

Vision-Based Obstacle Avoidance for Multiple Vehicles Performing Time-Critical Missions

Amanda Dippold

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

Doctor of Philosophy
in
Aerospace Engineering

Dr. Naira Hovakimyan, Committee Co-Chair
Dr. Craig Woolsey, Committee Co-Chair
Dr. Mazen Farhood, Committee Member
Dr. Chris Hall, Committee Member

May 5th, 2009
Blacksburg, Virginia

Keywords: Path generation, path following, obstacle avoidance, path deformation, vision-based estimation, visual sensors, coordinated control, time-critical missions

Copyright 2009, Amanda Dippold

Vision-Based Obstacle Avoidance for Multiple Vehicles Performing Time-Critical Missions

Amanda Dippold

(ABSTRACT) This dissertation discusses vision-based static obstacle avoidance for a fleet of nonholonomic robots tasked to arrive at a final destination simultaneously. Path generation for each vehicle is computed using a single polynomial function that incorporates the vehicle constraints on velocity and acceleration and satisfies boundary conditions by construction. Furthermore, the arrival criterion and a preliminary obstacle avoidance scheme is incorporated into the path generation. Each robot is equipped with an inertial measurement unit that provides measurements of the vehicle's position and velocity, and a monocular camera that detects obstacles. The obstacle avoidance algorithm deforms the vehicle's original path around at most one obstacle per vehicle in a direction that minimizes an obstacle avoidance potential function. Deconfliction of the vehicles during obstacle avoidance is achieved by imposing a separation condition at the path generation level. Two estimation schemes are applied to estimate the unknown obstacle parameters. The first is an existing method known in the literature as Identifier-Based Observer and the second is a recently-developed fast estimator. It is shown that the performance of the fast estimator and its effect on the obstacle avoidance algorithm can be arbitrarily improved by the appropriate choice of parameters as compared to the Identifier-Based Observer method. Coordination in time of all vehicles is completed in an outer loop which adjusts the desired velocity profile of each vehicle in order to meet the simultaneous arrival constraints. Simulation results illustrate the theoretical findings.

Acknowledgments

First and foremost, I would like to thank God for everything and for moving in mysterious ways.

I would like to thank the AFOSR, ARO, and ONR for their support throughout that past years that has enabled me to complete my research and attend many beneficial conferences and exhibits.

I would like to thank my advisors, Professor Naira Hovakimyan and Professor Craig Woolsey for their investment, support, and encouragement. I thank my committee members Professors Mazen Farhood and Chris Hall for their commitment and dedication on my behalf. I am further grateful to Dr. Hall for his past advice and support throughout my graduate career. I would like to thank my colleagues at the Nonlinear Systems Lab and beyond - Chengyu Cao, Chris Cotting, Bob Kraus, Lili Ma, Nina Mahmoudian, Aaron Marcus, Justin Murtha, Konda Reddy, Chris Sonnenburg, and Laszlo Techy for their wisdom, kind words, and insight. I would like to thank the department of Aerospace and Ocean Engineering and its faculty and staff for their professionalism and knowledge.

I thank my husband, Sam, for his unwavering support and strength, without which I would not have had the courage to continue. All my love to Kane for always being the sunshine in my life. I would like to thank my mom, Katherine, who has created my opportunities in life through her sacrifice. I am grateful to the prayers and encouragement of my family and friends, who have always urged me to forge ahead and believed in my success.

Contents

1	Introduction	1
1.1	Motivation and Background	1
1.1.1	Obstacle Avoidance	2
1.1.2	Vision-Based Estimation	3
1.1.3	Vehicle Coordination	5
1.2	Objectives	8
1.3	Contributions	9
1.4	Organization	11
2	Path Following in 2D Space	13
2.1	Review of Paths	14
2.2	Path Generation	15
2.2.1	Direct Methods	16
2.2.2	Single Polynomial Path Generation in 2D	19
2.2.3	Analysis of Virtual Speed for a Single Vehicle	29
2.2.4	Path Generation and Bezier Curves	36

2.3	Mobile Robot Equations of Motion	44
2.4	2D Path Following	48
2.5	Summary	52
3	Nonholonomic Path Deformation for Obstacle Avoidance	54
3.1	Path Deformation	57
3.2	Path Deformation for Two-Wheeled Mobile Robots in the Presence of Un- known Obstacles	65
3.3	Path Deformation With Input Constraints Using Projection	77
3.4	Summary	87
4	Vision-Based Range Identification	88
4.1	Problem Formulation	88
4.2	Range Identification using IBO	91
4.3	Range Identification using fast estimator	94
4.4	Effect of Estimation Schemes on Obstacle Avoidance	100
4.5	Summary	113
5	Multi-Vehicle Coordination for Simultaneous Arrival	115
5.1	Graph Theory: Review of Important Concepts	117
5.2	Time-Critical Coordination With Bi-Directional Communication	123
5.3	Time-Critical Coordination With Uni-Directional Communication	128
5.4	An Illustrative Example	138

5.5	Summary	149
6	Conclusion	150
6.1	Summary	150
6.2	Recommendations For Future Work	151

List of Figures

2.1	Trajectories corresponding to 5th and 6th order polynomials.	22
2.2	Speed and acceleration profiles for 5th and 6th order polynomials.	22
2.3	Path generation for two vehicles.	29
2.4	Admissible vehicle speed and acceleration profiles.	30
2.5	Final value of the parameter s_f as p increases.	37
2.6	Comparison of vehicle trajectories with increasing p	38
2.7	Comparison of speed and acceleration profiles with increasing p	39
2.8	Trajectory comparison of polynomial path and optimal path.	43
2.9	Speed and acceleration profile comparison of polynomial path and optimal path.	44
2.10	Unicycle-type mobile robot.	47
2.11	Body-fixed and Frenet-Seret frames.	49
2.12	Graphical representation of shaping functions.	51
3.1	Points $P_1(t)$ and $P_2(t)$ are the projected edges of $R_d(t)$ expressed in the vehicle's body-fixed frame.	56

