t_0}^{t_f} \left[\bar{\mathbf{q}}^{\mathrm{T}} H(t)\bar{\mathbf{q}} - \gamma^2 \mathbf{w}^{\mathrm{T}}\mathbf{w} + \varepsilon h(t,\bar{\mathbf{q}})\right] dt \end{aligned} \tag{5.16}$$

where $\bar{\mathbf{q}} := [\mathbf{q}^{\mathrm{T}}, \hat{\mathbf{q}}^{\mathrm{T}}]^{\mathrm{T}}$. In (5.9) the $A(\hat{\mathbf{q}})$ and $\alpha(\hat{\mathbf{q}})$ terms depend on $\hat{\mathbf{q}}$, which implies that the terms $F(t)$ and $\beta(t)$ in (5.15) depend on $\hat{\mathbf{q}}$. We observe that $\hat{\mathbf{q}}$ depends on $\mathbf{y}$, which in turn depends on $\mathbf{w}$, so $\mathbf{w}$ enters into the expression for $\bar{\mathbf{q}}$ through $F(t)$ and $\beta(t)$. We will shortly perform a maximization with respect to $\mathbf{w}$ and $\mathbf{q}_0$, which will have to account for this relationship. We make the following assumption to address this aspect of the problem.

**Assumption 5.1 (A5.1)** *The effect of the disturbance term* $\mathbf{w}$ *on the terms* $F(t)$ *and* $\beta(t)$ *is small enough that the upper value of the performance index* $L_\gamma^*$ *remains bounded.*

Our concern with the effect of $\mathbf{w}$ on $\bar{\mathbf{q}}$ stems from the performance index (5.16), where $\bar{\mathbf{q}}$ makes a positive contribution to the cost. For a fixed value of $\gamma$, if $\mathbf{w}$ has a large influence on $\bar{\mathbf{q}}$, it would be possible for the disturbance to choose a value to make the integrand in the cost function arbitrarily large. On the other hand, if $\mathbf{w}$ has a negligible effect on $\bar{\mathbf{q}}$, then the $-\gamma^2\mathbf{w}^{\mathrm{T}}\mathbf{w}$ term will dominate any positive contribution $\mathbf{w}$ makes through $\bar{\mathbf{q}}^{\mathrm{T}} H(t)\bar{\mathbf{q}}$, and the upper value of $L_\gamma^*$ will remain bounded.

Assumption A5.1 is reasonable if a large change in the disturbance does not cause a significant change in the values of $F(t)$ and $\beta(t)$. In other words, $F(t)$ and $\beta(t)$ should be relatively insensitive to $\mathbf{w}$. If this is the case, then a large value of $\mathbf{w}$ will affect $\bar{\mathbf{q}}$ primarily through the $G(t)\mathbf{w}$ term and not the $F(t)$ or $\beta(t)$ terms. (Note that the optimization problem will account for the role of $G(t)\mathbf{w}$ through a Riccati differential

equation to be presented shortly.) Equivalently, if the role of $\mathbf{w}$ is small enough that it can be modeled as part of the $\varepsilon$ disturbance in square brackets in (5.15), then we can find a value for $\gamma$ to keep the upper value of the performance index finite. To further support this assumption, the simulation results in Chapter 6 will demonstrate for the model system that changes in $\mathbf{w}$ have a small effect on $F(t)$ and $\beta(t)$ and, hence, there still exists a finite upper bound for performance index $L_\gamma^*$.

With Assumption A5.1 in place, we now want to find an upper bound for the value function of the maximization problem

$$\sup_{\mathbf{q}_0 \in I\!R^n} \sup_{\mathbf{w} \in \mathcal{H}_w} L_\gamma^*(\mathbf{w}, \bar{\mathbf{q}}_0) \tag{5.17}$$

subject to the constraint (5.15) and with $\mathcal{H}_w$ representing the class of admissible disturbances.[1] We start by separating the supremum operations to get

$$\begin{aligned} \sup_{\mathbf{q}_0 \in I\!R^n} \sup_{\mathbf{w} \in \mathcal{H}_w} L_\gamma^*(\mathbf{w}, \bar{\mathbf{q}}_0) \;\; \leq \;\; & \sup_{\mathbf{q}_0 \in I\!R^n} \Bigg\{ -\gamma^2 \bar{\mathbf{q}}_0^{\mathrm{T}} \bar{Q}_0 \bar{\mathbf{q}}_0 \\ & + \sup_{\mathbf{w} \in \mathcal{H}_w} \left\{ \bar{\mathbf{q}}^{\mathrm{T}}(t_f) \bar{Q}_f \bar{\mathbf{q}}(t_f) + \int_{t_0}^{t_f} \left[ \bar{\mathbf{q}}^{\mathrm{T}} H(t) \bar{\mathbf{q}} - \gamma^2 \mathbf{w}^{\mathrm{T}} \mathbf{w} + \varepsilon h(t, \bar{\mathbf{q}}) \right] dt \right\} \Bigg\}. \end{aligned} \tag{5.18}$$

Following Pan and Başar [86], we focus on the inner maximization and define

$$\widetilde{L}_\gamma(\mathbf{w}, \bar{\mathbf{q}}_0) := \sup_{\mathbf{w} \in \mathcal{H}_w} \left\{ \bar{\mathbf{q}}^{\mathrm{T}}(t_f) \bar{Q}_f \bar{\mathbf{q}}(t_f) + \int_{t_0}^{t_f} \left[ \bar{\mathbf{q}}^{\mathrm{T}} H(t) \bar{\mathbf{q}} - \gamma^2 \mathbf{w}^{\mathrm{T}} \mathbf{w} + \varepsilon h(t, \bar{\mathbf{q}}) \right] dt \right\}. \tag{5.19}$$

If $\varepsilon = 0$, this problem admits a maximum $\bar{\mathbf{q}}^{\mathrm{T}}(t_0) \Xi(t_0) \bar{\mathbf{q}}(t_0) + \zeta^{\mathrm{T}}(t_0) \bar{\mathbf{q}}(t_0) + m(t_0)$ where $\Xi(t)$ is the nonnegative definite solution to the following GRDE:

$$\dot{\Xi} + F^{\mathrm{T}} \Xi + \Xi F + \gamma^{-2} \Xi G G^{\mathrm{T}} \Xi + H = 0, \quad \Xi(t_f) = \bar{Q}_f \tag{5.20}$$

and $\zeta(t)$ and $m(t)$ satisfy the following differential equations:

$$\dot{\zeta} + 2\Xi\beta + F^{\mathrm{T}} \zeta + \gamma^{-2} \Xi G G^{\mathrm{T}} \zeta = 0, \quad \zeta(t_f) = 0 \tag{5.21}$$

$$\dot{m} + \zeta^{\mathrm{T}} \beta + \frac{1}{4\gamma^2} \zeta^{\mathrm{T}} G G^{\mathrm{T}} \zeta = 0, \quad m(t_f) = 0. \tag{5.22}$$

[1] Here, $\mathcal{H}_w$ is the Hilbert space of square-integrable functions on $[t_0, t_f]$, taking values in $I\!R^p$.

These last two equations are a result of the affine term $\beta$ in the constraint equation for the maximization problem. To demonstrate a disturbance attenuation level of $\gamma$, we follow the proof in [86] and consider the following GRDE when $\varepsilon > 0$:

$$\dot{\Xi}_\delta + F^{\mathrm{T}}\Xi_\delta + \Xi_\delta F + \gamma^{-2}\Xi_\delta G G^{\mathrm{T}}\Xi_\delta + H + \delta I = 0, \quad \Xi_\delta(t_f) = \bar{Q}_f \tag{5.23}$$

where $\delta$ is a small positive scalar. Equation (5.23) has a solution for sufficiently small $\delta > 0$. Let $\zeta_\delta$ and $m_\delta$ be the solutions to (5.21) and (5.22), respectively, with $\Xi$ replaced by $\Xi_\delta$. Partition $\Xi_\delta$ and $\zeta_\delta$ as

$$\Xi_\delta(t) = \begin{bmatrix} \Xi_{\delta 11}(t) & \Xi_{\delta 12}(t) \\ \Xi_{\delta 21}(t) & \Xi_{\delta 22}(t) \end{bmatrix}, \qquad \zeta_\delta(t) = \begin{bmatrix} \zeta_{\delta 1}(t) \\ \zeta_{\delta 2}(t) \end{bmatrix}.$$

Skipping some of the details in the paper [86], we can show that

$$\Xi_{\delta 11}(t_0) \leq \gamma^2\Sigma^{-1}(t_0) + \gamma^2\eta I = \gamma^2 Q_0. \tag{5.24}$$

For a fixed $\delta > 0$, the function $\bar{\mathbf{q}}^{\mathrm{T}}\Xi_\delta(t)\bar{\mathbf{q}}$ is an upper bound for $\bar{\mathbf{q}}^{\mathrm{T}}\Xi(t)\bar{\mathbf{q}}$ in the maximization problem (5.17). This result implies

$$\begin{aligned} &\sup_{\mathbf{w}\in\mathcal{H}_w} \left\{ \bar{\mathbf{q}}^{\mathrm{T}}(t_f)\bar{Q}_f\bar{\mathbf{q}}(t_f) + \int_{t_0}^{t_f} \left[ \bar{\mathbf{q}}^{\mathrm{T}}H(t)\bar{\mathbf{q}} - \gamma^2\mathbf{w}^{\mathrm{T}}\mathbf{w} + \varepsilon h(t,\bar{\mathbf{q}}) \right] dt \right\} \\ &\quad \leq \ \bar{\mathbf{q}}^{\mathrm{T}}(t_0)\Xi_\delta(t_0)\bar{\mathbf{q}}(t_0) + \zeta_\delta^{\mathrm{T}}(t_0)\bar{\mathbf{q}}(t_0) + m_\delta(t_0). \end{aligned} \tag{5.25}$$

We now consider the complete maximization problem in (5.18) to get

$$\begin{aligned} \sup_{\mathbf{q}_0\in\mathbb{R}^n} \sup_{\mathbf{w}\in\mathcal{H}_w} L_\gamma^* \ &\leq \ \sup_{\mathbf{q}_0\in\mathbb{R}^n} \left\{ -\gamma^2\bar{\mathbf{q}}_0^{\mathrm{T}}\bar{Q}_0\bar{\mathbf{q}}_0 + \bar{\mathbf{q}}_0^{\mathrm{T}}\Xi_\delta(t_0)\bar{\mathbf{q}}_0 + \zeta_\delta^{\mathrm{T}}(t_0)\bar{\mathbf{q}}_0 + m_\delta(t_0) \right\} \\ &= \ \sup_{\mathbf{q}_0\in\mathbb{R}^n} \left\{ -\gamma^2\mathbf{q}_0^{\mathrm{T}}Q_0\mathbf{q}_0 + \mathbf{q}_0^{\mathrm{T}}\Xi_{\delta 11}(t_0)\mathbf{q}_0 + \zeta_{\delta 1}^{\mathrm{T}}(t_0)\mathbf{q}_0 + m_\delta(t_0) \right\} \\ &= \ \frac{1}{4}\zeta_{\delta 1}^{\mathrm{T}}(t_0)\left[\gamma^2 Q_0 - \Xi_{\delta 11}(t_0)\right]^{-1}\zeta_{\delta 1}(t_0) + m_\delta(t_0). \end{aligned} \tag{5.26}$$

At this point, we cannot make any statements about the disturbance attenuation level for this system because the right-hand side of (5.26) could be greater than zero. We know

that the right-hand side of (5.26) is bounded because (5.21) and (5.22) lead to bounded values for $\zeta_{\delta 1}(t_0)$ and $m_\delta(t_0)$. Let $M > 0$ represent an upper bound for the right-hand side of (5.26). Then,

$$\begin{aligned} \sup_{\mathbf{q}_0 \in I\!R^n} \sup_{\mathbf{w} \in \mathcal{H}_w} L^*_\gamma &\leq \frac{1}{4}\zeta^{\mathrm{T}}_{\delta 1}(t_0)\left[\gamma^2 Q_0 - \Xi_{\delta 11}(t_0)\right]^{-1}\zeta_{\delta 1}(t_0) + m_\delta(t_0) \leq M \\ \Rightarrow \quad \sup_{\mathbf{q}_0 \in I\!R^n} \sup_{\mathbf{w} \in \mathcal{H}_w} &\left\{L^* - \gamma^2\left(|\mathbf{q}_0|^2_{Q_0} + \|\mathbf{w}\|^2\right)\right\} \leq M. \end{aligned} \tag{5.27}$$

Inequality (5.27) implies that the difference between $L^*$ and $\gamma^2\left(|\mathbf{q}_0|^2_{Q_0} + \|\mathbf{w}\|^2\right)$ is always bounded by a fixed value. This translates into a condition relating the rates of growth for $L^*$ and the disturbance terms. The costs on the state plus control cannot grow at a faster rate than the disturbances. This result is a modified form of disturbance attenuation for the nonlinear system. We note that it is possible that the right-hand side of (5.26) could have an upper bound of zero, in which case we achieve disturbance attenuation level $\gamma$, using the standard meaning of the phrase. (Recall that $\alpha(0) = 0$, which implies $\beta(t_0) = 0$. If $\beta(t) \equiv 0$, then the affine term vanishes, Equation (5.15) becomes linear, and the maximization preceding (5.26) becomes negative by construction, as shown in [86].) Since, in general, we have to account for the affine terms which are sign indefinite, we can only guarantee the modified form of disturbance attenuation.

The portion of the proof that shows that we cannot find a controller to achieve a performance level of $\gamma$ for $\gamma < \gamma^*(0)$ is similar to that presented in [86] and is not repeated here.

Our analysis above has thus designed a control law for the imperfect state measurement case and proven a form of local disturbance attenuation for the closed-loop system. We will simulate the system with this controller in Chapter 6 to verify its effectiveness as a tracking controller.

## 5.4 Analysis of Controller Design

We now provide in this section a more detailed analysis of the imperfect state measurement controller in terms of its disturbance attenuation properties. In addition, this section introduces the problem of proving local exponential stability for the closed-loop system using a separation principle argument. Although we are able to elaborate on the ability of the system to attenuate disturbances, we will leave the establishment of local stability results as an open research problem.

The imperfect state measurement tracking controller depends on the solution to two generalized Riccati differential equations [one for $Z(t)$ and one for $\Sigma(t)$] and one ordinary differential equation for the estimate of the state $\hat{\mathbf{q}}(t)$. We selected this approach because of the disturbance attenuation characteristics and the inherent robustness properties that arise from $H^\infty$ techniques. In addition, the approach provides a direct method for computing the control law. The resulting controller is based on a linearization of the original system, so we can prove only local disturbance attenuation results for the closed-loop system.

One way to provide a more complete description of the disturbance attenuation properties of the system is to solve the Hamilton-Jacobi inequalities associated with the full nonlinear time-varying problem. A general analytical solution for these inequalities is not readily available for the types of systems we are considering, and in fact may be impossible to obtain. Numerical solutions may certainly be possible, but we decided not to pursue this computationally expensive option.

Another way to characterize the disturbance attenuation results is to rewrite (5.8) as

$$\dot{\mathbf{q}} = A(\hat{\mathbf{q}})\mathbf{q} + \alpha(\hat{\mathbf{q}}) + B\mathbf{u} + \tilde{D}\tilde{\mathbf{w}} \tag{5.28}$$

where

$$\tilde{D}\tilde{\mathbf{w}} = D\mathbf{w} + \left[A(\mathbf{q})\mathbf{q} - A(\hat{\mathbf{q}})\mathbf{q} + \alpha(\mathbf{q}) - \alpha(\hat{\mathbf{q}})\right].$$

This representation parallels the one presented in Section 4.4 and allows us to show that the design leads to an optimal disturbance attenuating controller with respect to disturbance $\tilde{\mathbf{w}}$. As mentioned before, this approach leads to a more conservative controller because the new disturbance $\tilde{\mathbf{w}}$ is accounting for the remaining nonlinearities in the system. However, this characterization offers a simple alternative to the more complete description of the disturbance attenuation properties presented in Section 5.2.

A second reasonable goal for the imperfect state measurement controller design, which we will not pursue here, is to prove local exponential tracking for the nonlinear system using a separation principle. The separation principle allows us to individually design a state feedback controller and a state estimator, and then combine the results to form an output feedback control law. The separation principle holds in general for linear systems, but it must be demonstrated for each nonlinear system. We have shown that the perfect state feedback tracking controller is locally exponentially stable, which is the first step in proving the results for the imperfect state measurement case. We have also shown that the closed-loop imperfect state measurement controller achieves local disturbance attenuation, which may be helpful in establishing the separation principle.

To complete the proof of local exponential tracking, we suggest advancing the recent results on cascaded systems by Loria, Fossen, and Panteley [98] to develop a separation principle for nonlinear underactuated systems. There are two key challenges with the approach in [98] that must be resolved in order to apply the results to our problem. First, the authors designed a controller for a fully actuated system, so the results will have to be modified to handle the underactuated case. The second challenge is that the control law in [98] is a proportional-integral-derivative controller, which greatly simplifies

the analysis of the closed-loop system. The system using the $H^\infty$ controller is more mathematically involved, because it relies on coupled differential equations to compute the control law. Even though the computations are more involved for the $H^\infty$ case, the approach is well understood and there are techniques to analyze the controller, which should lead to a separation principle for the nonlinear underactuated system. Having formulated this problem and outlined one solution approach, we plan to leave this issue as an open research area for future work.

## 5.5 Imperfect State Measurement Summary

This chapter constructed an imperfect state measurement tracking controller by combining an $H^\infty$-optimal state estimate with the perfect state feedback controller developed in Chapter 4. Using recent results on the robustness of minimax controllers to nonlinear perturbations [86], we were able to prove a form of disturbance attenuation for the closed-loop system. The imperfect state measurement tracking controller is the final element in the design for planning and controlling the motion of an underactuated vehicle. We are now in a position to combine all of the elements and demonstrate how well they perform through a series of simulations.

# CHAPTER 6

# SIMULATION RESULTS

This chapter presents the results of simulations that implement the motion planning, the perfect state feedback control, and the imperfect state measurement control algorithms described in Chapters 3, 4, and 5. We will explain the details behind each simulation and comment on the results.

Except where noted, for most of the simulations we used the following values for the coefficients in the underactuated vehicle model given by (1.1) through (1.3):

$$m_u = 0.5, \qquad m_v = -2.0, \qquad m_r = 0.5$$

$$d_u = 1.0, \qquad d_v = 2.0, \qquad d_r = 1.0.$$

These parameter values represent a scaled realistic model. When we report a configuration for the system, the order of the variables will be $[u, v, r, x, y, \psi]$.