3.2	The iterative path deformation procedure. Obstacles impose repellent forces on the path so that the direction of deformation of the path ensures that an OA potential function $Y(\tau)$ diminishes.	59
3.3	Path deformation process modelled as a dynamic control system evolving on a τ time scale.	65
3.4	Plot of $U(s, \tau)$ for an obstacle with $\hat{R}_d(t_1) = 3, d_0 = 0.001$	66
3.5	Representation of the sets B_r, B_μ (dotted) and Ω_1, Ω_2 (solid).	71
3.6	Vehicle i and vehicle j can move to within at most ϵ_d of each other by the choice of Ξ	76
3.7	Pictorial representation of a convex function (solid line).	78
3.8	Pictorial representation of equation (8).	80
3.9	Pictorial representation of the projection operator.	81
3.10	Path deformation with and without projection.	86
4.1	Visual measurements $\alpha(t)$ and $\beta(t)$	89
4.2	Flow chart of range identification using the \mathcal{L}_1 fast estimator.	99
4.3	Estimation parameters using IBO and fast estimator.	106
4.4	Estimation errors using IBO and fast estimator.	106
4.5	IBO performance for various values of G	107
4.6	Error of IBO for various values of G	107
4.7	Fast estimator performance for various values of c	108
4.8	Error of fast estimator for various values of c	109
4.9	Zoomed errors of estimators for $G = 90, c = 2000$	110

4.10	Estimation parameters using IBO and fast estimator, second obstacle configuration.	110
4.11	Estimation errors using IBO and the fast estimator, second obstacle configuration.	111
4.12	IBO performance for various values of G , second obstacle configuration.	111
4.13	Error of IBO for various values of G , second obstacle configuration.	112
4.14	Fast estimator performance for various values of c , second obstacle configuration.	113
4.15	Error of the fast estimator for various values of c , second obstacle configuration.	113
4.16	Zoomed errors of estimators for $G = 50, c = 2000$, second obstacle configuration.	114
5.1	Undirected graph and an associated orientation.	119
5.2	Undirected and directed graphs: an example.	120
5.3	Symmetric Laplacian: an example	127
5.4	Desired speed profile of the leader with $v_{d_{\max}} = 5 \text{ m/s}$	128
5.5	Non-symmetric Laplacian: an example	131
5.6	Non-symmetric Laplacian: recovery of coordination objective with new controller (5.20)	133
5.7	Non-symmetric Laplacian: PD controller (5.23)	135
5.8	Eigenvalues corresponding to PI controllers (5.11) and (5.20) and root locus plot	137
5.9	Feasible paths	139
5.10	Feasible velocity and acceleration profiles	140
5.11	Desired path	141

5.12	Desired velocity and acceleration profiles	141
5.13	Vehicle separation distances from desired paths	142
5.14	Vehicle path following, no obstacle avoidance	143
5.15	Path following errors	144
5.16	Path deformation algorithm	144
5.17	Vehicle separation distances	145
5.18	Estimates of obstacle radii L_1, L_2, L_3	146
5.19	$\beta(t), \alpha(t)$ and their estimates $\hat{\beta}(t), \hat{\alpha}(t)$ for each vehicle	146
5.20	Vehicle trajectories with obstacle avoidance and estimation	147
5.21	Coordination parameters	148
5.22	Vehicle trajectories with obstacle avoidance, estimation, and coordination . .	148

List of Tables

2.1	Examples of computation of the coefficients of 5 th and 6 th order polynomial paths	25
2.2	Examples of computation of the coefficients of 5 th and 6 th order Bezier paths	41
3.1	Path deformation algorithm	64

Chapter 1

Introduction

1.1 Motivation and Background

The application of mobile vehicles for military and commercial purposes has increased dramatically in recent years. Unmanned aerial vehicles (UAVs) and autonomous underwater vehicles (AUV) platforms are becoming increasingly popular in the scientific community for many types of applications. These applications include - among others - military (e.g. surveillance and reconnaissance, enemy suppression and attack), biological (e.g. agricultural sampling, ocean floor exploration), protection and maintenance (e.g. firefighting, border patrol, search and rescue missions, hazard detection), and for transporting goods [53]. The ability of vehicles to perform their tasks with minimal human interaction, or *autonomy*, is advantageous in eliminating the risk to human operators. Furthermore, autonomous vehicles have the capability of operating in more “dull, dirty, or dangerous” environments than manned vehicles. Many areas of research exist in the development of autonomous vehicles, including the development of advanced control systems, seamless integration of existing sensors with onboard guidance, and “smart vehicles” that have the capability of making real-time decisions, to name a few. Some specific areas of interest to the scientific community are described in the following.

1.1.1 Obstacle Avoidance

Mobile vehicle obstacle avoidance is a well-studied topic. Conventionally, algorithms for OA have been divided into path planning algorithms and control algorithms [79]. Path planning approaches compute a continuous path based on a world model for the vehicle which avoids known obstacles. If the model is not well known or is incomplete, collisions may still occur. While path planning algorithms have the capability of generating optimal paths, they are computationally expensive [79]. Hence, the applicability of path planning methods in changing or unknown environments is limited. In contrast, control methods have been used to enable a vehicle to use sensing to close a feedback loop and interact with the environment for real-time collision avoidance. Sensor-based control approaches detect and avoid unknown obstacles in real-time. While path planning algorithms are computed prior to motion execution, control algorithms are determined based on feedback during the execution of the motion. Control algorithms are thus able to react to sudden changes in the environment. The drawback to these methods is that they may be suboptimal since only local information is used and cannot be used to solve a global problem [9].

A complete solution for vehicle OA exploits the benefits of both methods by formulating the problem as a local path planning problem. Local detours around sensed obstacles are generated while the vehicle follows the perturbed globally planned path, combining both methods. Artificial potential fields are widely used for generating these local detours. The original path is modified according to a potential field that a vehicle moves in. Its goal position represents an attractive force, while obstacles impose repellent forces [48]. Potential field methods (PFM) are an attractive approach to the obstacle avoidance problem due to the methods simplicity and efficiency in implementation [39]. However, it is well-known that PFM are vulnerable to undesired local minima. That is, attractive forces of a goal are balanced by repulsive forces from obstacles. This type of situation occurs primarily for concave obstacles, but can occur for any type of obstacle [83]. Another drawback to the PFM is the occurrence of oscillations when traveling down narrow passages or corridors [39]. To

address these issues, the authors in [83] introduce a special set of artificial potential functions which have only one minimum - the goal configuration. These special functions are referred to as *navigation functions*. Using these navigation functions generates an OA path on-line based on local sensor information. PFM is introduced in [79] for holonomic vehicles and extended in [9] for nonholonomic vehicles.

A PFM approach for integrating global path planning and real-time sensor-based vehicle control is the elastic band concept proposed in [79, 9]. An elastic band is a path that is deformable and collision-free. Initially, the shape of the elastic band corresponds to the original path generated by the path planning algorithm. The elastic band deforms in real time to avoid obstacles through forces imposed on it from an artificial potential function. The elastic band continues to deform as changes in the environment are detected by sensors, enabling the robot to accommodate uncertainties and react to unexpected and moving obstacles.

Other reactive OA approaches include the dynamic window approach, which modifies the speed profile of the vehicle in order to avoid obstacles. The dynamic window concept is merged in [72] with the navigation functions in [83] and its convergence properties are proved. Deformable virtual zone (DVZ) is another reactive OA approach where an ellipse is defined around the vehicle [55]. Any obstacle in close proximity deforms the ellipse and the vehicle corrects for this deformation in order to restore the ellipse boundary. In implementation, the DVZ method may lead to oscillations as the vehicle corrects for the deformation and tries to return to its original path as soon as the ellipse boundary is restored.

1.1.2 Vision-Based Estimation

An increasing demand has appeared for the development of unmanned or *autonomous* vehicles in an effort to reduce human risks. In order to minimize human interaction, vehicles must be able to perform mission-specific autonomous guidance and navigation using onboard sensors. Inertial measurement units augmented with global positioning systems (GPS) have traditionally been used for onboard sensing in a wide variety of missions for ground vehicles

and aerial vehicles [35]. GPS can continuously provide three-dimensional position, velocity, and time information under all weather conditions and receive low-power radio signals from satellites in orbit around the Earth. However, these low-power signals make it vulnerable to interference - intentional (e.g. jamming) or unintentional (e.g. electrical interference) - which can degrade system performance and render the vehicle unusable for the mission. The abundance of noise from radio-frequency emitters and the accessibility of jammers (e.g. 1-W jammer [86]) make interference elimination difficult, decreasing the reliability of GPS [25].

Vision-based sensors for vehicle navigation are advantageous over GPS due to reliability and light weight, while providing key information required for autonomous guidance and navigation of the vehicle. Vision-based sensors further enable vehicles to detect obstacles, allowing for a larger set of mission objectives and increasing the robustness and flexibility of the unmanned vehicle. For OA, visual information must be processed through an estimation algorithm that estimates the relative range between the vehicle and nearby obstacles of unknown size. It is therefore crucial to develop a robust and efficient estimation scheme before vehicles can perform autonomous OA. Extended Kalman filtering (EKF) has been used widely for this purpose (see Refs. [88, 91, 3, 76, 95, 96, 15, 21, 13], for example, and the literature reviews in those) to extract the unknown parameters. Application of the EKF requires linearization about the desired trajectory and is very sensitive to initial errors [41, 53, 54, 98]. Ref. [18] uses a special method of initializing the Kalman filter which results in faster convergence and less sensitivity to initial errors. Further improvements of EKF performance with application to OA have been reported in Refs. [53, 73, 54] by using unscented Kalman filters and sigma-point Kalman filters. Unscented Kalman filters provide at least second-order accuracy as compared to the first order accuracy of EKF for the same computational cost [53, 73]. Furthermore, computation of the messy Jacobian matrices of a nonlinear process model and a measurement model in application of EKF are no longer required. Ref. [54] uses a sigma-point Kalman filter to estimate unknown obstacle parameters using only inertial measurements and bearings to landmarks, which has been prone to diverge when using extended Kalman filter. The sigma-point Kalman filter is able

to cope with more significant nonlinearities in the system equations and uncertainties in state estimates as compared to EKF. EKF can also be augmented with filters to reduce measurement noise as seen in Refs. [80, 18].

In EKF application for parameter estimation, convergence guarantees for the parameter and range estimates are not derived for the EKF or any of its variations [41]. In [60], convergence characteristics for EKF of a linear system have been proven. However, the nonlinear nature of the measurement equation [12] makes the results of [60] difficult to apply without further extension. In contrast, convergence properties and error bounds for estimation of unknown parameters for nonlinear systems has been derived for two estimators. The first is an existing method known as Identifier-Based Observer [41, 31, 32] (IBO). The IBO generalizes the problem of parameter estimation for constant unknown parameters to consider a more general class of unknown parameters that can change as a nonlinear function of the states. The stability proofs in [41, 31, 32] extend the ideas in [67] to include these general unknown parameters. Upon satisfying certain assumptions on the states of the system and the system dynamics, the IBO framework guarantees exponential convergence of the unknown range and parameter estimates. The second estimator is an adaptive estimator known as fast estimator, which has been shown recently in Refs. [61, 17, 62] to give an estimation of the range between the vehicle and a target with quantifiable performance bounds. The IBO is well-established in the literature and is known to perform well, which serves as a basis for comparison for the newly developed fast estimator. The authors in [61] have compared the performance of the fast estimator with the IBO in simulation for the problem of estimating unknown rotational parameters of a moving object.

1.1.3 Vehicle Coordination

Coordination between vehicles is a well researched area motivated by a diverse range of mission scenarios. Typically, UAV missions require a single autonomous vehicle to be managed by a crew using a ground station provided by the vehicle manufacturer [43]. To execute

more challenging missions and to exploit the availability of powerful embedded systems and communication networks requires the use of multiple vehicles working together to achieve a common objective. Examples of cooperative mission scenarios for UAVs are sequential auto-landing and coordinated ground target suppression. The first refers to the situation where a fleet of UAVs must break up and arrive at the assigned glideslope point, separated by pre-specified safe-guarding time-intervals. For the case of ground target suppression, a formation of UAVs must again break up and execute a coordinated maneuver to arrive at a pre-defined position over the target at the same time. Coordination amongst vehicles to stay within a certain formation is relevant for marine applications, such as fast acoustic coverage of the seabed [74]. In this scenario, multiple vehicles traverse above the ocean floor at various depths along their respective similar spatial paths to map the seabed with the use of acoustic sensors. Multiple vehicles can cover a larger area in a lesser amount of time, reducing operating costs and required time. This type of application has inspired the problem of *coordinated path-following* - vehicles must satisfy a coordination objective while traveling along a pre-specified path [30]. Additional AUV applications include image acquisition (i.e. one vehicle carries a strong light source and illuminates the scenery around while a second underwater vehicle acquire images) and underway replenishment, or replenishment at sea (i.e. fuel, food, parts or personnel are transferred from one vehicle to another while both vehicles are moving underwater) [30]. Further coordination work has been completed for spacecraft and aircraft formation flying [63], [6], coordinated control of land robots [27, 28, 71], control of multiple AUVs [75], [74], and networked robots [14]. In the above applications, it is common to assume that the vehicles are coupled through the common task they are trying to accomplish, but are otherwise dynamically decoupled, meaning the motion of one does not directly affect the others [22]. Control algorithms with this type of decoupling between vehicles is referred to as *decentralized* [89].