## 6.1 Motion Planning

Motion planning is the first step in controlling the underactuated vehicle. The techniques presented in Chapter 3 allow us to move the vehicle between a variety of configurations and generate very reasonable trajectories. The simulations presented in this section illustrate the results of the motion planning algorithm with a sample of the types of motions that can be generated.

### 6.1.1 Basic motion planning examples

We applied the basic motion planning algorithm to five example problems to demonstrate its features. The first example shown in Figure 6.1 illustrates how the iterative $H^\infty$-filter helps improve the feasible trajectory and make the vehicle's final configuration approach the goal configuration. In this example, we left the initial configuration of the vehicle as its *actual* configuration for the motion planning algorithm. This design choice appears in Figure 6.1 as a difference between the candidate position curve (dotted line) and the feasible planned position curves along the initial segment of the path. This difference does not diminish as the iterative $H^\infty$-filter operates on the system.

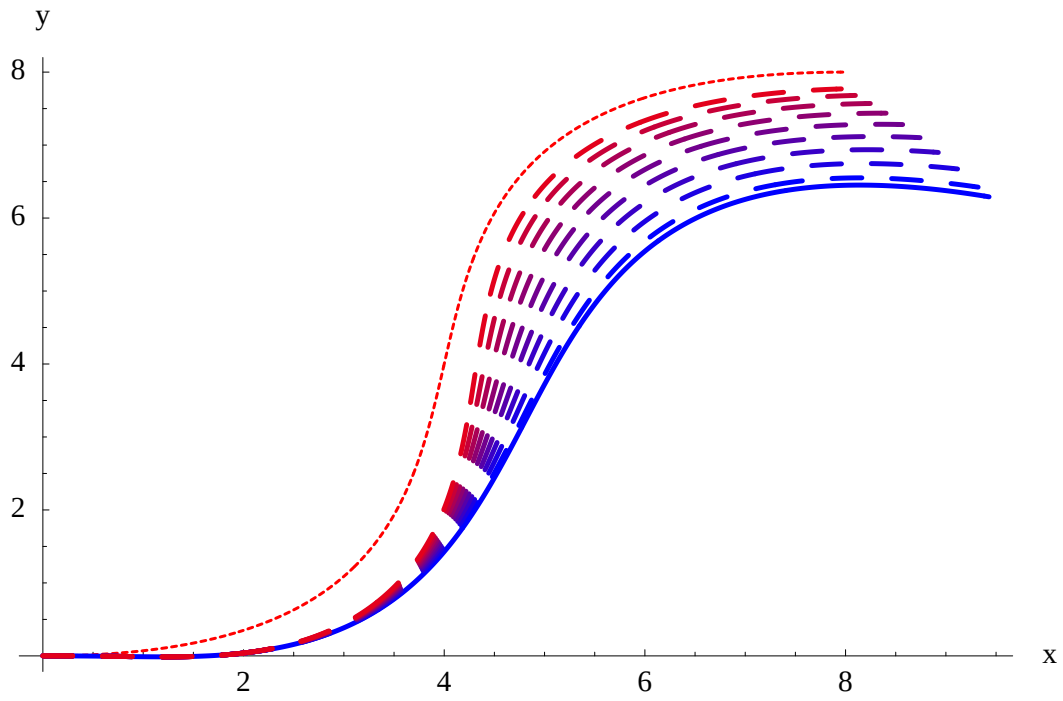



**Figure 6.1** Basic motion planning results using the iterative $H^\infty$-filter. The dotted line shows the candidate polynomial position curve, the solid line shows the first feasible trajectory, and the dashed lines show the feasible trajectories as the $H^\infty$-filter iterates on the results.

Initial Configuration: [1, 0, 0, 0, 0, 0]
Goal Configuration: [1, 0, 0, 8, 8, 0]
Final Configuration: [1.00, 0.10, -0.09, 7.89, 7.81, -0.06]

As described in Chapter 3, we can reduce the initial error between the candidate position curve and the feasible position curve by using the *estimated* initial conditions to plan the vehicle's motion. Figure 6.2 shows the results of applying this technique. In this case, we did not have to apply the iterative $H^{\infty}$-filter to achieve a small final configuration error. Figure 6.3 illustrates the second method for correcting the initial configuration to improve the results. In this example, we developed a motion using the iterative $H^{\infty}$-filter and the actual initial configuration and then shifted the initial orientation and position to completely eliminate orientation and position errors at the final configuration. The amount by which we shift the initial configuration is relatively small and, if it falls within the region of attraction for the tracking controllers, we will be able to follow the new feasible trajectory. Table 6.1 summarizes the initial and final configurations for the first three motion planning examples.

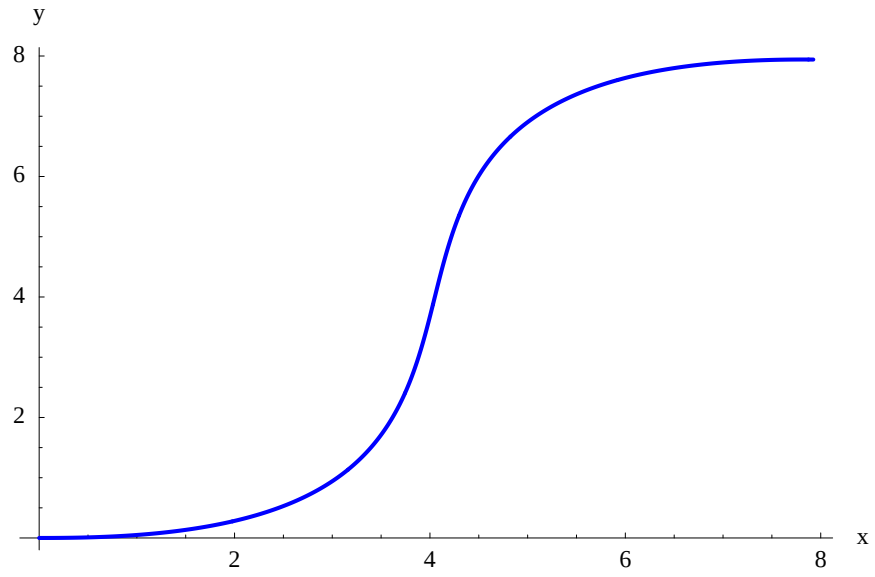



**Figure 6.2** Basic motion planning results using the estimated initial conditions. This example did not use the iterative $H^{\infty}$-filter. The final planned configuration is very close to the goal configuration.
Initial Configuration: [0.99, -0.12, 0.15, 0, 0, 0.12]
Goal Configuration: [1, 0, 0, 8, 8, 0]
Final Configuration: [0.99, 0.10, -0.10, 7.92, 7.94, -0.12]

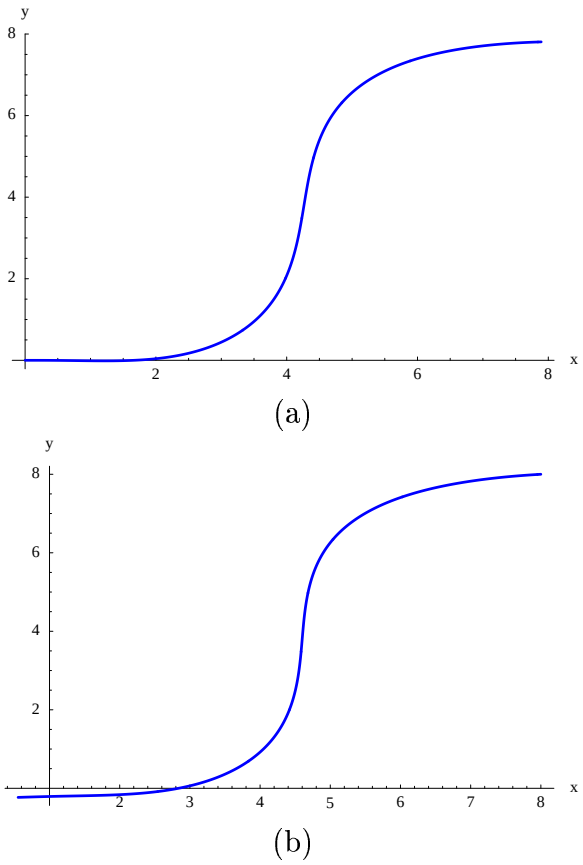



**Figure 6.3** Basic motion planning results with the corrected initial configuration. Part (a) shows the original position curve and part (b) shows the curve after the correction. Note that the values for the final planned position and orientation exactly match the corresponding values in the goal configuration.

Initial Configuration: [1, 0, 0, 0.55, -0.23, 0.06]
Goal Configuration: [1, 0, 0, 8, 8, 0]
Final Configuration: [1.00, 0.10, -0.09, 8, 8, 0]

**Table 6.1** Basic motion planning results with different initial configurations. The table compares the initial and final configurations for three different variations of the basic motion planning algorithm. The three variations are to use the iterative $H^\infty$-filter, to use the estimated initial configuration, and to correct the initial configuration to exactly match the final position and orientation requirements.

| Algorithm | Initial Configuration | Final Configuration |
|---|---|---|
| $H^\infty$-Filter | [1, 0, 0, 0, 0, 0] | [1.00, 0.10, -0.09, 7.89, 7.81, -0.06] |
| Estimated Initial | [0.99, -0.12, 0.15, 0, 0, 0.12] | [0.99, 0.10, -0.10, 7.92, 7.94, -0.12] |
| Corrected Initial | [1, 0, 0, 0.55, -0.23, 0.06] | [1, 0.10, -0.09, 8, 8, 0] |

Figures 6.4 and 6.5 provide two more examples of feasible motions developed using the basic motion planning algorithms. Figure 6.4 shows a U-turn motion and Figure 6.5 displays how the vehicle would make a lateral displacement. These examples verify that the planning approach can solve a wide range of practical problems for an underactuated vehicle.

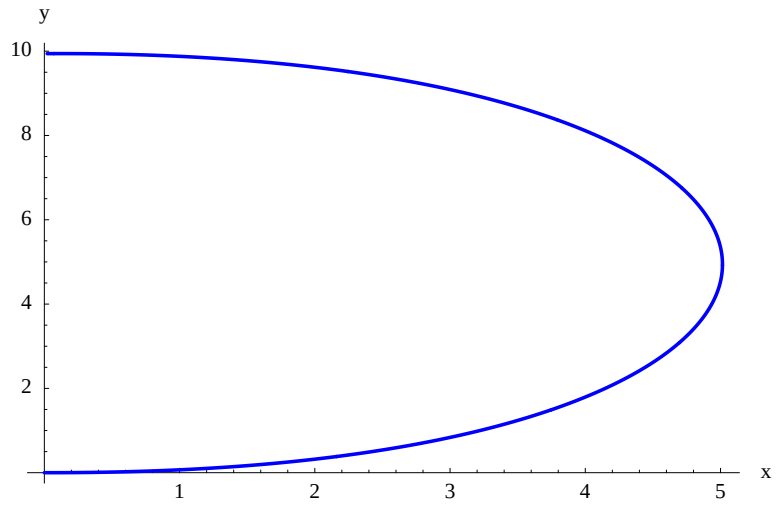



**Figure 6.4** Additional basic motion planning results using the estimated initial configuration. The motion is a feasible position curve for a U-turn and it relied on the $H^{\infty}$-filter to improve the results.
Initial Configuration: [0.99, -0.15, 0.16, 0, 0, 0.15]
Goal Configuration: [1, 0, 0, 0, 10, $\pi$]
Final Configuration: [0.99, -0.14, 0.14, 0.02, 9.94, 3.30]

### 6.1.2 Extensions to basic motion planning examples

To solve the obstacle avoidance and multiple vehicle planning problems, we combined aspects of the basic motion planning algorithm with the rapidly-exploring random trees search technique to develop feasible plans. Figures 6.6 and 6.7 illustrate two examples of these techniques. In Figure 6.6, we planned the motion for a single vehicle around a set of obstacles. The figure shows the raw position curve as planned by the modified RRT software as well as the smoothed curve after applying the iterative $H^{\infty}$-filter. This

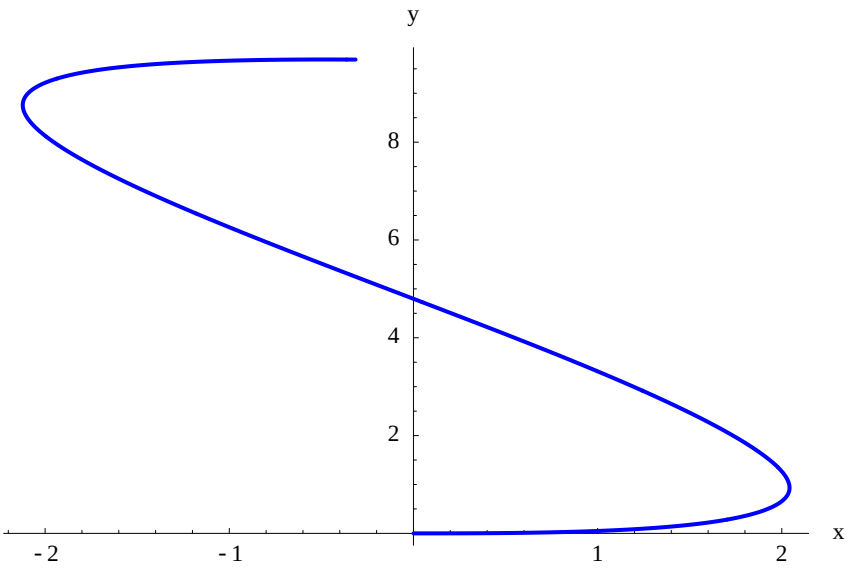
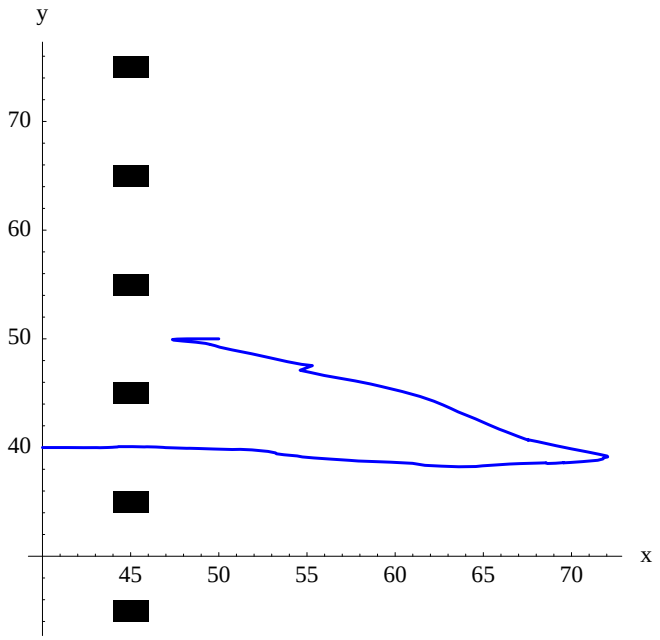


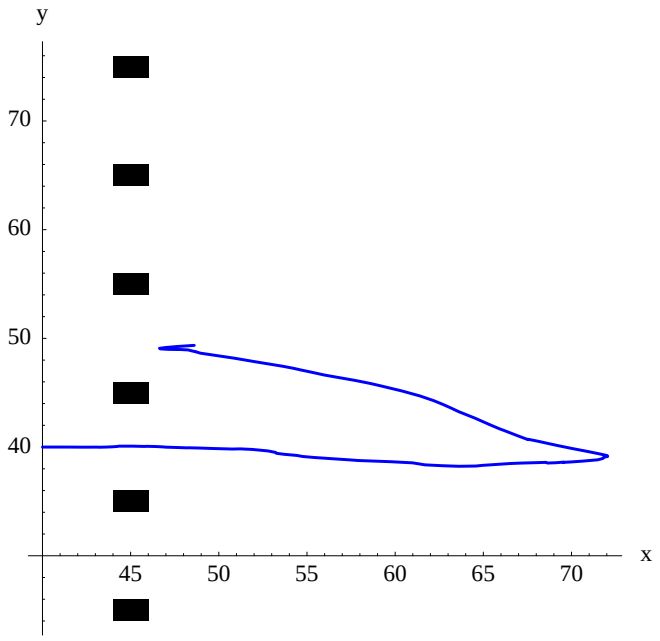

**Figure 6.5** Additional basic motion planning results using the estimated initial configuration. The motion is a feasible position curve for a lateral displacement and it did not require the $H^\infty$-filter to improve the results.
Initial Configuration: [0.99, -0.15, 0.22, 0, 0, 0.15]
Goal Configuration: [1, 0, 0, 0, 10, 0]
Final Configuration: [0.99, 0.10, -0.10, -0.31, 9.69, -0.12]

example highlights how the $H^\infty$-filter softens the transition between the two rapidly-exploring trees that appears as a zigzag at coordinates $(x, y) = (55, 47)$ in Figure 6.6(a), but is absent in Figure 6.6(b).

Figure 6.7 illustrates the results of planning motions for two vehicles around obstacles and applying the sampling smoothing approach to improve the trajectories. To develop the smoothed trajectories in Figure 6.7(b), we sampled the trajectories for each vehicle at the initial and final configurations and at two intermediate configurations. We then used the basic motion planning algorithm to plan motions between successive configurations to achieve substantially smoother feasible trajectories. In this case, we required a minimum amount of sampling to capture the important aspects of the vehicles' motions to avoid collisions. Other examples may require more intermediate sample points to achieve similar results.