For each type of coordination, stability and performance of the entire fleet of vehicles depends on the nature of the underlying communication topology [43, 44]. Therefore, a major focus of current research is to analyze various communication models that model inter-agent

communication networks. Typically, practical limitations are present that impede the flow of information between vehicles. For example, underwater communications (that rely on the propagation of acoustic waves) are susceptible to intermittent failures, delays, and multi-path effects. As a result, tight limits are set on the effective communication bandwidths that can be achieved [30]. In general, no vehicle will be able to communicate with the entire formation and the vehicles cannot all communicate at the same time. Communication links between vehicles are classified in two ways: bi-directional, where a communication link between two vehicles implies two-way communications, and uni-directional, where communication links are associated with a direction with one vehicle receiving and the other vehicle sending information. Furthermore, communication networks can be fixed or can change with time. In [22], the authors consider fixed bi-directional communication between vehicles to solve a formation control objective. In [59], the authors consider a more complicated case of fixed, uni-directional topology and solve the *rendezvous problem* - coordinating all vehicles to arrive at their final destinations simultaneously - for a fleet of mobile robots whose field of view has a fixed radius. The more complicated case of time-varying network topologies have been addressed in [40], [65], [59], in a deterministic setting, and in [63], [89], [90] using stochastic models. The authors in [40] address a particular discrete-time model of M agents with the same speed and different headings. Each agent's heading is updated using a local rule based on the average of its own heading plus the headings of its neighbors. The paper provides theoretical results for this model (previously only shown in simulation) and convergence properties for similar types of models with time-varying communication topologies. In [65], each vehicle updates its current state based upon the current information received from neighboring vehicles. Necessary and/or sufficient conditions for the convergence of the individual vehicle's states to a common value are presented in discrete time. In [59], the communication topology is state independent and generalizes results from [65] in discrete time to continuous time using vector field assumptions. Using stochastic models, state-dependent graphs are used to model the distance between agents or the signal strength in each link in the formation in [63]. The authors in [89] present a decentralized control design methodology

for regulating global functions (e.g. center and distribution about the center of a platoon of AUVs). A network of oscillators is considered in [90] and it is shown that synchronization can be achieved with sufficiently fast switching.

Finally, decentralized control for a fleet of vehicles can have a “leader-follower” approach, which has the advantage of simplicity in that a reference trajectory is clearly defined by the leader. This increases flexibility in the mission, as the leader can change its reference trajectory to adjust the entire fleet [22]. However, leader-follower architecture is almost entirely dependent on the leader, and over-reliance on a single vehicle in the formation is undesirable in adversarial environments as damage or loss of the leader may lead to mission failure. The second approach is the virtual leader approach [22], [30], in which vehicles in the formation utilize information from a fictitious leader vehicle whose trajectory acts as a reference trajectory for the group. This approach decreases vulnerability of the mission to any single vehicle at the expense of either higher communication to communicate the virtual leader’s information, or mission flexibility by having all vehicles know a reference signal beforehand which cannot be changed throughout the mission.

1.2 Objectives

The objective of this dissertation is to solve the complete path-following rendezvous problem of multiple mobile robots with obstacle avoidance using vision-based sensors. This includes path generation at a kinematic level that incorporates constraints on vehicle speed and acceleration. Furthermore, the path generation must address simultaneous arrival of all vehicles at their desired (not necessarily the same) final destinations in the absence of delays. This simultaneous rendezvous problem is referred to throughout as the *time-critical coordination* problem. At the path generation level, obstacle avoidance can be considered at a basic level. Inaccurate or unknown obstacle information implies that obstacle avoidance at this point may not be reliable. An obstacle avoidance scheme must be developed during the mission as

more accurate real-time information is obtained from onboard visual sensors. Along each vehicle's respective path, at most one obstacle per vehicle can be encountered, which the vehicle must avoid. During avoidance, each vehicle must not collide with its neighboring vehicles. Visual information must be processed through an estimator that outputs obstacle size and location. These quantities act as inputs to a real-time obstacle avoidance scheme. Lastly, vehicles must meet the time-critical coordination objective in the presence of bi-directional and uni-directional communication topologies. This is achieved using a decentralized scheme in a leaderless setting. In order to keep the ideas present as general as possible, 2D mobile robot systems are considered which can be extended to UAV, AUV, and ground platforms.

1.3 Contributions

- The path generation method in [29] is extended to include OA by introducing penalties in a cost function. An analysis of the virtual speed functions and its effect on the path generation is included to provide a guideline in the design process. Specifically, the virtual speed is chosen as a polynomial function of a dummy variable, s . An analysis is provided for various order polynomials to provide a relationship between the choice of the virtual speed and its effect on the path generation method. The method is compared to paths generated by Bezier curves as an example of other types of polynomial path generation. It is shown in Corollary 1 that for a special case, the paths generated by Bezier curves are identical to the paths generated by the polynomial path generation method. This further conveys the generality of the method and shows that it performs well as compared to other path approximations.
- The path deformation method for single vehicle obstacle avoidance in [52] is augmented with the projection operator in order to keep input signals bounded. The projection operator is applied in a novel way to prevent control saturation as an alternative to conventional methods, which has not been previously considered. Furthermore, the

path deformation algorithm viewed as a dynamical system is shown to be stable and that the potential function decreases in the presence of this augmentation. In addition, the algorithm takes into account the multi vehicle set-up in avoiding obstacles to generate collision-free deformed paths. This is formally done by relating the spatial separation between vehicles imposed at the path generation level to the obstacle avoidance scheme. In contrast to [52], polynomial basis functions are used to define the deformed path instead of Fourier series to remain consistent with the polynomial path generation method.