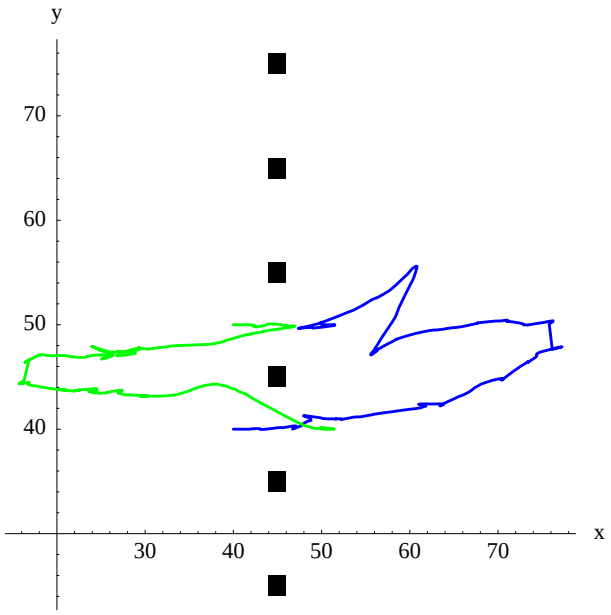

(a)



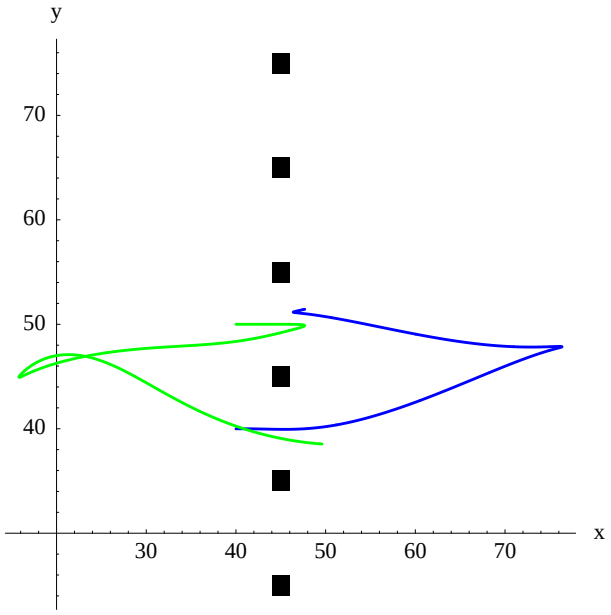

(b)

**Figure 6.6** Motion planning with obstacles in the environment. Part (a) shows the results of the RRT algorithm and part (b) shows how smoothing the trajectory with the iterative $H^{\infty}$-filter improves the results. Both trajectories are feasible. The dark boxes are the obstacles.

| | |
|---|---|
| Initial Configuration: | [1, 0, 0, 40, 40, 0] |
| Goal Configuration: | [1, 0, 0, 50, 50, 0] |
| Final Configuration: | [1.21, -0.03, 0.01, 48.61, 49.36, 0.11] |




(a)




(b)

**Figure 6.7** Motion planning for multiple vehicles with obstacles. Part (a) shows the results of the RRT algorithm and part (b) shows how smoothing the trajectories with sampling improves the results. We used the basic motion planning algorithm to plan the motions between the sample points. Both trajectories are feasible.

| | Vehicle 1 | Vehicle 2 |
|---|---|---|
| Initial Configurations: | [1, 0, 0, 40, 40, 0] | [1, 0, 0, 40, 50, 0] |
| Goal Configurations: | [1, 0, 0, 50, 50, 0] | [1, 0, 0, 50, 40, 0] |
| Final Configurations: | [0.96, 0, 0, 49.74, 50, 0] | [0.99, -0.11, 0.11, 49.62, 38.61, 0.06] |

## 6.2 Perfect State Feedback Examples

The perfect state feedback tracking controller forces the vehicle to follow the feasible trajectories generated by the motion planning algorithms. Figures 6.8 and 6.9 illustrate the results of applying this controller when different disturbances are present and the vehicle has significantly different initial conditions. In Figure 6.8, the system is subjected to a white noise disturbance with $\sigma = 0.5$ and the vehicle's initial configuration aligns with the expected initial configuration. The controller successfully rejects the influence of the noise to follow the feasible trajectory and approach the desired final configuration.

Figure 6.9 illustrates the effect of applying a sinusoidal disturbance to the system and giving the vehicle an exceptionally large initial configuration error. This type of large initial error could occur if the underactuated controller was used as a backup system for a fully actuated vehicle. In this case, there would be a delay between the actuator failure and the application of the new controller, which would allow the vehicle to drift away from the intended trajectory. Section 6.6 provides a more detailed discussion of this example. Even though the initial configuration is drastically different from the expected initial configuration, the tracking controller corrects the error using a natural motion and makes the vehicle approach the feasible trajectory. The sinusoidal nature of the disturbances is visible in the velocity plots in Figure 6.9(a) and in the control input plots in Figure 6.9(c).

To verify that the tracking control algorithm works for the obstacle avoidance and multiple vehicle cases, we applied the perfect state feedback tracking controller to the motions planned in Figures 6.6 and 6.7. Figure 6.10 displays the results. In both cases, the vehicles tracked the feasible trajectories without collisions, as expected.

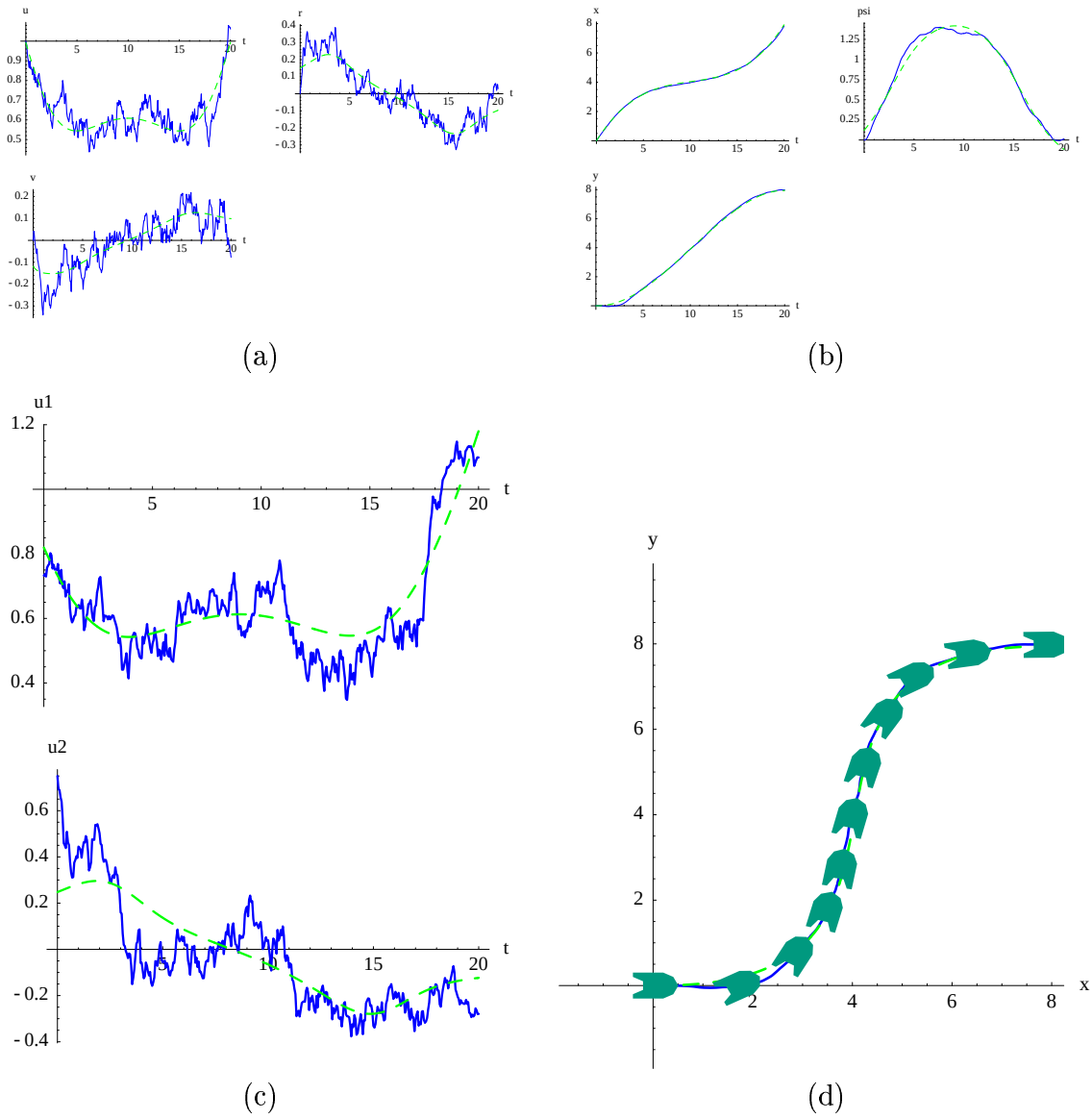



**Figure 6.8** Perfect state feedback trajectory tracking. Part (a) shows the desired (dashed lines) and actual (solid lines) velocities for the vehicle, part (b) shows the positions and orientations, part (c) shows the two control inputs, and part (d) displays stop-motion images of the vehicle as it tracks the $x$-$y$ position curve. The disturbances are white noise with $\sigma = 0.5$.

| | |
|---|---|
| Initial Configuration: | [1, 0, 0, 0, 0, 0] |
| Goal Configuration: | [1, 0, 0, 8, 8, 0] |
| Final Configuration: | [1.06, -0.08, 0.03, 7.84, 7.98, 0.01] |

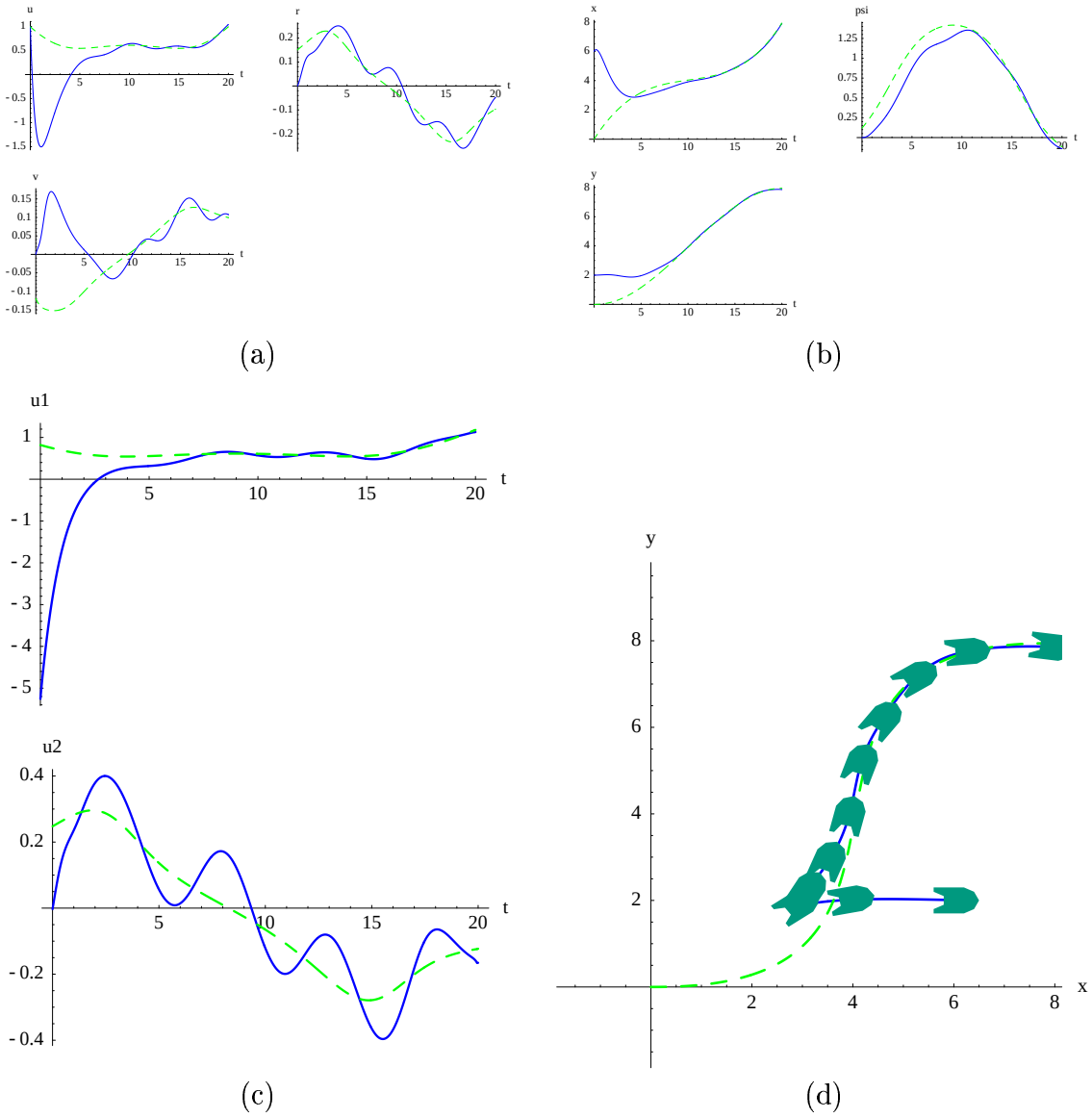
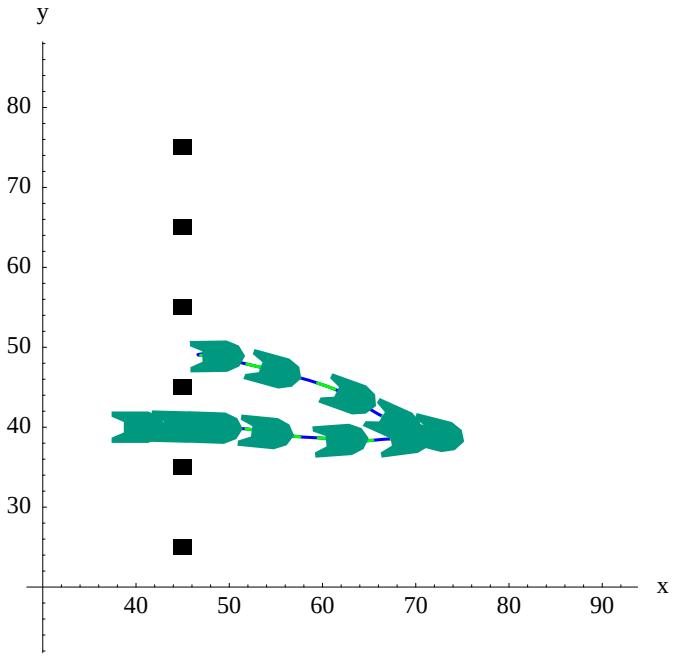
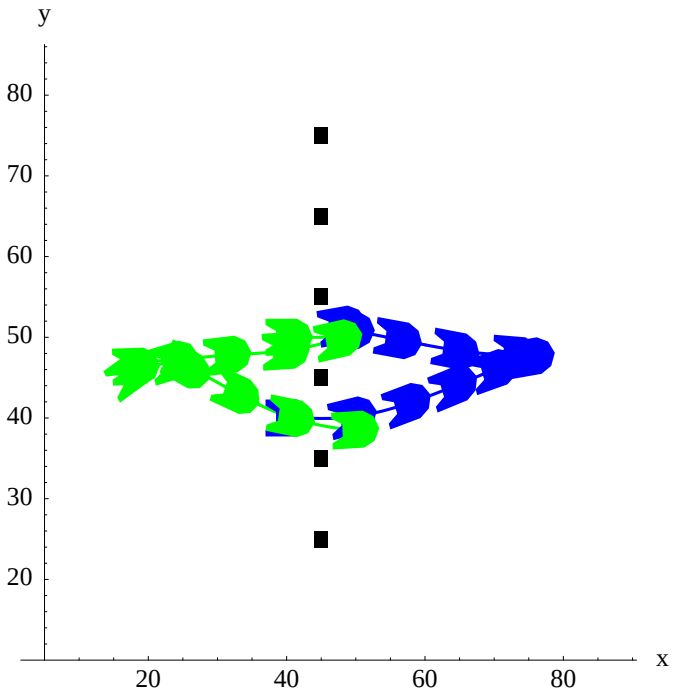



**Figure 6.9** Perfect state feedback trajectory tracking with poor initial conditions. The vehicle has a significant displacement from the desired trajectory, but the tracking controller is able to generate a natural motion to recover from the initial error. The disturbances are sinusoidal signals with an amplitude of 0.1 and a frequency of 0.2 Hz.

Initial Configuration: [1, 0, 0, 6, 2, 0]
Goal Configuration: [1, 0, 0, 8, 8, 0]
Final Configuration: [1.04, 0.11, -0.05, 7.91, 7.86, -0.14]




(a)




(b)

**Figure 6.10** Perfect state feedback tracking with obstacles and multiple vehicles. Part (a) shows stop-motion images of the vehicle tracking the smoothed $x$-$y$ position curve around obstacles, as planned in Figure 6.6. Part (b) shows stop-motion images for two vehicles with coordinated motions around obstacles, as planned in Figure 6.7.

## 6.3 Imperfect State Measurement Examples

The imperfect state measurement controller can only directly access the position and orientation variables for feedback, so it must develop estimates for the other three states to implement the control law. Figures 6.11 and 6.12 depict the results of applying the imperfect state measurement controller to the same examples illustrated in Figures 6.8 and 6.9. The figures corresponding to the imperfect state measurement case include estimates for the state variables as dotted lines in parts (a) and (b) of the figures.

Comparing Figure 6.11 with Figure 6.8, we observe that the tracking performance for the imperfect state measurement case is not as good as for the perfect state measurement case. We expect the imperfect state measurement approach will produce larger errors because of the extra estimation required in the algorithm, but the final configuration errors are similar for the two approaches and the overall results are acceptable.

In Figure 6.12 with the imperfect state measurement controller, we note that the vehicle actually converges on the desired position curve sooner than in Figure 6.9, where we applied the perfect state feedback control law. This occurs because the imperfect state measurement controller is using the state estimates to develop the control inputs and, in this case, the estimates positively interact with the initial configuration to improve the overall performance. We expect this performance to be atypical for the imperfect state measurement approach.

To complement the examples for the imperfect state measurement controller, we now present a small set of simulations that demonstrate the validity of Assumption A5.1 for an underactuated ship model. The three figures show how changes in the disturbance term $w$ affect the eigenvalues of $F(t)$ and the values in $\beta(t)$. Section 5.3 defines these terms and explains their importance to Assumption A5.1. Each simulation is based on the vehicle tracking a feasible circular trajectory. Figures 6.13 through 6.15 each contain

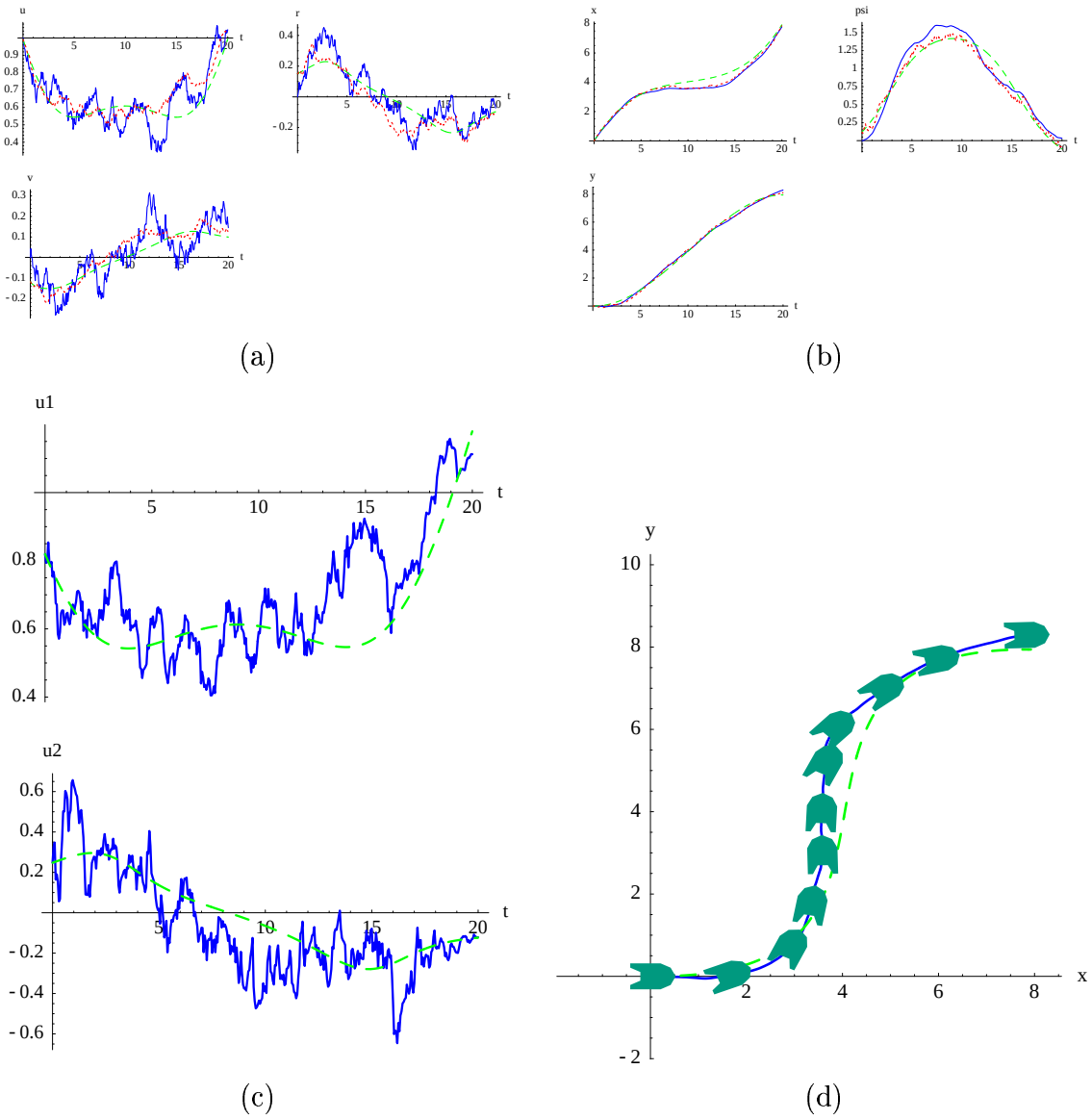



**Figure 6.11** Imperfect state measurement tracking control. Part (a) shows the desired (dashed lines), estimated (dotted lines), and actual (solid lines) velocities for the vehicle; part (b) shows the positions and orientations; part (c) shows the two control inputs; and part (d) displays stop-motion images of the vehicle as it tracks the $x$-$y$ position curve. The disturbances are white noise with $\sigma = 0.5$.

Initial Configuration: [1, 0, 0, 0, 0, 0]
Goal Configuration: [1, 0, 0, 8, 8, 0]
Final Configuration: [1.05, 0.15, -0.07, 7.79, 8.29, 0.03]

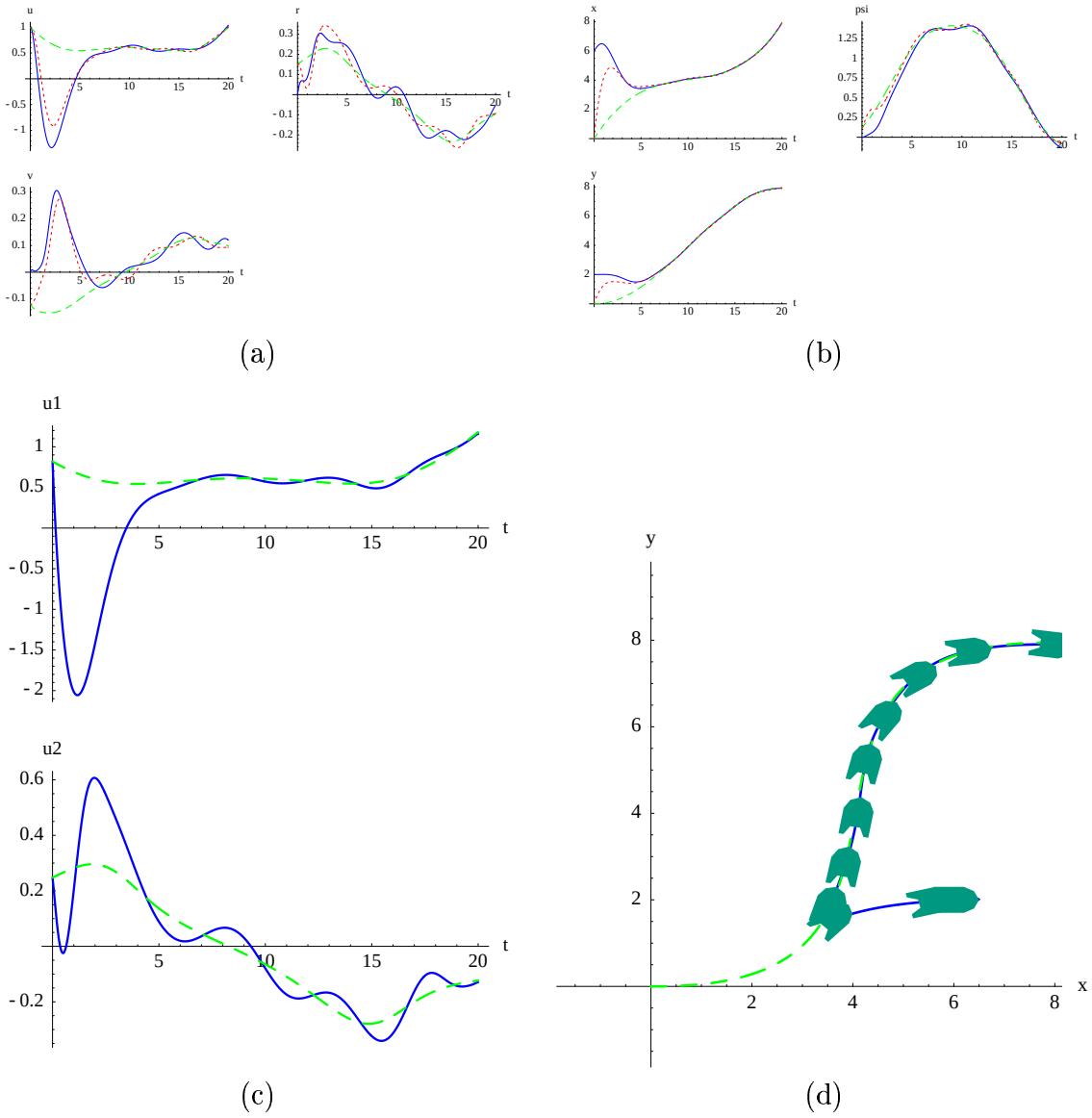



**Figure 6.12** Imperfect state measurement case with poor initial conditions. The system disturbances are sinusoidal signals with an amplitude of 0.1 and a frequency of 0.2 Hz, and the measurement disturbances are white noise with $\sigma = 0.5$.

Initial Configuration: [1, 0, 0, 0, 0, 0]
Goal Configuration: [1, 0, 0, 8, 8, 0]
Final Configuration: [1.04, 0.12, -0.05, 7.91, 7.90, -0.14]

three parts. Parts (a) and (b) plot the eigenvalues of $F(t)$ versus time, and part (c) displays the nonzero elements of $\beta(t)$ versus time. In each plot, the solid line shows the results for some nonzero disturbance and, for comparison, the dashed line shows the values when there are no disturbances.

Figure 6.13 applies the worst-case disturbance and the other figures use sampled white noise with $\sigma$ set to 0.5 and 5. In each case, the disturbances have a relatively mild effect on the eigenvalues of $F(t)$ and on $\beta(t)$. With $\sigma = 5$ in Figure 6.15, the role of the noise is more prominent, but the magnitude of the changes is not significantly different from that with $\sigma = 0.5$ in Figure 6.14.

These simulations show that Assumption A5.1 is reasonable for the desired trajectory examined is this case. The validity of Assumption A5.1 should be verified for each system and for all desired trajectories to ensure the controller designed in Section 5.3 will deliver acceptable performance.

## 6.4 Comparison of Results to Existing Techniques

To help judge the value of the controller designs, we compared the tracking performance to two other controller design techniques for underactuated vehicles. The other techniques were developed by Godhavn [51] and by Pettersen and Nijmeijer [3]. As described in Chapter 2, both of these approaches rely on backstepping to develop their results. We provide a single simulation example that highlights the key differences between our approach and the other two controller designs. Figure 6.16 shows the stop-motion images of the vehicle following the $x$-$y$ curve using the state feedback tracking controller designed in Chapter 4. The motion planning algorithm assumed that the vehicle had an initial orientation of $\psi(t_0) = 0$, but the actual orientation was $\psi(t_0) = \pi$. The state feedback controller was able to correct for the initial error and make the vehicle converge

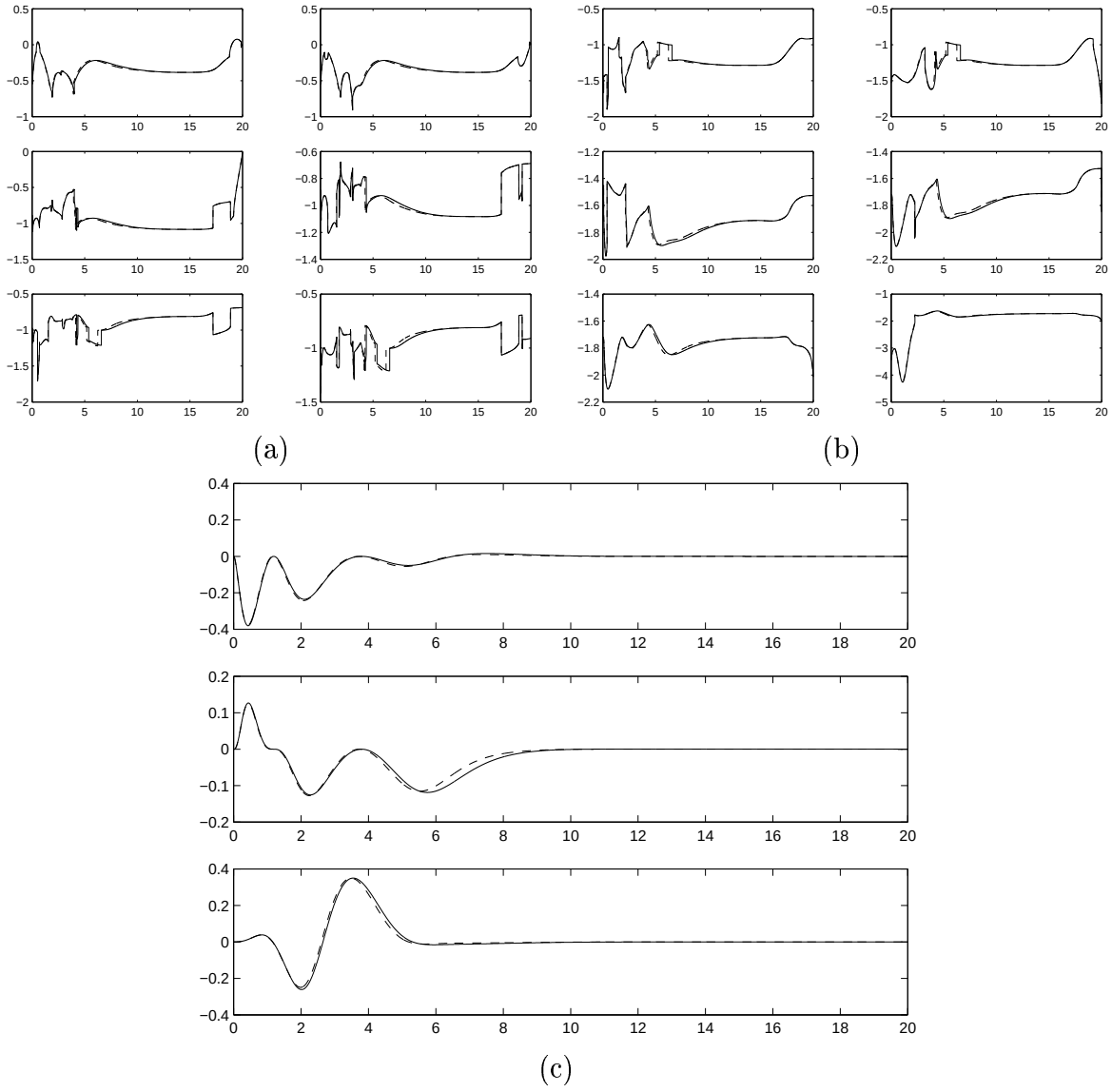



**Figure 6.13** Analysis of terms influenced by the disturbance. Parts (a) and (b) show the 12 eigenvalues of $F(t)$ plotted versus time. Part (c) shows the values of the three nonzero elements of $\beta(t)$ versus time. The solid lines are the values when the disturbance is the worst-case disturbance, and the dashed lines show the values when there are no disturbances.

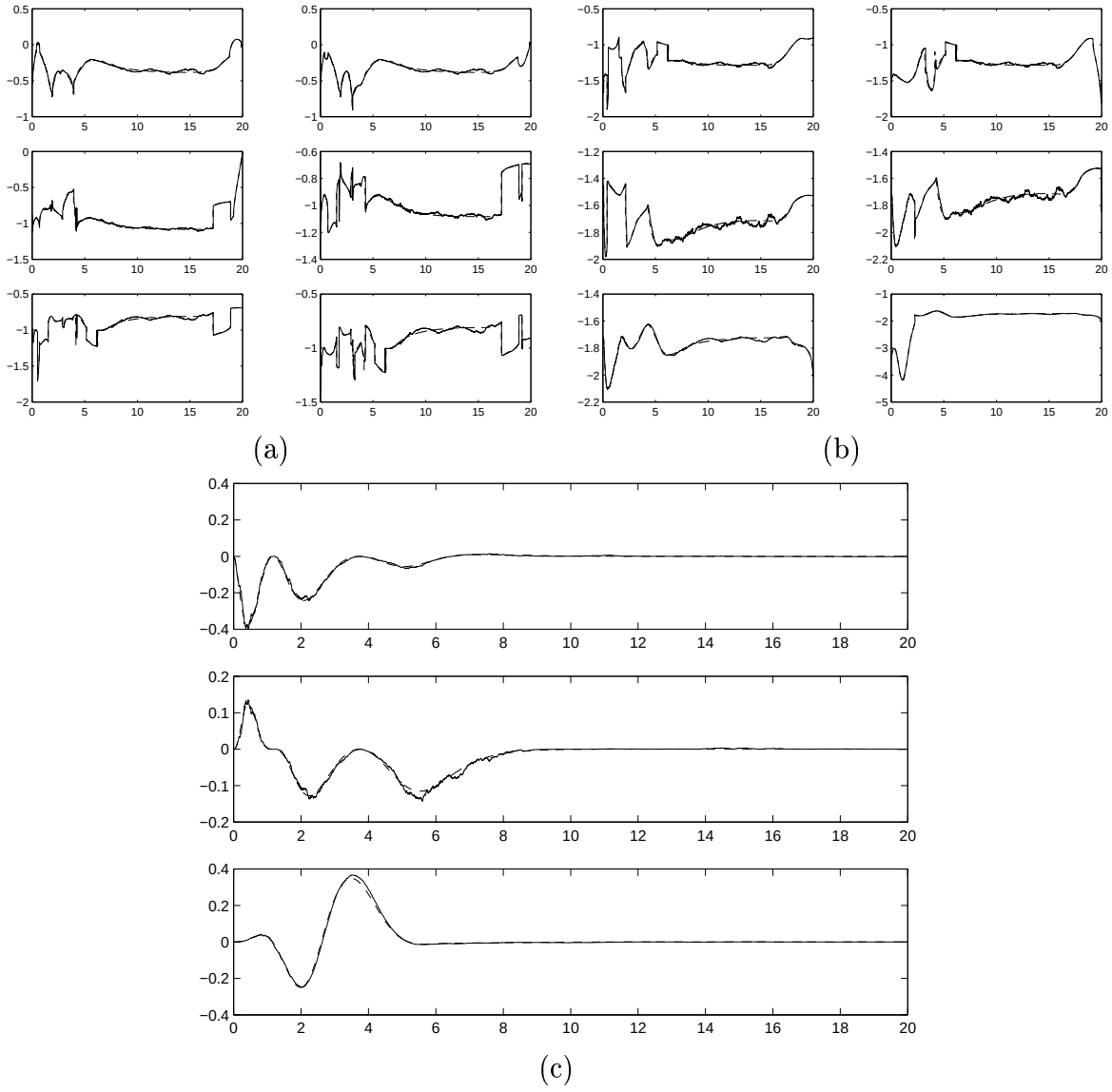



**Figure 6.14** Analysis of terms influenced by the disturbance. Parts (a) and (b) show the 12 eigenvalues of $F(t)$ plotted versus time. Part (c) shows the values of the three nonzero elements of $\beta(t)$ versus time. The solid lines are the values when the disturbance is the sampled white noise with $\sigma = 0.5$, and the dashed lines show the values when there are no disturbances.