- Two estimation schemes are compared to estimate the unknown obstacle size and position. The effect of the estimation errors on the OA component is explicitly shown in Propositions 9 and 10 and the relationship between the estimator parameters and the estimator performance is discussed in Remarks 16 and 17. The comparison between estimators provides a theoretical framework for determining which estimation scheme is appropriate and how the estimator parameters are related to the obstacle avoidance parameters. The estimators are shown to perform comparably in a variety of simulation environments which illustrate the theoretical results.
- A coordination control law for the velocity command of each vehicle is presented for both a connected bi-directional and uni-directional communication topology. The first is a leader-follower algorithm where only the leader knows a desired speed profile beforehand. For the second case of a uni-directional communication environment, a leaderless control law is introduced. The trade off for this method is that all vehicles must know the reference speed. Both controllers are Proportional-Integral (PI) and it can be proved that the resulting coordination error goes to zero exponentially. A previously-reported Proportional-Derivative (PD) law is reviewed for comparison and the PI controller presented in this dissertation is shown to be advantageous by eliminating steady-state error.

1.4 Organization

The dissertation is organized as follows:

Chapter 2: A real-time path generation method for collision-free maneuvers of multiple vehicles based on polynomial curve fitting is presented. The method is designed to incorporate the coordination objective. Furthermore, position and velocity are decoupled by defining the path with respect to a dummy variable, s , whose relationship with time differs from vehicle to vehicle. Each vehicle's velocity profile can be adjusted to achieve the coordination task in Chapter 5. The problem of tracking a path with a given speed profile (i.e. trajectory tracking) becomes the simplified problem of path following. A baseline path following controller is implemented which ensures that the vehicle tracks the generated path.

Chapter 3: An online OA algorithm for each vehicle is introduced that avoids isolated obstacles by iteratively deforming the original path using information about an obstacle's size and location. The OA algorithm utilizes a potential function that yields larger values for paths that are close to obstacles and lesser values for paths that are far from obstacles.

Chapter 4: Visual information collected by each vehicle's onboard camera is processed through an estimator that provides estimations of obstacle sizes and ranges within each vehicle's sensing range. This information provides inputs to the OA algorithm in Chapter 3. The IBO and fast estimator and the effect of both estimators' performance on the OA scheme is presented in detail. This dissertation does not address the case when visual measurements are intermittent or unavailable (e.g. occlusion of obstacles, poor lighting, etc.). Future work can be directed towards extending this work to include missing estimates. Simulations are presented which illustrate the theoretical findings.

Chapter 5: Multiple vehicle coordination for the rendezvous problem is achieved by adjusting the speed profiles of each vehicle along their paths in response to informa-

tion exchanged over a communication network. First, bi-directional communication between vehicles is considered within a leader-follower architecture. Then a virtual leader approach is considered for uni-directional communication between vehicles.

Chapter 6: The dissertation is summarized and concluding remarks are made in regards to future work.

Chapter 2

Path Following in 2D Space

In this Chapter, the concept of paths is reviewed. A polynomial path generation method which takes into consideration vehicle limits on velocity and acceleration is given. The method makes use of a single polynomial function for each vehicle to generate paths that satisfy initial and final boundary conditions. The method uses a direct method of calculus of variations that takes explicitly into account simplified vehicle dynamics, a general performance criterion to be optimized, and safety rules for collision avoidance. The generated paths guarantee that all vehicles arrive at their destination simultaneously in the absence of delays and disturbances. This step yields - for each vehicle - a spatial path to be followed. A path following controller is reviewed which guarantees that each vehicle follows the desired path while tracking a desired speed profile. The desired speed profile is determined by the coordination objective which requires that all vehicles arrive at their final destinations simultaneously and is given in Chapter 5.

2.1 Review of Paths

Consider a nonholonomic system of dimension n with kinematics

$$\dot{x}(t) = F(x(t)) + G(x(t))u(t), \quad (2.1)$$

where $x(t) \in \mathbb{R}^n$ is the state vector of the system, $u(t) = [u_1(t) \ u_2(t) \ \dots \ u_m(t)]^\top \in \mathbb{R}^m$ is the vector of kinematic control inputs, $F(x(t)) \in \mathbb{R}^{n \times 1}$ and $G(x(t)) \in \mathbb{R}^{n \times m}$ depend upon the state of the system, and $m < n$. For physical systems, such as the mobile robot, $x(t)$ can represent a vector of the position coordinates and orientation of the vehicle, and $u(t)$ can represent the vehicle's speed and rate of rotation. The system in (2.1) can be re-written as

$$\dot{x}(t) = F(x(t)) + G_1(x(t))u_1(t) + G_2(x(t))u_2(t) + \dots + G_m(x(t))u_m(t), \quad (2.2)$$

where $G_1(x(t)), G_2(x(t)), \dots, G_m(x(t))$ are the *control vector fields* and $G_j(x(t))$ corresponds to the j th column of $G(x(t))$ in (2.1). The form in (2.2) will be useful in later analysis. The kinematic control inputs $u_1(t), u_2(t), \dots, u_m(t)$ are the elements of $\dot{x}(t)$ with respect to the vectors $(G_1(x(t)), \dots, G_m(x(t)))$ [56]. The vector $F(x(t))$ is denoted as the *drift* of the system and for kinematic systems, is equal to zero. For example, consider the simple kinematic system

$$\dot{x}(t) = u(t), \quad (2.3)$$

where $u(t)$ is the velocity which is taken as the kinematic control input. It follows that

$$\dot{x}(t) = G_1(x(t))u(t), \quad (2.4)$$

where $G_1(x(t)) = 1$ and the drift is zero. If acceleration is taken as the control input (i.e. a second order system), then

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ u(t) \end{bmatrix} \quad (2.5)$$

implies that

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \underbrace{\begin{bmatrix} x_2(t) \\ 0 \end{bmatrix}}_{F(x(t))} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{G_1(x(t))} u(t). \quad (2.6)$$

In the 2D space, a path $p(s)$ is a curve $p : \Omega \rightarrow \mathbb{R}^2$ parameterized by s , which lies within a closed subset Ω of \mathbb{R}^+ . In general, the subset Ω may be restricted depending on the path (e.g. Bezier curves are restricted so that $s \in \Omega = [0, 1]$ [77]). If s is identified with, or is a function of time, $p(s) = p(s(t)) = p(t)$ is referred to as a trajectory [42, 43, 44].