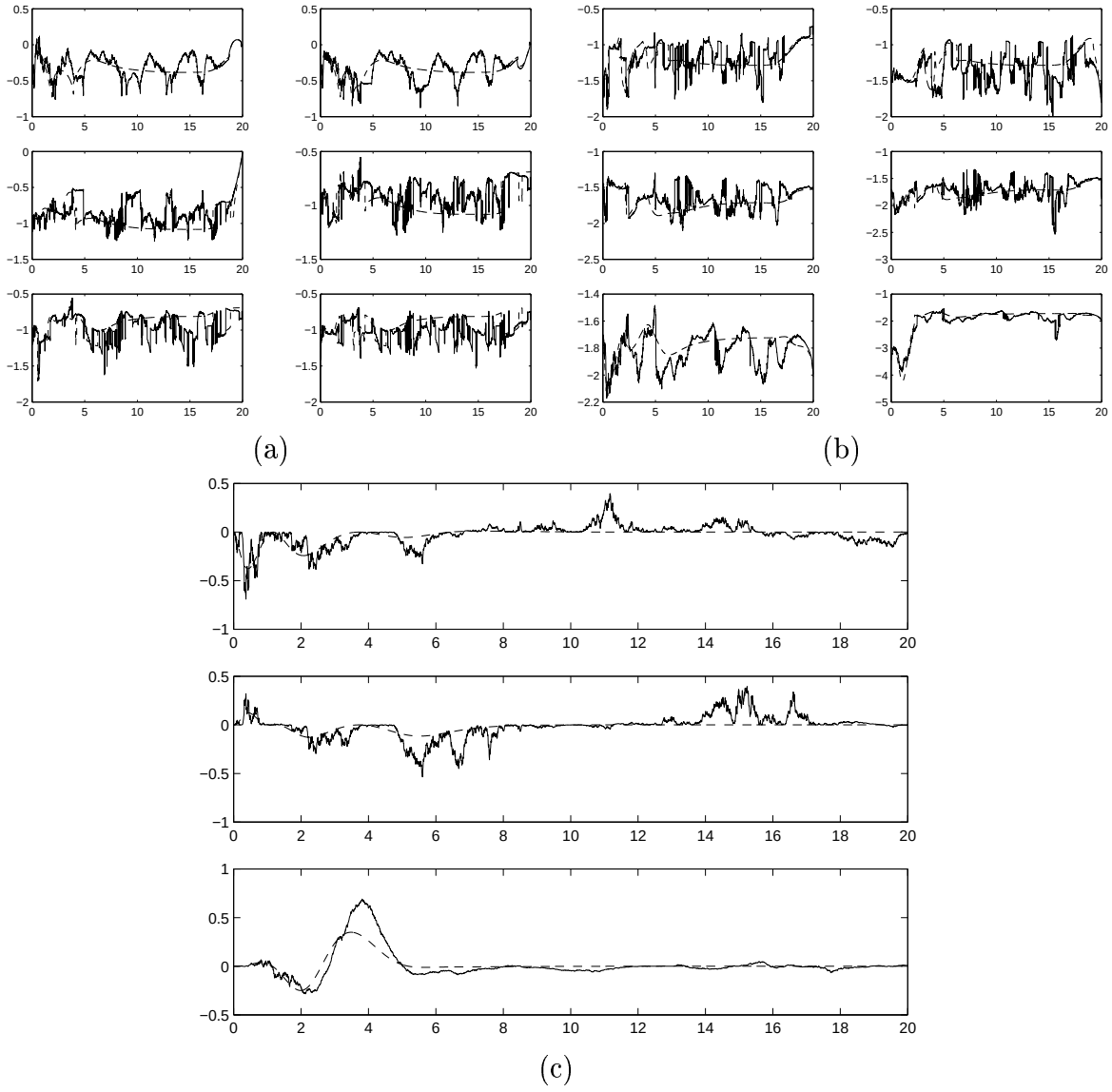



**Figure 6.15** Analysis of terms influenced by the disturbance. Parts (a) and (b) show the 12 eigenvalues of $F(t)$ plotted versus time. Part (c) shows the values of the three nonzero elements of $\beta(t)$ versus time. The solid lines are the values when the disturbance is the sampled white noise with $\sigma = 5.0$, and the dashed lines show the values when there are no disturbances.

on the desired trajectory. There is a small final orientation error, but the final position and velocities are close to the desired values.

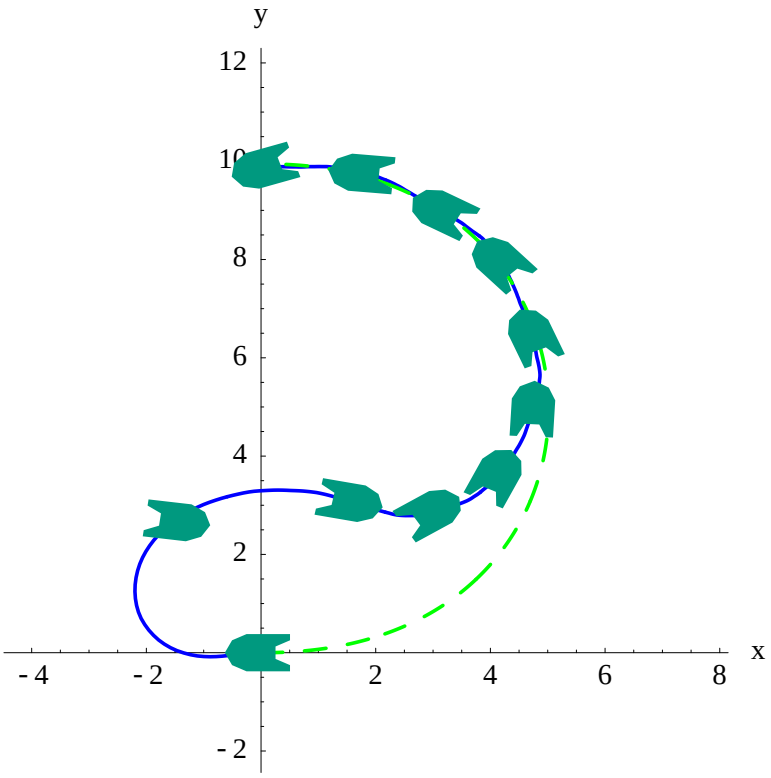



**Figure 6.16** Comparison of perfect state feedback tracking to other techniques. The motion planning algorithm assumed the vehicle had an initial orientation of $\psi(t_0) = 0$, but the actual orientation was $\psi(t_0) = \pi$. The state feedback controller corrects for the initial configuration error and forces the vehicle to track the desired trajectory, even though there are white noise disturbances with $\sigma = 0.5$ affecting the system.

| | |
|---|---|
| Actual Initial Configuration: | [1, 0, 0, 0, 0, $\pi$] |
| Expected Initial Configuration: | [1, 0, 0, 0, 0, 0] |
| Goal Configuration: | [1, 0, 0, 0, 10, $\pi$] |
| Final Configuration: | [1.04, -0.32, 0.26, 0.09, 9.88, 3.46] |

The results shown in Figure 6.16 are an improvement over the technique proposed by Godhavn for two reasons. First, the tracking controller developed in Chapter 4 corrects the orientation of the vehicle. Godhavn's backstepping controller does not penalize configurations when the vehicle is oriented backwards along the trajectory and cannot correct this situation. Godhavn's technique would force the vehicle to move backwards

along the path. The second advantage of our approach is that the technique is designed to be robust to disturbances and to attenuate their effect on the system. As shown in Figure 6.16, the vehicle still captures the desired trajectory when disturbances are present. The backstepping derivation requires an accurate model of the system and does not consider the effect of disturbances. Godhavn remarks in [51] that disturbances can be included in the model and addressed with backstepping, but does not present the results. We should note that Godhavn's approach is still a reasonable method for performing tracking for an underactuated vehicle if the initial configuration has a small orientation error and there are no disturbances in the system.

Our techniques also offer some improvements over those developed by Pettersen and Nijmeijer [3]. The results in Figure 6.16 show that the vehicle converges to the desired $x$-$y$ position curve after correcting the initial error. The simulations reported in [3] indicate that the controller was not able to make the vehicle converge to the desired position curve. As in our example, the desired position curve in [3] was a circular arc, but the vehicle always maintained a slight offset from the desired position. In addition, one version of the Pettersen and Nijmeijer controller was designed with pure backstepping and did not perform well with unmodeled dynamics. The authors do describe another controller to correct for this deficiency, which makes their approach robust to modeling errors. With the exception of the minor convergence problem, the Pettersen and Nijmeijer approach offers a viable alternative to our design,

## 6.5 Extension of Techniques to Additional Models

In this section we will consider three extensions to the basic model used for the majority of the experiments. First, we will consider the performance of the motion planning and control algorithms when we introduce a nonminimum phase behavior into

the model. Next we will change the parameter values in the model to demonstrate that the techniques work for vehicles with different properties. Finally, we will provide a few simulations that illustrate how the state feedback tracking controllers perform when the disturbances affect the unactuated directions in the model.

We performed a set of three simulations for the nonminimum phase system described in Section 4.5 and compared the results to the original system with $k_v = 0$. To simplify the comparison, we set the disturbances to zero, but including disturbances should not significantly alter the results. In each simulation, the system started at $\mathbf{q} = [1,\ 0,\ 0,\ 0,\ 0,\ 0]^{\mathrm{T}}$. Table 6.2 summarizes the results of the three experiments. In each case, the final tracking performance was similar for the original system and the nonminimum phase formulation. There were only minor and insignificant differences between the results. These experiments indicate that the motion planning and control algorithms perform well for both minimum and nonminimum phase systems.

The second extension to the original vehicle model shows that the motion planning and tracking algorithms still provide good solutions when we fundamentally change the parameters in the model. We selected the following values for the model parameters:

$$m_u = 2.0, \qquad m_v = -0.5, \qquad m_r = -2.5$$

$$d_u = 2.0, \qquad d_v = 1.0, \qquad d_r = 5.0.$$

**Table 6.2** Comparison of minimum and nonminimum phase systems. The final configurations are nearly identical for the two approaches.

| Desired Final | Minimum Phase Final | Nonminimum Phase Final |
|---|---|---|
| [1, 0, 0, 8, 8, 0] | [0.99, 0.10, -0.10, 7.92, 7.94, -0.11] | [1.00, 0.09, -0.10, 7.85, 7.95, -0.12] |
| [1, 0, 0, 0, 10, 0] | [0.99, 0.10, -0.10, -0.31, 9.69, -0.12] | [0.98, 0.08, -0.07, -0.19, 9.59, -0.20] |
| [1, 0, 0, 0, 10, $\pi$] | [0.99, -0.14, 0.14, 0.02, 9.94, 3.30] | [0.99, -0.13, 0.13, -0.07, 10.01, 3.28] |

These new parameter values contain two significant changes from the original set. First, we exchanged the relative magnitudes of $m_u$ and $m_v$ and, second, we changed the sign of $m_r$. These values represent a change in the shape of the vehicle from one that moves with its major axis perpendicular to the forward velocity to one with the major axis aligned with the forward velocity. We simulated this system with a motion from [1, 0, 0, 0, 0, 0] to [1, 0, 0, 8, 8, 0] and Figure 6.17 displays the results. The motion planning and perfect state feedback tracking control algorithms successfully handled a fundamental change in the parameter values for the model.

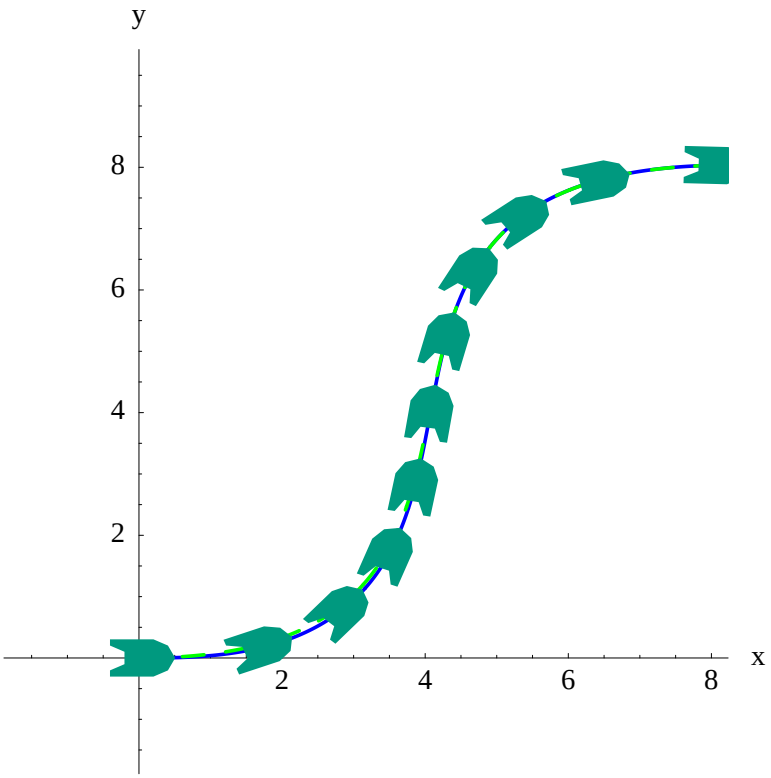



**Figure 6.17** Motion planning and tracking with a new vehicle model. The algorithms successfully planned the motion and controlled the vehicle after a fundamental change in the model parameter values.

The final set of simulations in this section examines how well the tracking controller attenuates different types of disturbances. We used the basic motion planning algorithm to plan a motion from [1, 0, 0, 0, 0, 0] to [1, 0, 0, 8, 8, 0], and then subjected the system

to four different types of disturbances. The first simulation applied disturbances with constant amplitude signals along the actuated directions and no signal in the unactuated direction, so that we had $\mathbf{w} = [w_1, w_2, w_3]^{\mathrm{T}} = [0.1, 0, 0.1]^{\mathrm{T}}$, in the equations of motion. For the second simulation, we applied the constant amplitude disturbance in only the unactuated direction, so that we had $\mathbf{w} = [0, 0.1, 0]^{\mathrm{T}}$. The third and fourth simulations mirrored the first two in terms of the directions the disturbances influenced, but the constant amplitude signals were replaced with sinusoidal signals of the form $w_i(t) = 0.1\cos[2\pi(0.2)t]$. Table 6.3 summarizes the results of applying the perfect state feedback tracking controller with the different types of disturbances. The controller successfully compensates for the disturbances regardless of their direction. In all cases, the controller was able to force the vehicle to approach the desired final configuration at the appointed time. The velocity errors for $v$, the unactuated direction, were slightly larger in the two cases where the disturbances affected only that direction, but the final configurations were still very close to the desired values.

We performed additional simulations where the disturbances depended on the orientation of the vehicle. This situation would arise if the vehicle was turning and the disturbances were fixed with respect to the inertial frame. An example of this case would be a ship turning while the waves continued to move in a constant direction. The perfect state feedback tracking controller successfully handled this situation by rejecting the dis-

**Table 6.3** Disturbance attenuation results with disturbances affecting different directions. The table shows the final configurations using the perfect state feedback tracking controller with the various combinations of disturbances. The controller attenuates the influence of the disturbances even when they only affect the unactuated directions in the system.

| Directions | Constant Amplitude | Sinusoidal |
|---|---|---|
| Actuated | [1.04, 0.07, -0.05, 8.00, 8.10, -0.05] | [1.04, 0.07, -0.05, 7.91, 7.90, -0.14] |
| Unactuated | [1.01, 0.13, -0.08, 7.89, 8.11, -0.13] | [0.99, 0.13, -0.09, 7.93, 7.91, -0.12] |

turbances, regardless of the vehicle's orientation, and approximately tracking the desired trajectory.

Taken together, the three extensions to the motion planning and control algorithms described in this section convincingly show that the approach can be generalized to handle different types of systems and is robust enough to neutralize the effect of a variety of disturbances.

## 6.6 Failure Recovery for a Fully Actuated Vehicle

One potential application for the ability to plan motions for an underactuated vehicle is to allow a fully actuated vehicle to gracefully recover from an actuator failure. As mentioned in Chapter 1, this capability could be extremely helpful for aircraft or spacecraft systems where an actuator failure can mean the difference between being able to control the vehicle and suffering a catastrophic loss. The motion planning and tracking control problems for the fully actuated system are fundamentally different from those for an underactuated system because, in the fully actuated case, the vehicle can potentially follow *any* reference trajectory. The only limitations on the motion would be from actuator saturation. If the system suffers a failure, the control law designed for the fully actuated case will not have the desired effect and we will have to replan the motion and apply a new controller to satisfy the motion requirements. We will outline how the motion planning and control algorithms for an underactuated vehicle could serve as a key element in a failure recovery system for a fully actuated vehicle. We will leave experimentation and further study of this topic as an area for future research.

Suppose we are given a model for a fully actuated system that may be subject to actuator failure during normal operation. If an actuator failed, we would execute the following actions to maintain control over the vehicle. First, we would have to identify

the actuator failure and adjust the model of the vehicle to reflect the change in dynamics. Identifying the failure can be a nontrivial problem. One clear indication of a failure would be that the vehicle is no longer properly tracking planned motions. For critical applications, the vehicle could also be equipped with sensors to indicate actuator failures, but the sensors add an extra layer of expense and complexity which may not be desirable. It is reasonable to assume that we will not be able to completely identify the new system model and there will be unmodeled dynamics in the revised model. These unmodeled dynamics can be considered as disturbances for the new control law to attenuate.

Once the actuator failure is identified and the model corrected, we can use the original motion plan for the vehicle and the basic motion planning algorithm to develop a set of alternative feasible motions for the vehicle. With a new set of feasible motions, we could apply one of the tracking controllers developed in Chapters 4 or 5 to control the vehicle's motion. Since we used the $H^\infty$ design approach, the tracking controllers are robust and can handle a variety of bounded disturbances, including those that result from unmodeled dynamics. The revised system may not be able to complete the original mission objectives, but the underactuated design may provide sufficient control over the vehicle to allow us to bring it to a safe configuration where repairs can be made.

We presented a brief characterization of how the motion planning and control algorithms for an underactuated vehicle could be used to assist a fully actuated vehicle that suffered an actuator failure. These ideas point to a line of research that is rich with opportunity for future study.

## 6.7 Computational Complexity

As noted above, we completed most of the motion planning and tracking control simulations using the Mathematica software. This software allowed us to solve the problems

by writing a high-level code that is very similar to the equations used to describe the original problems. The drawback with using a high-level language such as Mathematica is that the software involves a significant amount of processing overhead, which increases the time required to complete the simulations. To develop more efficient software, the motion planning and tracking algorithms could be implemented in C/C++, which would allow us to eliminate the processing overhead associated with Mathematica and decrease the simulation times. We will leave these types of enhancements to the simulations as an area for future study.

Regardless of the manner in which we implement the motion planning and tracking control algorithms, the overall computational complexity of the algorithms will remain the same. We will now outline the key computational requirements for the basic motion planning algorithm and the two tracking control algorithms to help characterize the solution. For this discussion, we will use the variable $m$ to denote the number of states in the configuration vector and the variable $n$ to denote the number of time steps between $t_0$ and $t_f$ in a simulation. We will also use the order notation $O(\cdot)$ to characterize the rate at which the processing time grows with respect to the number of states and the number of time steps. We will present a general estimate of the processing growth rate and will not attempt an in-depth investigation to this aspect of the simulations.

As described in Section 3.2, the basic motion planning algorithm consists of four main steps. The first step generates the polynomial curve in the $x$-$y$ plane that connects the initial and final positions with a curve having the correct orientation at each end-point. This operation depends on only the initial and final configurations and the time interval, but not the number of steps in the time interval. It does not matter if we use the cubic spline or the Pythagorean hodograph curve solution for the first step, since both approaches are independent of the variable $n$. Increasing the number of states in the configuration will increase the processing time, and the algorithm will have to be

adapted to plan position curves for nonplanar systems. With these factors in mind, we estimate the processing time for the first step as $O(m)$. The second motion planning step estimates the complete trajectory and the corresponding control inputs at each time step $n$, so this step is order $O(mn)$. The third step uses the estimated inputs in conjunction with the differential equations of motion to generate a feasible trajectory for the vehicle, which requires $O(mn)$ operations. The final motion planning step implements the iterative $H^{\infty}$-filter. Each iteration of the filter requires solving a differential equation to estimate each state and a matrix differential equation to find the filter covariance matrix. Both differential equations are evaluated at each time step, so the fourth step is $O(m^2n)$. Combining the four steps in the motion planning algorithm, we conclude that the processing time is $O(m^2n)$.