An *admissible path* with respect to the nonholonomic system in (2.2) is a smooth curve in \mathbb{R}^n defined over $\Omega \in \mathbb{R}^+$ such that there exist m mappings $u_1(s), u_2(s), \dots, u_m(s)$ defined over Ω , with [56]

$$\frac{d}{ds}(p(s)) = \sum_{j=1}^m u_j(s) G_j(p(s)), \quad \forall s \in \Omega, \quad (2.7)$$

where $u_j(s)$ and $G_j(p(s))$ are the j th component and j th column of u and G introduced in (2.1), respectively. The mappings $u_1(s), \dots, u_m(s)$ are referred to as the *kinematic inputs*. The kinematic inputs are the coordinates of $\frac{d}{ds}(p(s))$, which can be viewed as the velocity vector at s along the current path $p(s)$ in basis $(G_1(p(s)), \dots, G_m(p(s)))$. The kinematic inputs must lie within compact sets that respect the vehicle's physical limitations to ensure that the path is feasible for the vehicle to follow. For instance, the vehicle may be susceptible to control saturation and is limited in velocity, acceleration, and turn rate. The concept of perturbing the kinematic inputs is the key to the path deforming OA algorithm in Chapter 3.

2.2 Path Generation

This section describes an algorithm for 2D path generation with OA of known obstacles. The algorithm ensures that the simultaneous arrival objective of all vehicles is guaranteed.

2.2.1 Direct Methods

To illustrate the path generation problem for the rendezvous mission, consider a fleet of mobile robots that are tasked to start from different locations and arrive at their respective final destination simultaneously. The exact time of arrival is not specified, but it is restricted to lie within certain bounds, t_1, t_2 . The vehicles must complete the mission while avoiding collisions between vehicles, respecting dynamical constraints (e.g. acceleration bounds), and minimizing a weighted combination of “costs” (e.g. fuel consumption and obstacle avoidance). A straightforward solution to this problem would be to let $s = t$ and solve a constrained optimization problem off-line that would generate feasible paths $p_i(t), t \in [0, t_f]$, where $i = 1, 2, \dots, n$ denotes the number of vehicles and t_f is the final time [10]. Onboard systems would guarantee that the vehicles track the desired trajectories generated, thus meeting the mission objectives. This solution has one severe drawback: it depends on the absolute timing by enforcing a fixed speed profile for each vehicle and does not allow for any on-line modifications due to “deviations from the plan.” In the event that one or more of the vehicles is delayed (e.g. due to unforeseen obstacles, for example) or cannot accurately follow the desired trajectory due to adverse terrain conditions or inadequate propulsion, the proposed method will fail. It is vital to develop a path generation algorithm that can be implemented on-line so that the changes can be made as necessary. To achieve this, a sufficiently fast and easily implementable solution must be used [97].

The main idea of direct methods is to simplify the path generation problem by considering each component of the path as a finite set of variables [97]. Direct methods make simplifications and may discretize a given problem so that the subsequent problem is more easily solved using nonlinear programming techniques [58]. Solutions obtained with these direct methods are generally considered sub-optimal due to these simplifications and discretization [81]. In contrast, indirect methods use calculus of variations techniques to recast the problem as a two-point boundary value problem. Indirect methods are generally more difficult to formulate and obtain a solution, whereas with direct methods, mathematical suboptimal

solutions are obtained. In this research, direct methods are useful in order to more easily obtain solutions to the path generation problem. Considering each component of the path as a finite set of variables, the problem can then be reduced by considering a finite series instead of infinite [19]. An admissible function can be expressed by an infinite power series

$$f(x) = \sum_{k=0}^{\infty} a_k x^k, \quad (2.8)$$

a Fourier series

$$f(x) = \frac{b_0}{2} + \sum_{k=1}^{\infty} (b_k \cos kx + c_k \sin kx), \quad (2.9)$$

or by any series

$$f(x) = \sum_{k=0}^{\infty} d_k \varphi_k(x), \quad (2.10)$$

where a_k 's, b_k 's, c_k 's, and d_k 's are constants and $\varphi_k(x)$ are known basis functions. Euler has been credited as being the first to apply the direct method of finite differences [20]. Further extension of Euler's work with application of direct methods in the theory of elasticity can be found in [84, 51, 24, 45]. In [92], the author first applied the idea of direct methods to generate flight trajectories by using polynomial functions parameterized by a variable, s , to define a flight vehicle's Cartesian coordinates and velocity in 3D. The parameter s can be equal to, for example, time, path length, or any other measure, depending on the vehicle's task. Assigning a relationship between time and s explicitly defines an associated speed profile along with the path. For instance, the choice of $s = t$ assigns a speed profile given by

$$v(t) = \sqrt{\dot{x}_1^2(t) + \dot{x}_2^2(t) + \dot{x}_3^2(t)}. \quad (2.11)$$

In order to vary speed and spatial constraints separately, the author in [92] keeps s as an abstract parameter for additional flexibility and refers to it as the *virtual arc*. By defining s separately for each vehicle, multiple vehicles are allowed to follow the same path with different speed histories, exploiting a separation between spatial and temporal constraints. The choice of using polynomials allows more flexibility as the shape of any path generated

by polynomial functions can be changed by simply varying $s \in \Omega = [s_0, s_f]$, where $s_f > s_0$ are the initial and final virtual arc lengths. Polynomials are easy to compute, affording high computational efficiency. In [92], the path for an aircraft is given by

$$x_i = x_{i_0} + \frac{(x_{i_f} - x_{i_0})(s - s_0)}{s_f - s_0} + \Phi_i(s), \quad (2.12)$$

where the coordinates x, y, z are represented as x_1, x_2, x_3 for notational simplicity, $x_4 = V$, the true airspeed, and x_{i_0}, x_{i_f} represent the initial and final values of x_i , respectively. The continuously differentiable functions $\Phi_i(s)$ are chosen as a linear combination of the following functions:

$$\begin{aligned} \Phi_i^1(s) &= \sum_{k=1}^N h_k \sin k\pi \frac{s - s_0}{s_f - s_0} \\ \Phi_i^2(s) &= \sum_{k=1}^N g_k (s - s_0)^k (s - s_f)^k, \\ \Phi_i^3(s) &= (s - s_0)^{l_1} (s - s_f)^{l_2}, \end{aligned} \quad (2.13)$$