The perfect state feedback tracking control algorithm contains two key steps to develop the control law. The first step is to linearize the equations of motion at each time step along the desired trajectory. This operation is $O(mn)$ because each state contributes one linearization equation and the equations are evaluated at each time step. The second step in the algorithm is to solve the generalized Riccati differential equation at each time step, which adds $O(m^2n)$ operations to the processing time. The $m^2$ term appears because the GRDE is a matrix differential equation. Overall, the perfect state feedback tracking control algorithm is $O(m^2n)$.

The analysis of the computational complexity for the imperfect state measurement tracking controller includes the operations required for the perfect state feedback case. We must also account for the additional processing to estimate the complete state vector. To estimate the state, we must solve a vector differential equation with $O(mn)$ and an associated matrix differential equation of order $O(m^2n)$. Therefore, the imperfect state measurement tracking controller is also order $O(m^2n)$.

If we couple the motion planning algorithm with either of the tracking controllers, the computational complexity of the solution grows with order $O(m^2 n)$. This result indicates that increasing the number of states $m$ in the underactuated system increases the processing time at a faster rate than increasing the number of time steps $n$ in the simulation. As indicated above, this estimate for the computational complexity does not depend on the type of software used to implement the algorithms.

## 6.8 Simulation Results Summary

We have presented in this chapter a series of simulations that illustrate the motion planning and controlling algorithms developed in Chapters 3, 4, and 5. The simulation results also offer empirical support for Assumption 5.1 by demonstrating that the disturbances do not significantly affect the terms $F(t)$ and $\beta(t)$, which appear in the imperfect state measurement controller design. We compared our design techniques to other approaches that address similar problems and showed how our design approaches work well with different vehicle models. In addition, this chapter described how to apply the planning and controlling techniques to allow a fully actuated vehicle to recover from an actuator failure. Finally, we outlined the computation complexity of the significant algorithms used to solve the motion planning and tracking control problems. The simulation results verify that our design techniques provide practical solutions to the challenging problem of planning and controlling the motion of a nonlinear underactuated vehicle when disturbances are present and only imperfect state measurements are available for feedback.

# CHAPTER 7

# SUMMARY

This concluding chapter summarizes the contributions of the completed research and outlines the additional research areas related to this effort that could be explored in the future.

## 7.1 Completed Research and Contributions

Our research has focused on developing a suite of tools to plan and robustly control the motion of a nonlinear underactuated vehicle when disturbances are present in the system and only imperfect state measurements are available. Where possible, we have used $H^\infty$-optimal control techniques to efficiently construct solutions that provide excellent performance under a variety of conditions. There are five main contributions of the completed research. First, we showed how to use polynomial curves and numerical methods to develop feasible trajectories for underactuated vehicles. This approach allowed us to directly develop trajectories without extensive search routines. The second contribution is the introduction of an iterative $H^\infty$-filter to improve the planned motions for the vehicle. The filter accounted for the underactuated nature of the system to help improve the results. Third, we showed how to combine the results of the basic motion planning algorithm with existing planning techniques to generate solutions to the obstacle avoidance and multiple vehicle planning problems. The fourth contribution is the

development and analysis of the linearized $H^\infty$ controller for an underactuated system. This design combined two standard techniques to generate a control law with excellent tracking performance under demanding conditions. Finally, this research constructed an interlaced $H^\infty$-filter based on a unique decomposition of the equations for the underactuated model. This approach allowed us to prove disturbance attenuation results for the closed-loop system when only imperfect state measurements were available for feedback.

Combined, these results demonstrate that we have made useful advances in the field of underactuated vehicle motion planning and control. Along the way, we have identified a host of additional research opportunities that could be addressed in the near future.

## 7.2 Potential Research Areas

We will sketch a range of potential research areas spawned from our work on motion planning and control for underactuated systems.

There are several ways to build upon our motion planning results. First, our results have focused on the planar vehicle case, but the basic motion planning algorithm could be adapted to handle three-dimensional scenarios as well. The key challenges with this approach would include finding a way to efficiently generate the initial polynomial curve, as well as using the equations of motion for the system to derive a feasible trajectory from the initial curve. Another research area would be to further explore the rapidly-exploring random trees approach to solving motion planning problems for underactuated systems. The current approach could be improved by refining the algorithm and through more efficient software coding to help reduce the processing time. A third motion planning contribution would be to use our basic planning algorithm to investigate the decoupled approach to solving the multiple vehicle planning problem. The decoupled approach

offers computational advantages over the centralized method we employed and may offer more elegant solutions to the problem.

The list of potential research areas related to the controller design is quite extensive and offers some significant challenges. Two control design techniques that could be applied to the underactuated vehicle tracking problem are pseudolinearization and gain scheduling. Pseudolinearization may provide a reasonable solution to the problem, but requires developing a transformation to make the system linearizable about the desired trajectory. A gain scheduling approach can likely solve this problem as well, but the key elements in the design will be to select appropriate operating points that characterize different types of motion and to determine how to schedule among the operating points. These control techniques are closely related and it may be efficient to conduct a combined study of the two approaches for underactuated vehicles.

In Chapter 1, we mentioned three restrictions on our research to help limit the scope of the project. Each of these restrictions points to a new research direction for underactuated vehicle control and we will comment on two of three areas. For realistic vehicles, adaptive control techniques are very practical and should be explored. Along with adaptive control, parameter estimation is an important aspect of the problem. Both of these techniques would help generate controllers that could respond to changing conditions in the environment. The biologically inspired control techniques of neural networks, fuzzy logic, and genetic algorithms also offer practical methods for addressing the control problem. These approaches are especially useful when the underlying system is nonlinear and standard approaches do not yield a solution.

Another meaningful area of research would be to consider bounded control inputs or the effects of saturation on the actuators. The model used for our results did not have these limitations, but they represent practical situations that the controller should be able to handle. On a related note, extensions to our research could also account for

actuator dynamics. We assumed that the actuators could instantaneously achieve the commanded inputs. A more realistic model would account for the transient behavior of the actuators when developing a control signal.

The final two suggested research topics are items that we considered, but we were not able to develop complete solutions for them. It would be very helpful to prove a separation principle for the imperfect state measurement problem and use the principle to prove local exponential stability for the closed-loop system. This line of investigation would allow us to describe a region of attraction for the imperfect state measurement case and allow system designers to know exactly when this control technique would be useful for a particular system. The other area that is worth exploring in more detail is how to use the underactuated controllers to recover from failures with a fully actuated system. This research could consider the complete design and develop experiments to demonstrate the approach.

This outline of additional research areas serves as a first step to extend our contributions on planning and controlling the motion of an underactuated vehicle.

# APPENDIX A

# CUBIC SPLINES AND HODOGRAPH CURVES

## A.1 Introduction

This appendix describes two methods for generating polynomial position curves for the motion planning problem. The first technique uses cubic splines and the second approach uses the concept of Pythagorean hodograph curves. We will use both methods to solve the Hermite interpolation problem. Cubic splines are a standard technique and Pythagorean hodograph curves offer an alternative method for generating polynomials. Pythagorean hodograph curves are valuable because they allow for an analytical computation of curvature, length, and bending energy in a candidate path. We can also directly calculate offset paths located a fixed distance from the curve, which may help in analyzing the performance of the motion planning algorithm. We solve the Pythagorean hodograph curve problem using polynomials with complex numbers to provide a compact representation. We used Mathematica to perform the calculations and display the results.

## A.2 Problem Formulation

The problem we would like to solve is to find a planar $x$-$y$ position path that connects two points and satisfies tangent conditions at each point. This problem is known as the

Hermite interpolation problem [99] and has a well-known solution if the path coordinates are characterized by cubic polynomials. This section will present the notation used to describe the problem. Section A.3 will provide the solution using cubic splines and Sections A.4 and A.5 will describe the second solution to the problem using Pythagorean hodograph curves. Section A.6 will provide a few examples to illustrate the Pythagorean hodograph technique for solving the Hermite interpolation problem.

Our objective is to find polynomials $x(t)$ and $y(t)$ such that the path generated by $\{x(t), y(t)\}$ with $t \in [0, 1]$ satisfies a set of boundary conditions. The time interval has been normalized to $[0, 1]$ without loss of generality. We define $\mathbf{r}(t) := \{x(t), y(t)\}$ and $\mathbf{r}'(t) := \{x'(t), y'(t)\}$ to simplify the notation for the parametric curve and its derivative. We are given the values of $\mathbf{r}(0)$, $\mathbf{r}(1)$, $\mathbf{r}'(0)$, and $\mathbf{r}'(1)$ as the boundary conditions for the problem. We note that we can easily compute these values from the information available to solve the motion planning problem presented in Chapter 3.

## A.3 Cubic Spline Solution

The solution for the cubic spline interpolation problem solves a set of simultaneous equations based on the polynomials for each coordinate. We have

$$x(t) = a_x t^3 + b_x t^2 + c_x t + d_x \tag{A.1}$$

$$y(t) = a_y t^3 + b_y t^2 + c_y t + d_y \tag{A.2}$$

and

$$x'(t) = 3a_x t^2 + 2b_x t + c_x \tag{A.3}$$

$$y'(t) = 3a_y t^2 + 2b_y t + c_y. \tag{A.4}$$

Consider the $x(t)$ coordinate first. If we evaluate $x(t)$ and $x'(t)$ each at $t = 0$ and $t = 1$, we have four equations. There are four unknown parameters $(a_x, b_x, c_x, d_x)$, so we can

solve for the parameters uniquely. In terms of the boundary conditions, the solution is

$$\begin{aligned} a_x &= 2[x(0) - x(1)] + x'(0) + x'(1) \\ b_x &= 3[x(1) - x(0)] - 2x'(0) - x'(1) \\ c_x &= x'(0) \\ d_x &= x(0). \end{aligned}$$

We can construct a similar solution for the $y$ coordinate.

In practice, we generate the cubic spline using a single MATLAB command, `spline.m`, which returns the polynomials for the $x$ and $y$ coordinates. The MATLAB routine uses the method of divided difference to find the polynomials that represent the spline. The MATLAB documentation and de Boor's text on splines [100] provide more detailed information about the approach. The MATLAB function `spline.m` is equivalent to solving a system of four linear equations described above for each coordinate.

The solution presented above using cubic splines readily generates a candidate path, but it may be difficult to find analytical expressions for curvature along the path or path length. We can numerically approximate these values for specific problems, but it is difficult to find general solutions. Pythagorean hodograph curves offer an alternative approach that solves the Hermite interpolation problem and provides analytical expressions for curvature and path length.

## A.4 Pythagorean Hodograph Curves

Pythagorean hodograph curves were introduced by Farouki and Sakkalis [101] as a mathematical concept that could assist with computer modeling and graphics. Farouki and Neff [102] used Pythagorean hodograph curves to solve the Hermite interpolation problem, and Farouki [103] further developed the concept to compute the bending energy

of the curves. Bruyninckx and Reynaerts [104] applied Pythagorean hodograph curves to path planning for mobile and hyper-redundant robots, which is closely related to the path planning problem. This section will explain the general concept of Pythagorean hodograph curves and describe how some of their properties are particularly useful for path planning.

The hodograph of the planar curve $\mathbf{r}(t) = \{x(t), y(t)\}$ is the curve of velocities given by $\mathbf{r}'(t) := \{x'(t), y'(t)\}$. A planar curve is a Pythagorean hodograph (PH) curve if its hodograph satisfies the algebraic Pythagorean constraint

$$x'^2(t) + y'^2(t) = \sigma^2(t) \tag{A.5}$$

for some polynomial $\sigma(t)$. To solve for the Pythagorean hodograph curves we will express the curves as complex-valued polynomials of the form $x(t) + iy(t)$. This approach will allow us to develop a more compact solution and to exploit the complex algebra capabilities of standard computational software. Before we present the solution using complex notation, we consider some facts about PH curves using real polynomials.

Following [102], we let $u(t)$, $v(t)$, and $w(t)$ be nonzero real polynomials such that $u(t)$ and $v(t)$ are relatively prime and not both constants. If we form a parametric curve with the derivatives

$$x'(t) = w(t)[u^2(t) - v^2(t)] \quad \text{and} \quad y'(t) = 2w(t)u(t)v(t) \tag{A.6}$$

then the curve will be a Pythagorean hodograph curve with

$$\sigma(t) = w(t)[u^2(t) + v^2(t)]. \tag{A.7}$$

We will use this type of construction with the complex polynomials when we present the detailed solution.

Pythagorean hodograph curves are different from other polynomial curves because we can express the parametric speed, unit tangent, unit normal, and curvature as rational

functions of parameter $t$. The parametric speed is given by

$$\frac{ds}{dt} = \sqrt{x'^2(t) + y'^2(t)} = |\mathbf{r}'(t)| = \sigma(t).$$

Having a polynomial expression for the speed will make it easy to compute the length of each curve without numerical approximations. We can express the unit tangent $\mathbf{t}$, unit normal $\mathbf{n}$, and curvature $\kappa$ in terms of the polynomials $u$, $v$, and $w$ (assuming $w$ is nonnegative) as

$$\mathbf{t} = \frac{(u^2 - v^2, 2uv)}{u^2 + v^2}, \quad \mathbf{n} = \frac{(2uv, v^2 - u^2)}{u^2 + v^2}, \quad \kappa = \frac{2(uv' - u'v)}{w(u^2 + v^2)^2}. \tag{A.8}$$

We can use the unit normal to find offset curves located at a signed distance $d$ from the original curve using

$$\mathbf{r}_d(t) = \mathbf{r}(t) + d\mathbf{n}(t). \tag{A.9}$$

These offset curves will help us describe a region around a desired path that the vehicle must stay within to meet performance objectives. Again, having a rational polynomial expression will simplify the calculations.

The lowest-order Pythagorean hodograph curves that satisfy the Hermite interpolation constraints are the quintic (fifth-order) splines. The quintics do not give a unique solution and there are always four curves that satisfy the constraints. Of the four curves, there is one that gives a reasonable shape without too much turning or bending [102]. We would like to find all of the PH curves that satisfy a set of boundary conditions and then automatically select the one that has the most appropriate shape.

Using another standard representation, we can express the curve $\mathbf{r}(t)$ in Bernstein-Bézier spline form:

$$\mathbf{r}(t) = \sum_{k=0}^{n} \mathbf{p}_k \binom{n}{k} (1-t)^{n-k} t^k, \qquad t \in [0, 1]. \tag{A.10}$$

We can solve for the first two and last two control points $\mathbf{p}_k$ using the boundary conditions for the curve given by $\mathbf{r}(0)$, $\mathbf{r}(1)$, $\mathbf{r}'(0)$, and $\mathbf{r}'(1)$. The resulting expressions are

$$\mathbf{p}_0 = \mathbf{r}(0), \qquad \mathbf{p}_1 = \mathbf{r}(0) + \frac{1}{n}\mathbf{r}'(0),$$
$$\mathbf{p}_{n-1} = \mathbf{r}(1) - \frac{1}{n}\mathbf{r}'(1), \qquad \mathbf{p}_n = \mathbf{r}(1).$$

For quintic PH curves, we can solve for control points $\mathbf{p}_0$, $\mathbf{p}_1$, $\mathbf{p}_4$, and $\mathbf{p}_5$ directly. Farouki and Neff [102] present a procedure for finding the other two control points using real polynomials, but we will not discuss it here.

This section has presented the basic concept of a Pythagorean hodograph curve and some of its valuable properties. The next section will describe how to find the four PH curves that solve the Hermite interpolation problem and how to select the most reasonable curve for the path planning problem.

## A.5 Pythagorean Hodograph Solution

We now present the solution to the Hermite interpolation problem using Pythagorean hodograph curves and a complex number representation. As mentioned above, the solution can be stated using only real numbers, but the complex number approach allows us to take advantage of some of the computational features of the standard computational software packages, such as MATLAB or Mathematica. These results were originally reported by Farouki and Neff [102] and were summarized by Bruyninckx and Reynaerts [104]. We also extract some of the results from Farouki [103] to compute the length and energy of the curves.

We will represent the parametric curves $\mathbf{r}(t)$ as a complex-valued polynomial $x(t) + iy(t)$ and use the standard algebraic manipulations for complex numbers. The results depend on a key lemma from Farouki and Neff [102]:

**Lemma A.1** *In the complex representation, the regular Pythagorean hodograph curves correspond to those curves whose hodographs are perfect squares of complex polynomials having relatively prime real and imaginary parts.*

**Proof.** The square of the complex polynomial $\mathbf{p}(t) = u(t) + iv(t)$ is $\mathbf{p}^2(t) = u^2(t) - v^2(t) + i2u(t)v(t)$, whose real and imaginary parts are seen to be of the Pythagorean form (A.6) with $w(t) = 1$. Conversely, any complex hodograph having real and imaginary parts of the form (A.6) with $w(t) = 1$ is simply the square of the complex polynomial $\mathbf{p}(t) = u(t) + iv(t)$. Moreover, we note that the greatest common divisor of $u$ and $v$, $\gcd(u, v)$, is equal to one if and only if $\gcd(u^2 - v^2, 2uv) = 1$. □

Using Lemma A.1, the hodograph of any regular PH quintic curve can be written as

$$\mathbf{r}'(t) = \mathbf{k}\,[(t - \mathbf{a})(t - \mathbf{b})]^2 \tag{A.11}$$

where $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{k}$ are complex numbers. To ensure we do not have a degenerate solution, the numbers $\mathbf{a}$ and $\mathbf{b}$ should have nonzero imaginary parts and they should not be conjugates of each other.

The first step in the solution process is to convert the boundary conditions to standard form

$$\mathbf{r}(0) = 0,\ \mathbf{r}'(0) = \mathbf{d}_0 \quad \text{and} \quad \mathbf{r}(1) = 1,\ \mathbf{r}'(1) = \mathbf{d}_1$$

by subtracting $\mathbf{r}(0)$ from the end points and dividing the end points and end derivatives by $\mathbf{r}(1) - \mathbf{r}(0)$. We will revert back to normal form by reversing these two steps at the end.

Next, we integrate the expression (A.11) and use the boundary condition $\mathbf{r}(0) = 0$ to determine the constant of integration to get

$$\mathbf{r}(t) = \frac{\mathbf{k}}{30}\left[(t - \mathbf{a})^5 - 5(t - \mathbf{a})^4(t - \mathbf{b}) + 10(t - \mathbf{a})^3(t - \mathbf{b})^2\right] + \mathbf{c} \tag{A.12}$$

where the constant is

$$\mathbf{c} = \frac{\mathbf{k}}{30}(\mathbf{a}^5 - 5\mathbf{a}^4\mathbf{b} + 10\mathbf{a}^3\mathbf{b}^2). \tag{A.13}$$

We have now reduced the problem to finding the complex constants $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{k}$ such that the curve defined by (A.12) and (A.13) satisfies $\mathbf{r}'(0) = \mathbf{d}_0$, $\mathbf{r}'(1) = \mathbf{d}_1$, and $\mathbf{r}(1) = 1$. The solution is given as the following proposition taken from [102].

**Proposition A.1** *Let $\rho$ be either of the two complex numbers defined by*

$$\rho^2 = \frac{\mathbf{d}_0}{\mathbf{d}_1} \tag{A.14}$$

*and hence let $\alpha$ be either of the two solutions to the quadratic equation*

$$\alpha^2 - 3(1+\rho)\alpha + 6\rho^2 + 2\rho + 6 - \frac{30}{\mathbf{d}_1} = 0. \tag{A.15}$$

*Then if $\mu_1$ and $\mu_2$ are the two roots of*

$$\mu^2 - \alpha\mu + \rho = 0 \tag{A.16}$$

*the values of $\mathbf{a}$ and $\mathbf{b}$ in* (A.12) *are given by*

$$\mathbf{a} = \frac{\mu_1}{\mu_1 + 1} \quad \text{and} \quad \mathbf{b} = \frac{\mu_2}{\mu_2 + 1}. \tag{A.17}$$

*The corresponding value of $\mathbf{k}$ is then*

$$\mathbf{k} = \frac{\mathbf{d}_0}{\mathbf{a}^2\mathbf{b}^2} = \frac{d_1}{(1-\mathbf{a})^2(1-\mathbf{b})^2} \tag{A.18}$$

*and $\mathbf{c}$ is given in terms of $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{k}$ by expression* (A.13).

Farouki and Neff [102] present a brief proof for this proposition which consists of verifying that the results satisfy the boundary conditions after some algebraic manipulations. The two solutions from (A.14) and the two solutions from (A.15) combine to give

the four possible values for $\alpha$ and lead to four different curves that satisfy the boundary conditions.

We can also find expressions for the curve length and curvature given as

$$l = |\mathbf{k}|^2 \left[ |\mathbf{a}|^2|\mathbf{b}|^2 - |\mathbf{a}|^2\mathrm{Re}(\mathbf{b}) - |\mathbf{b}|^2\mathrm{Re}(\mathbf{a}) + \frac{|\mathbf{a}|^2 + 4\mathrm{Re}(\mathbf{a})\mathrm{Re}(\mathbf{b}) + |\mathbf{b}|^2}{3} + \frac{1}{5} \right] \tag{A.19}$$

and

$$\kappa(t) = \frac{\mathrm{Im}(\dot{\bar{\mathbf{r}}}\ddot{\mathbf{r}})}{|\dot{\mathbf{r}}|^3} \tag{A.20}$$

where

$$\dot{\mathbf{r}} = \mathbf{k}(t-\mathbf{a})^2(t-\mathbf{b})^2 \tag{A.21}$$

$$\ddot{\mathbf{r}} = 2\mathbf{k}(t-\mathbf{a})(t-\mathbf{b})(2t-\mathbf{a}-\mathbf{b}). \tag{A.22}$$

Note that $|\cdot|$ is the absolute value and $\bar{\cdot}$ is the complex conjugate of a complex number.

The bending energy of the curve is given by

$$E = \int \kappa^2(s)\, ds. \tag{A.23}$$

A detailed expression for the total bending energy in terms of the parameters $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{k}$ is given by Farouki [103, p. 234-235] and is not repeated here. To select a single curve from the four available, we will choose the PH curve with the minimum total bending energy as the candidate curve for the path planning process. Farouki and Neff [102] propose using the curve with the smallest absolute rotation index given by

$$R_{\mathrm{abs}} = \frac{1}{2\pi} \int |\kappa|\, ds \tag{A.24}$$

as an alternative criterion to bending energy. The two criteria will often give the same results. The bending energy criterion is more appropriate for the path planning problem because it will avoid regions of high curvature. The rotation index criterion avoids

solutions that loop, but allows higher curvatures. We consider making a low curvature loop a more acceptable path than one with less overall turning, but with regions of higher curvature.

Bruyninckx and Reynaerts [104] present a more general performance index given by

$$\int_0^l \kappa^2(s)\, ds + \gamma \int_0^l ds \tag{A.25}$$

where $\gamma$ is a weighting factor between bending energy and arc length. We have all of the tools in place to calculate this performance index, but did not use it at this time.

This section has described how to compute the Pythagorean hodograph curves that solve the Hermite interpolation problem using a complex polynomial representation. We also explained how to use the total bending energy as a selection criterion to pick an appropriate curve for path planning from the four available solutions. The next section will illustrate the approach with a few simple examples.

## A.6 Example Pythagorean Hodograph Curves

We used Mathematica to calculate the solution to the Hermite interpolation problem for several examples. Bruyninckx and Reynaerts used MATLAB to generate the examples shown in [104] and generously made their code available to us. We implemented their MATLAB code and made minor improvements in it to replicate some of the results in [104]. The MATLAB code does not explicitly solve for the quintic polynomials for $x(t)$ and $y(t)$ and instead relies on numerical evaluation to generate the curves. We corrected this deficiency by implementing the algorithm in Mathematica and using the software to further improve the results. All of the simulations presented here were completed using Mathematica and the results verified with the original MATLAB code.

Figure A.1 is replica of Figure 3 in [104] and shows the four PH curves satisfying the conditions $\mathbf{r}(0) = \{0, 0\}$, $\mathbf{r}(1) = \{1, 1\}$, $\mathbf{r}'(0) = 3 \times \{1, 0\}$, and $\mathbf{r}'(1) = 2 \times \{-1, 1\}$. Figure A.2 shows the corresponding curvature profiles for these four curves, and it is fairly obvious which one has the lowest bending energy (i.e., the lower-left plot). Figure A.3 shows the PH curve with the lowest bending energy along with two offset curves located a distance $d = 0.41$ units from the original curve. This offset distance represents the maximum value for a smooth offset curve. Choosing a larger offset distance would create a boundary curve that crosses over itself because of the shape of the original curve.

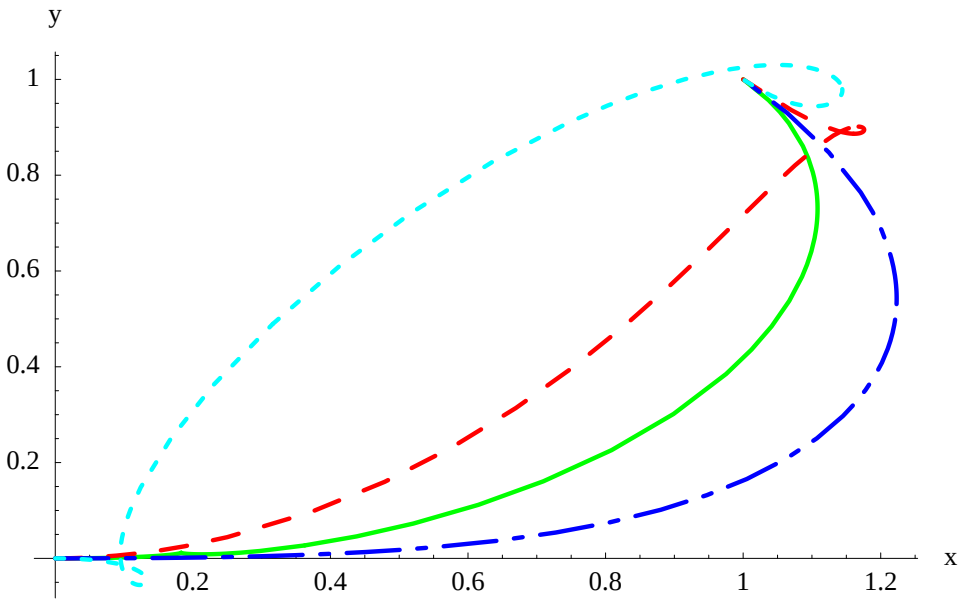



**Figure A.1** Four example Pythagorean hodograph curves. The curves satisfy the boundary conditions $\mathbf{r}(0) = \{0, 0\}$, $\mathbf{r}(1) = \{1, 1\}$, $\mathbf{d}_0 = 3 \times \{1, 0\}$, and $\mathbf{d}_1 = 2 \times \{-1, 1\}$. Only one curve represents a reasonable candidate path for motion planning.

We present two more figures that illustrate the types of solutions available with Pythagorean hodograph curves. Figure A.4 shows the four PH curves for $\mathbf{r}(0) = \{0, 0\}$, $\mathbf{r}(1) = \{0, 4\}$, $\mathbf{r}'(0) = 3 \times \{1, 0\}$, and $\mathbf{r}'(1) = 2 \times \{1, 0\}$. Figure A.5 shows the four PH curves for $\mathbf{r}(0) = \{0, 0\}$, $\mathbf{r}(1) = \{1, 1\}$, $\mathbf{r}'(0) = 2 \times \{1, 0\}$, and $\mathbf{r}'(1) = 2 \times \{1, 0\}$. In all three examples, the curve with the minimum bending energy represents the most

reasonable path to use as the starting point for planning the motion of a vehicle. It is very easy to identify the appropriate curve and, since it is a polynomial position curve, there is no difference between the Pythagorean hodograph curve and the cubic spline for the remainder of the motion planning algorithm.

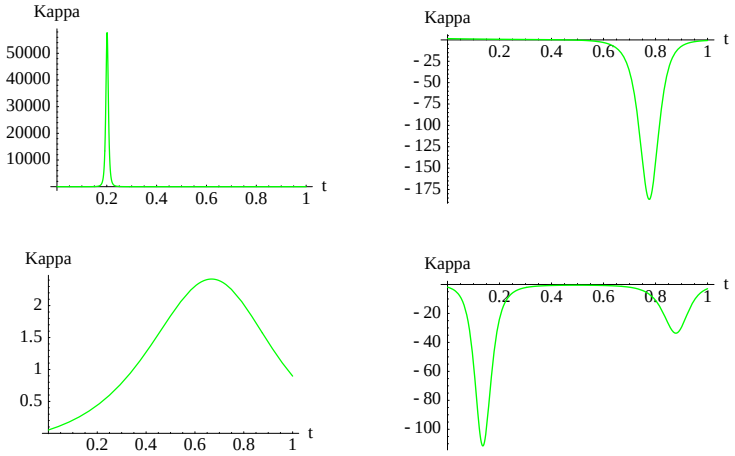



**Figure A.2** Curvature profiles for the four Pythagorean hodograph curves. Notice the large variation in the magnitude of the curvatures. The lower left plot corresponds to the minimum energy curve.

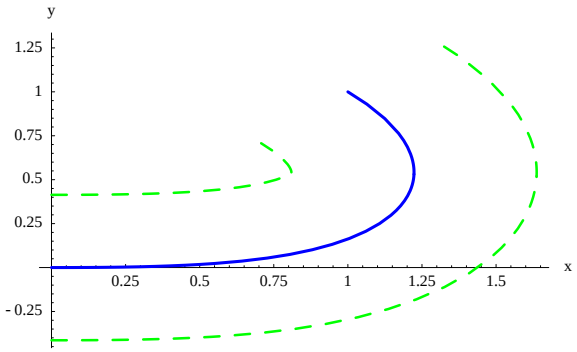



**Figure A.3** Minimum energy curve with offset boundaries. The energy of this curve is 3.99 units and the length is 2.01 units. The offset curves are located 0.41 units away from the original curve. The maximum curvature for this curve is $\kappa = 2.42$ and we placed the offset curves at a distance equal to the minimum radius of curvature $\rho = 1/\kappa$. Constructing offset curves at a distance more than $\rho_{\min}$ causes a distorted offset boundary.

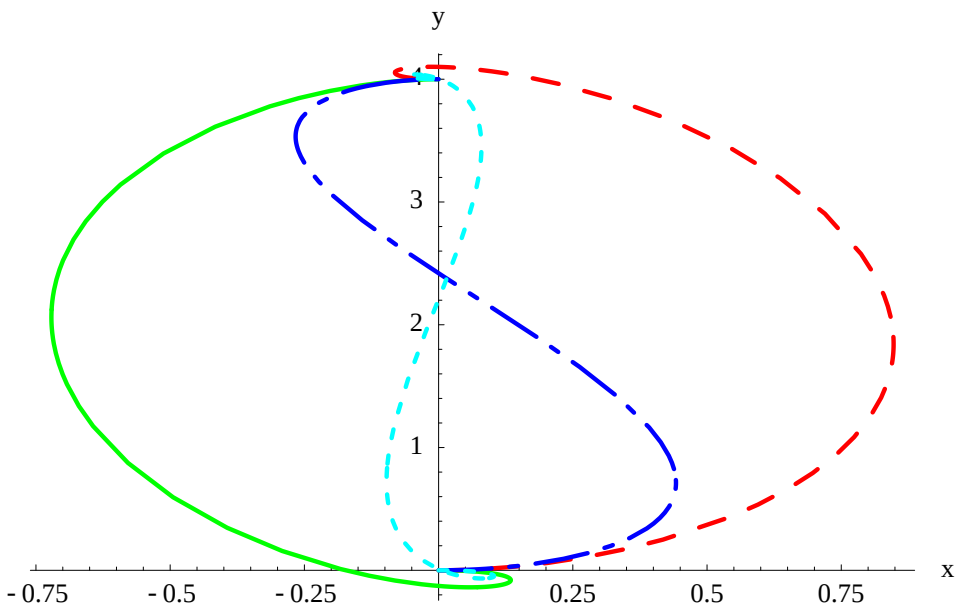



**Figure A.4** Four example Pythagorean hodograph curves. The curves satisfy the boundary conditions $r(0) = \{0, 0\}$, $r(1) = \{0, 4\}$, $\mathbf{d}_0 = 3 \times \{1, 0\}$, and $\mathbf{d}_1 = 2 \times \{1, 0\}$.

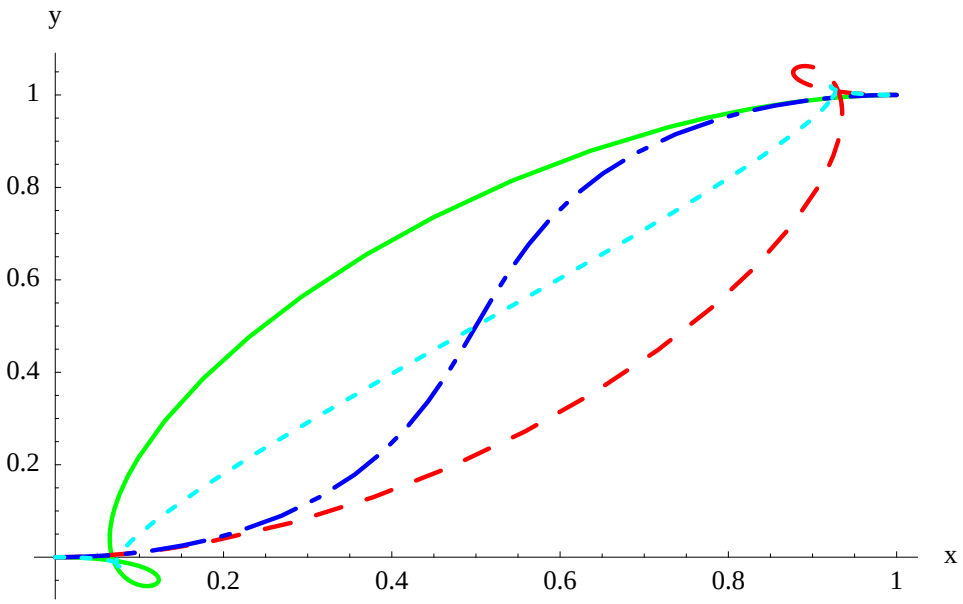



**Figure A.5** Four example Pythagorean hodograph curves. The curves satisfy the boundary conditions $r(0) = \{0, 0\}$, $r(1) = \{1, 1\}$, $\mathbf{d}_0 = 2 \times \{1, 0\}$, and $\mathbf{d}_1 = 2 \times \{1, 0\}$.

## A.7 Summary

This appendix presented two techniques for solving the Hermite interpolation problem. The first technique is relatively common and relies on cubic splines to find the solution. The second approach uses the concept of Pythagorean hodograph curves, which have some attractive properties for motion planning problems. We also illustrated three examples of the Pythagorean hodograph curve approach.

# REFERENCES


[1] J.-M. Godhavn, "Topics in nonlinear motion control: Nonholonomic, underactuated and hybrid systems," Ph.D. dissertation, Norwegian University of Science and Technology, N-7034 Trondheim, Norway, 1997.

[2] M. Reyhanoglu, A. van der Schaft, N. H. McClamroch, and I. Kolmanovsky, "Dynamics and control of a class of underactuated mechanical systems," *IEEE Transactions on Automatic Control*, vol. 44, pp. 1663–1671, Sept. 1999.

[3] K. Y. Pettersen and H. Nijmeijer, "State tracking control of an underactuated surface vessel," in *Proceedings of the 37th IEEE Conference on Decision and Control*, (Tampa, Florida), IEEE, Dec. 1998, pp. 4561–4566.

[4] T. I. Fossen, *Guidance and Control of Ocean Vehicles*. Chichester: John Wiley & Sons, 1994.

[5] J.-C. Latombe, "Motion planning: A journey of robots, molecules, digital actors, and other artifacts," *International Journal of Robotics Research*, vol. 18, pp. 1119–1128, Nov. 1999.

[6] J.-C. Latombe, *Robot Motion Planning*. Boston: Kluwer Academic Publishers, 1991.

[7] Y. K. Hwang and N. Ahuja, "Gross motion planning—a survey," *ACM Computing Surveys*, vol. 24, pp. 219–291, Sept. 1992.

[8] Z. Li and J. Canny, *Nonholonomic Motion Planning*. Boston: Kluwer Academic Publishers, 1993.

[9] J.-P. Laumond, *Robot Motion Planning and Control*, vol. 229. London: Springer-Verlag, 1998.

[10] K. Djouani and Y. Hamam, "Minimum time-energy trajectory planning for automatic ship berthing," *IEEE Journal of Oceanic Engineering*, vol. 20, pp. 4–12, Jan. 1995.

[11] K. Djouani and Y. Hamam, "Optimal constrained multi-criteria trajectory planning for ship maneuvering and control," *Proceedings of IEEE International Conference on Control and Applications*, vol. 2, pp. 525–530, Sept. 1993.