where N, l_1, l_2, h_k 's and g_k 's are optimization parameters. Increasing the numbers N, l_1 , and l_2 affords more flexibility to the optimization problem by introducing a higher number of optimization parameters. The higher the order of the polynomials, the closer a near-optimal solution is to the optimal one [19] and the harder the optimization task in terms of computational efficiency and complexity. In [93], this problem is addressed by choosing x_i 's as the sum of three cubic polynomials. The number of polynomial coefficients is decreased by using lower order polynomials, reducing the number of optimization parameters. Even with a reduced number of optimization parameters, the method in [93] implies relatively difficult numerical calculations that make on-line implementation difficult [97]. With this method, many trajectory optimization problems for flight vehicles have been solved for a variety of applications [8, 2, 69] off-line. Onboard, on-line optimization using direct methods requires further reduction in optimization parameters and simplifications in order to converge quickly and efficiently. Furthermore, the method in [93] does not always yield analytical solutions (i.e. paths that can be continuously parameterized by a parameter), which is desirable for

tracking since pilots are trained to follow a given set of basic trajectories [97]. Numerical solutions require the pilot to interpolate between points, which is difficult in active flight. While the goal of this dissertation is not necessarily intended for piloted flight and navigation, the path generation method is chosen to yield analytical paths so as to provide a more general framework for a wider variety of applications.

An online path generation algorithm which yields analytical expressions for the path of a single aircraft is introduced in [97] for trajectory generation of aircraft. The main idea of [97] is to represent each coordinate of the desired path by a single polynomial function whose order is determined by the sum of the highest derivative of the boundary conditions that must be satisfied at the initial and final points. The coefficients of the polynomials are computed via an algebraic relationship to ensure that the path satisfies the boundary conditions by construction. The only optimization parameter is the final arc length s_f . Additional optimization parameters can be added by introducing fictitious boundary conditions for higher-order derivatives. The method in [97] generates near-optimal paths for single vehicles as compared to solutions obtained by applying Pontryagin's maximum principle. These results have been expanded upon and applied to the unmanned aerial vehicle (UAV) and autonomous underwater vehicle (AUV) in [42, 43, 29], respectively. The authors in [42, 43, 29] used this methodology to generate paths for multiple vehicles tasked to arrive at a pre-specified destination simultaneously. This is known in the literature as the *rendezvous problem*. To extend the results for multiple vehicles, [42, 43, 44, 29] include inter-vehicle collision avoidance and path generation with the objective of simultaneous arrival. In the following, the method of [97, 42, 43, 44, 29] is reviewed for the 2D case, which can be used for some UAV or AUV applications.

2.2.2 Single Polynomial Path Generation in 2D

Let $p(s) = [x(s) \ y(s)]$ be a desired path to be followed by a single vehicle, $s_0 = 0$, and $s \in \Omega = [0, s_f]$, where s_f has yet to be defined. For notational simplicity, let $x_1(s) = x(s)$

and $x_2(s) = y(s)$. Following [42, 43, 44], these components are represented by algebraic polynomials of degree N of the form

$$x_i(s) = \sum_{k=0}^N a_{ik} s^k, \quad s \in [0, s_f], \quad i = 1, 2, \quad (2.14)$$

where the degree N is determined by the number of boundary conditions that must be satisfied. Note that in contrast to [93], $x_4 = V$ is not considered and the coefficients a_{ik} will be determined algebraically, rather than being considered as optimization parameters. It is also useful to note that [97] uses a slightly different form of (2.14) given by

$$x_i(s) = \sum_{k=0}^N a_{ik} \frac{(\max(1, k-2)!) s^k}{k!}, \quad i = 1, 2. \quad (2.15)$$

Comparison of (2.14) to (2.15) shows that the methodologies are equivalent and differ only in the values of the coefficients a_{ik} by some scaling factors. The expression in (2.14) is a simplification of (2.15) by eliminating the extra factorial and multiplicative terms. In the sequel, the form in (2.14) is used for simplicity.

Let d_0 and d_f be the highest-orders of the spatial-derivatives of $x_1(s)$ and $x_2(s)$ that must meet pre-specified boundary conditions at the initial and final points of the path. The minimum degree of each polynomial in (2.14) is $N^* = d_0 + d_f + 1$. The higher the maximum degree of time derivative of an aircraft coordinate at its initial and terminal points, the higher the degree of the polynomial. Choosing N to be larger than N^* introduces additional degrees of freedom and enlarges the set of paths that meet the prescribed boundary conditions. For notational purposes, primes $'$ denote derivation with respect to s and overdot $\dot{}$ denotes time derivatives. As an illustrative example, if a desired path must include constraints on initial and final positions, velocities, and accelerations (second-order derivatives), then the

minimum degree of each polynomial is $N^* = 2 + 2 + 1 = 5$. From (2.14), one obtains:

$$\begin{aligned} x_i(s) &= \sum_{k=0}^5 a_{ik} s^k, \\ x'_i(s) &= \sum_{k=1}^5 k a_{ik} s^{k-1}, \\ x''_i(s) &= \sum_{k=2}^5 k(k-1) a_{ik} s^{k-2}, \quad i = 1, 2. \end{aligned} \tag{2.16}$$

Substituting for $s = 0$ and $s = s_f$, it follows that

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & s_f & s_f^2 & s_f^3 & s_f^4 & s_f^5 \\ 0 & 1 & 2s_f & 3s_f^2 & 4s_f^3 & 5s_f^4 \\ 0 & 0 & 2 & 6s_f & 12s_f^2 & 20s_f^3 \end{bmatrix} \begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \end{bmatrix} = \begin{bmatrix} x_i(0) \\ x'_i(0) \\ x''_i(0) \\ x_i(s_f) \\ x'_i(s_f) \\ x''_i(s_f) \end{bmatrix}. \tag{2.17}$$

This in turn yields twelve linear algebraic equations that can be used to solve for the twelve unknown coefficients a_{ik} . One can also add additional boundary conditions on higher-order derivatives in order to introduce additional design parameters. For example, adding a fictitious constraint on initial jerk (third-order derivative) $x'''_i(0)$ increases N from $N = 5$ to $N = 3 + 2 + 1 = 6$. The number of coefficients a_{ik} is increased to fourteen ($i = 1, 2, k = 0, 1, \dots, 6$) and the fictitious boundary value $x'''_i(0)$ is used as an additional design parameter. Figure 2.1 shows how adding an extra boundary condition for initial jerk (third-order derivative) allows for a larger set of admissible paths. Figure 2.2 shows the corresponding speed and acceleration profiles of the admissible paths.