[12] H. B. Keller, *Numerical Solution of Two Point Boundary Value Problems*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1976.

[13] C.-C. Yih and P. I. Ro, "Near-optimal motion planning for nonholonomic systems using multi-point shooting method," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, pp. 2943–2948, Apr. 1996.

[14] C. Fernandes, L. Gurvits, and Z. X. Li, "A variational approach to optimal nonholonomic motion planning," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 680–685, Apr. 1991.

[15] C. Fernandes, L. Gurvits, and Z. Li, "Near-optimal nonholonomic motion planning for a system of coupled rigid bodies," *IEEE Transactions on Automatic Control*, vol. 39, pp. 450–463, Mar. 1994.

[16] I. Spangelo and O. Egeland, "Trajectory planning and collision avoidance for underwater vehicles using optimal control," *IEEE Journal of Oceanic Engineering*, vol. 19, pp. 502–511, Oct. 1994.

[17] H. Arai, K. Tanie, and N. Shiroma, "Time-scaling control of an underactuated manipulator," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2619–2626, May 1998.

[18] H. Arai, K. Tanie, and N. Shiroma, "Time-scaling control of an underactuated manipulator," *Journal of Robotic Systems*, vol. 15, pp. 525–536, Sept. 1998.

[19] H. Arai, K. Tanie, and N. Shiroma, "Feedback control of a 3-DOF planar underactuated manipulator," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 703–709, Apr. 1997.

[20] H. Arai, K. Tanie, and N. Shiroma, "Nonholonomic control of a three-DOF planar underactuated manipulator," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 681–695, Oct. 1998.

[21] K. M. Lynch, N. Shiroma, H. Arai, and K. Tanie, "Motion planning for a 3-DOF robot with a passive joint," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 927–932, May 1998.

[22] D. B. Reister and S. M. Lenhart, "Time-optimal paths for high-speed maneuvering," *International Journal of Robotics Research*, vol. 14, pp. 184–194, Apr. 1995.

[23] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[24] H. J. Sussmann and G. Tang, "Shortest paths for the Reeds-Shepp car: a worked out example of the use of geometric techniques in nonlinear optimal control," Rutgers Center for Systems and Control, Rutgers University, Tech. Rep. SYCON-91-10, Sept. 1991.

[25] P. Souères and J.-P. Laumond, "Shortest paths synthesis for a car-like robot," *IEEE Transactions on Automatic Control*, vol. 41, pp. 672–688, May 1996.

[26] S. M. LaValle, "A game-theoretic framework for robot motion planning," Ph.D. dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, 1995.

[27] S. M. LaValle and S. A. Hutchinson, "Optimal motion planning for multiple robots having independent goals," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 912–925, Dec. 1998.

[28] C. Tomlin, G. J. Pappas, and S. Sastry, "Noncooperative conflict resolution air traffic management," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 2, (San Diego, CA), IEEE, Dec. 1997, pp. 1816–1821.

[29] C. Tomlin, G. J. Pappas, and S. Sastry, "Conflict resolution for air traffic management: a study in multiagent hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, pp. 509–521, Apr. 1998.

[30] S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, (Atlanta, GA), IEEE, May 1993, pp. 802–807.

[31] M. Khatib, H. Jaouni, R. Chatila, and J. P. Laumond, "Dynamic path modification for car-like nonholonomic mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, (Albuquerque, NM), IEEE, Apr. 1997, pp. 2920–2925.

[32] O. Brock and O. Khatib, "Real-time obstacle avoidance and motion coordination in a multi-robot workcell," in *International Symposium on Assembly and Task Planning*, (Porto, Portugal), IEEE, July 1999, pp. 274–279.

[33] J. Barraquand and J.-C. Latombe, "Robot motion planning: A distributed representation approach," *International Journal of Robotics Research*, vol. 10, pp. 628–649, Dec. 1991.

[34] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, pp. 566–580, Aug. 1996.

[35] L. E. Kavraki and J.-C. Latombe, "Probabilistic roadmaps for robot path planning," in *Practical Motion Planning in Robotics: Current Approaches and Future Directions*, K. Gupta and P. del Pobil, Eds., John Wiley & Sons Ltd., 1998, pp. 33–53.

[36] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, (Minneapolis, MN), IEEE, Apr. 1996, pp. 3020–3025.

[37] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 166–171, Feb. 1998.

[38] P. Švestka and M. H. Overmars, "Motion planning for car-like robots using a probabilistic learning approach," Utrecht University, the Netherlands, Tech. Rep. UU-CS-1994-33, Aug. 1994.

[39] P. Švestka and M. H. Overmars, "Coordinated path planning for multiple robots," Utrecht University, the Netherlands, Tech. Rep. UU-CS-1996-43, 1996.

[40] J. Barraquand, L. Kavraki, J.-C. Latombe, R. Motwani, T.-Y.Li, and P. Raghavan, "A random sampling scheme for path planning," *International Journal of Robotics Research*, vol. 16, pp. 759–774, Dec. 1997.

[41] S. M. LaValle and J. J. Kuffner, Jr., "Randomized kinodynamic planning," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 473–479, May 1999.

[42] D. Hsu, J.-C. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," *International Journal of Computational Geometry & Applications*, vol. 9, pp. 496–512, Oct. 1999.

[43] R. Kindel, D. Hsu, J.-C. Latombe, and S. Rock, "Kinodynamic motion planning amidst moving obstacles." To appear in IEEE International Conference on Robotics and Automation, 2000.

[44] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," in *Workshop on Algorithmic Foundations of Robotics*, (Hanover, NH), WAFR 2000, Mar. 2000.

[45] J. Hu, J. Lygeros, M. Prandini, and S. Sastry, "Aircraft conflict prediction and resolution using Brownian motion," in *Proceedings of the IEEE Conference on Decision and Control*, (Phoenix, AZ), IEEE, Dec. 1999, pp. 2438–2443.

[46] P. Leven and S. Hutchinson, "Toward real-time path planning in changing environments," in *Workshop on Algorithmic Foundations of Robotics*, (Hanover, NH), WAFR 2000, Mar. 2000, pp. 393–406.

[47] K. Y. Pettersen and H. Nijmeijer, "Underactuated ship tracking control: Theory and experiments." Submitted for publication, 1999.

[48] Z. P. Jiang and H. Nijmeijer, "A recursive technique for tracking control of nonholonomic systems in chained form," *IEEE Transactions on Automatic Control*, vol. 44, pp. 265–279, Feb. 1999.

[49] K. Y. Pettersen and H. Nijmeijer, "Global practical stabilization and tracking for an underactuated ship - a combined averaging and backstepping approach," in *IFAC Conference on Systems Structure and Control*, IFAC, July 1998, pp. 59–64.

[50] K. Y. Pettersen and H. Nijmeijer, "Output feedback tracking control for ships," in *New Directions in Nonlinear Observer Design*, H. Nijmeijer and T. I. Fossen, Eds., Springer, 1999, pp. 311–334.

[51] J.-M. Godhavn, "Nonlinear tracking of underactuated surface vessels," in *Proceedings of the 35th IEEE Conference on Decision and Control*, vol. 1, IEEE, Dec. 1996, pp. 975–980.

[52] G. J. Toussaint, T. Başar, and F. Bullo, "Tracking for nonlinear underactuated surface vessels with generalized forces." *Proceedings of the Conference on Control Applications*, 2000, (to appear).

[53] N. E. Leonard, "Compensating for actuator failures: Dynamics and control of underactuated underwater vehicles," in *9th International Symposium on Unmanned Untethered Submersible Technology*, Sept. 1995, pp. 168–177.

[54] N. E. Leonard, "Stabilization of steady motions of an underwater vehicle," in *Proceedings of the 35th IEEE Conference on Decision and Control*, vol. 1, IEEE, Dec. 1996, pp. 961–966.

[55] N. E. Leonard, "Stabilization of underwater vehicle dynamics with symmetry-breaking potentials," *Systems & Control Letters*, vol. 32, pp. 35–42, Oct. 1997.

[56] F. Bullo and N. E. Leonard, "Motion control for underactuated mechanical systems on Lie groups," in *Proceedings of the 1997 European Control Conference*, July 1997.

[57] F. Bullo and N. E. Leonard, "Motion primitives for stabilization and control of underactuated vehicles," in *Nonlinear Control Systems Design Symposium 1998*, vol. 1, IFAC, July 1998, pp. 133–138.

[58] F. Bullo, N. E. Leonard, and A. D. Lewis, "Controllability and motion algorithms for underactuated Lagrangian systems on Lie groups." Caltech-CDS Technical Report 97-013, Nov 1997. To appear in IEEE Transactions on Automatic Control.

[59] F. Bullo and R. M. Murray, "Tracking for fully actuated mechanical systems: A geometric framework," *Automatica*, vol. 35, pp. 17–34, Jan. 1999.

[60] F. Bullo, "Stabilization of relative equilibria for systems on Riemannian manifolds," in *Proceedings of the 1999 American Control Conference*, (San Diego, CA), ACC, June 1999, pp. 1618–1622.

[61] J. Hauser, S. Sastry, and G. Meyer, "Nonlinear control design for slightly non-minimum phase systems: Application to V/STOL aircraft," *Automatica*, vol. 28, pp. 665–679, July 1992.

[62] P. Martin, S. Devasia, and B. Paden, "A different look at output tracking: Control of a VTOL aircraft," *Automatica*, vol. 32, pp. 101–107, Jan. 1996.

[63] M. Reyhanoglu, "Control and stabilization of an underactuated surface vessel," in *Proceedings of the 35th IEEE Conference on Decision and Control*, vol. 3, IEEE, Dec. 1996, pp. 2371–2376.

[64] M. Reyhanoglu, "Exponential stabilization of an underactuated autonomous surface vessel," *Automatica*, vol. 33, pp. 2249–2254, Dec. 1997.

[65] M. Reyhanoglu, A. van der Schaft, N. H. McClamroch, and I. Kolmanovsky, "Nonlinear control of a class of underactuated systems," in *Proceedings of the 35th IEEE Conference on Decision and Control*, vol. 2, IEEE, Dec. 1996, pp. 1682–1687.

[66] M. W. Spong, "Energy based control of a class of underactuated mechanical systems," in *Proceedings of the 13th World Congress, International Federation of Automatic Control*, vol. F, Pergamon, June 1997, pp. 431–435.

[67] M. W. Spong, "Underactuated mechanical systems," in *Control Problems in Robotics and Automation*, B. Siciliano and K. P. Valavanis, Eds., Berlin, Germany: Springer-Verlag, Dec. 1998, pp. 135–150.

[68] A. Isidori, *Nonlinear Control Systems*, Third ed. Great Britain: Springer, 1995.

[69] S. Devasia, D. Chen, and B. Paden, "Nonlinear inversion-based output tracking," *IEEE Transactions on Automatic Control*, vol. 41, pp. 930–942, July 1996.

[70] M. Krstić, I. Kanellakopoulos, and P. Kokotović, *Nonlinear and Adaptive Control Design*. New York: John Wiley & Sons, Inc., 1995.

[71] H. K. Khalil, *Nonlinear Systems*, Second ed. Upper Saddle River, NJ: Prentice Hall, 1996.

[72] C. Reboulet and C. Champetier, "A new method for linearizing non-linear systems: the pseudolinearization," *International Journal of Control*, vol. 40, pp. 631–638, Oct. 1984.

[73] A. J. van der Schaft, "On a state space approach to nonlinear $H^\infty$ control," *Systems & Control Letters*, vol. 16, p. 1991, Jan. 1991.

[74] A. J. van der Schaft, "$L_2$-gain analysis of nonlinear systems and nonlinear state-feedback $H^\infty$ control," *IEEE Transactions on Automatic Control*, vol. 37, pp. 770–784, June 1992.

[75] A. J. van der Schaft, "Nonlinear state space $H^\infty$ control theory," in *Essays on Control: Perspectives in Theory and its Applications*, H. L. Trentelman and J. C. Willems, Eds., Boston, MA: Birkhäuser, 1993, pp. 153–190.

[76] T. Başar and P. Bernhard, *$H^\infty$-Optimal Control and Related Minimax Design Problems*, Second ed. Boston: Birkhäuser, 1995.

[77] A. Isidori, "Attenuation of disturbances in nonlinear control systems," in *Systems, Models and Feedback: Theory and Applications*, A. Isidori and T. J. Tarn, Eds., Boston, MA: Birkhäuser, 1992, pp. 275–300.

[78] A. Isidori and C. I. Byrnes, "Output regulation of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 35, pp. 131–140, Feb. 1990.

[79] A. J. Krener, "The construction of optimal linear and nonlinear regulators," in *Systems, Models and Feedback: Theory and Applications*, A. Isidori and T. J. Tarn, Eds., Boston, MA: Birkhäuser, 1992, pp. 301–322.

[80] J. A. Ball, P. Kachroo, and A. J. Krener, "$H^\infty$ tracking control for a class of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 44, pp. 1202–1206, June 1999.

[81] A. Isidori and A. Astolfi, "Disturbance attenuation and $H^\infty$-control via measurement feedback in nonlinear system," *IEEE Transactions on Automatic Control*, vol. 37, pp. 1283–1293, Sept. 1992.

[82] J. A. Ball, J. W. Helton, and M. L. Walker, "$H^\infty$ control for nonlinear systems with output feedback," *IEEE Transactions on Automatic Control*, vol. 38, pp. 546–559, Apr. 1993.

[83] M. R. James and J. S. Baras, "Robust $H^\infty$ output feedback control for nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 40, pp. 1007–1017, June 1995.

[84] W.-M. Lu, "$H^\infty$-control of nonlinear time-varying systems with finite time horizon," *International Journal of Control*, vol. 64, pp. 241–262, May 1996.

[85] T. Başar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, Second ed. Philadelphia: Society for Industrial and Applied Mathematics, Classics in Applied Mathematics, 1999.

[86] Z. Pan and T. Başar, "Robustness of minimax controllers to nonlinear perturbations," *Journal of Optimization Theory and Applications*, vol. 87, pp. 631–678, Dec. 1995.

[87] V. H. L. Cheng, "A direct way to stabilize continuous-time and discrete-time linear time-varying systems," *IEEE Transactions on Automatic Control*, vol. AC-24, pp. 641–643, Aug. 1979.

[88] G. Walsh, D. Tilbury, S. Sastry, R. Murray, and J. P. Laumond, "Stabilization of trajectories for systems with nonholonomic constraints," *IEEE Transactions on Automatic Control*, vol. 39, pp. 216–222, Jan. 1994.

[89] M. R. Elgersma, G. Stein, M. R. Jackson, and J. Yeichner, "Robust controllers for space station momentum management," *IEEE Control Systems Magazine*, vol. 12, pp. 14–22, Oct. 1992.

[90] M. Elgersma, G. Stein, M. Jackson, and J. Yeichner, "Robust controllers for space station momentum management," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 3, (Brighton, England), IEEE, Dec. 1991, pp. 2206–2212.

[91] W. Kang, "Nonlinear $H^\infty$ control and its application to rigid spacecraft," *IEEE Transactions on Automatic Control*, vol. 40, pp. 1281–1285, July 1995.

[92] M. Dalsmo and O. Egeland, "State feedback $H^\infty$ suboptimal control of a rigid spacecraft," *IEEE Transactions on Automatic Control*, vol. 42, pp. 1186–1189, Aug. 1997.

[93] M. Dalsmo, "Contributions to nonlinear control and mathematical description of physical systems," Ph.D. dissertation, Norwegian University of Science and Technology, N-7034 Trondheim, Norway, 1997.

[94] M. Zasadzinski, E. Richard, M. F. Khelfi, and M. Darouach, "Disturbance attenuation and trajectory tracking via a reduced-order output feedback controller for robot manipulators," *Automatica*, vol. 34, pp. 1539–1546, Dec. 1998.

[95] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Iowa State University, Ames, IA, Tech. Rep. 98-11, Oct. 1998.

[96] N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones, and D. Vallejo, "Choosing good distance metrics and local planners for probabilistic roadmap methods," in *Proceedings of the IEEE International Conference on Robotics and Automation*, IEEE, 1998, pp. 630–637.

[97] L. E. Dubins, "On curves of minimal length with a constraint on average curvature and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, pp. 497–516, 1957.

[98] A. Loria, T. I. Fossen, and E. Panteley, "A cascaded approach to a separation principle for dynamic ship positioning." To appear in IEEE Transactions of Control Systems Technology, 2000.

[99] R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to Splines for use in Computer Graphics and Geometric Modeling.* Los Altos, CA: Morgan Kaufmann Publishers, Inc., 1987.

[100] C. de Boor, *A Practical Guide to Splines.* New York: Springer-Verlag, 1978.

[101] R. T. Farouki and T. Sakkalis, "Pythagorean hodographs," *IBM Journal of Research & Development*, vol. 34, pp. 736–752, Sept. 1990.

[102] R. T. Farouki and C. A. Neff, "Hermite interpolation by Pythagorean hodograph quintics," *Mathematics of Computation*, vol. 64, pp. 1589–1609, Oct. 1995.

[103] R. T. Farouki, "The elastic bending energy of Pythagorean-hodograph curves," *Computer Aided Geometric Design*, vol. 13, pp. 227–241, Apr. 1996.

[104] H. Bruyninckx and D. Reynaerts, "Path planning for mobile and hyper-redundant robots using pythagorean hodograph curves," *Proceedings of the 1997 8th International Conference on Advanced Robotics*, pp. 595–600, July 1997.

# VITA

**Gregory James Toussaint** was born in St. Johnsbury, VT, in 1967. He received a B.S. degree in electrical engineering from Cornell University, Ithaca, NY, in 1989, and an M.S. degree in systems engineering from the Air Force Institute of Technology, Wright-Patterson Air Force Base, Dayton, OH, in 1992. He is currently a captain on active duty in the United States Air Force pursuing a Ph.D. degree in electrical engineering at the University of Illinois at Urbana-Champaign. His research interests include nonlinear control theory, $H^\infty$-optimal control, and applications to underactuated vehicles. He completed the Air Force Reserve Officer Training Corps program as part of his undergraduate studies at Cornell University and was commissioned as an officer in the Air Force in 1989. His Air Force assignments include working as a laser communications engineer at Wright Laboratory (now part of the Air Force Research Laboratory) at Wright-Patterson Air Force Base, OH, being a program manager for the antisatellite program at Hanscom Air Force Base, MA, and teaching in the electrical engineering department at the U.S. Air Force Academy in Colorado Springs, CO. His planned future assignments include working at the National Air Intelligence Center at Wright-Patterson Air Force Base and returning to teach in the electrical engineering department at the Air Force Academy. He is married to the former Tricia C. Taylor from Dayton, OH, and their son, Andrew James, is one year old.