It is necessary to relate the parameter s to time in order to define a desired speed profile and obtain the boundary conditions $x'_i(0), x'_i(s_f), x''_i(0), x''_i(s_f), \dots$ with respect to s . This is done by defining the *virtual speed* $\eta(s) = ds/dt$. The choice of $\eta(s)$ relates s with time, recasting $p(s)$ as a trajectory. The boundary conditions can subsequently be computed with respect

Vehicle Trajectories: 5th Order (Solid) and 6th Order (Dashed) Polynomials

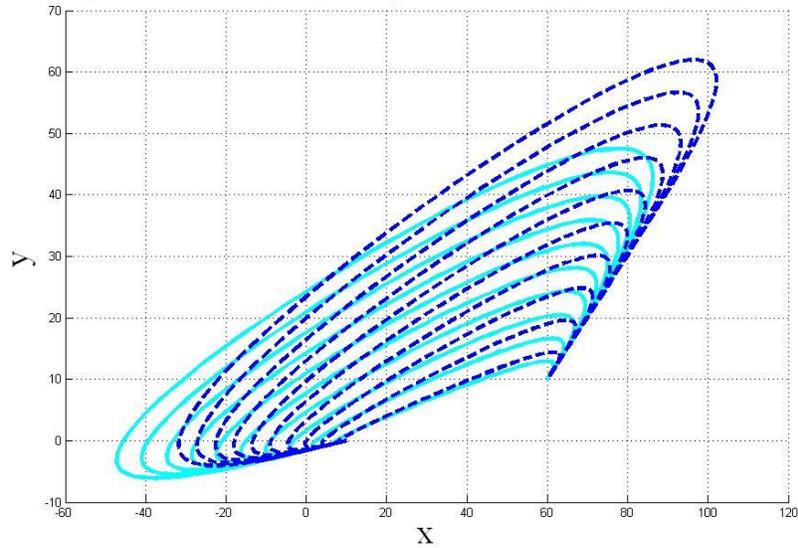
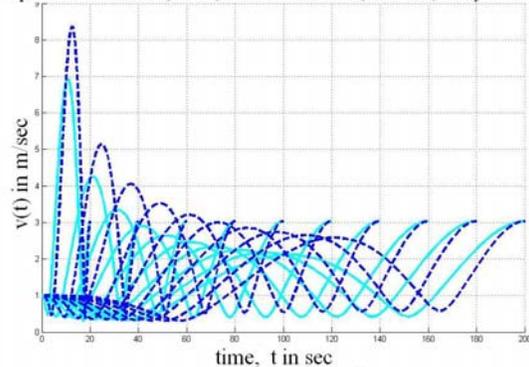


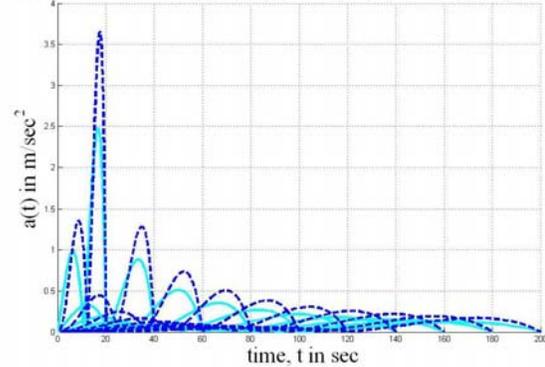
Figure 2.1: Trajectories corresponding to 5th and 6th order polynomials.

Speed: 5th Order (Solid) and 6th Order (Dashed) Polynomials



(a) Speed profiles

Acceleration: 5th Order (Solid) and 6th Order (Dashed) Polynomials



(b) Acceleration profiles

Figure 2.2: Speed and acceleration profiles for 5th and 6th order polynomials.

to s . As stated before, the choice of $s = t$ implies that defining spatial profiles imposes the fixed speed profiles along a given path given by (2.11). For multiple vehicles, different choices of $\eta(s)$ for each vehicle results in different speed profiles along the same path.

Note that the parameterization in (2.14) along with the definition of the coefficients a_{ik}

implies that the resulting 2D path satisfies all boundary constraints by construction. Table 2.1 shows how to compute the polynomial coefficients a_{ik} in (2.14) for fifth and sixth order polynomial paths. The sixth order polynomial path includes a fictitious extra constraint on the initial jerk (third-order derivative), where $x_1'''(0), x_2'''(0)$ are added as extra design parameters. Note that the boundary conditions $x_i(0), x_i'(0), x_i''(0), x_i'''(0), x_i(s_f), x_i'(s_f), x_i''(s_f)$ are expressed with respect to s , whose relationship with time has yet to be defined. The derivation of the s -derivatives, their relationship to time, and how temporal constraints are included in the path generation process are discussed next.

In [97], the author considers the three-dimensional point-mass aircraft model over a flat Earth with zero sideslip angle and calculates a numerical solution for $\eta(s)$ at each time step as $\eta_j = \Delta s \Delta t_j^{-1}$, where j is the iteration number, Δs is a constant s -step, and Δt_j is the computed time-step that depends on Δs , the true airspeed V_j , and the position of the aircraft. In [42], $\eta(s)$ was chosen to be a general polynomial function whose coefficients along with s_f were chosen as optimization parameters.

The case of path generation for multiple vehicles is considered in [43], where the authors exploited the fact that small UAVs operate at essentially constant speeds and set $\eta(s) = v_p / \|\dot{p}(s)\|$, where $v_p \in [v_{\min}, v_{\max}]$ is the constant desired speed. Choosing $\eta(s)$ in this way eliminates the coefficients of $\eta(s)$ in the optimization process and increases computational efficiency without sacrificing an analytical solution for the path. For this case, s_f becomes the only optimization parameter for each vehicle. Hence, the number of optimization parameters increases linearly with the number of vehicles. Furthermore, by utilizing the speed bounds v_{\min} and v_{\max} , [43] was able to compute the minimum and maximum times of arrival of each vehicle, defining an arrival window for each vehicle. The method in [43] also optimized the *arrival margin*, which is defined as the intersection of all vehicle arrival windows. Including this additional constraint affords additional flexibility in the mission due to unforeseen events, such as disturbances. The maximum arrival margin is used to coordinate simultaneous arrival by dictating the fleet leader's speed profile [43].

With respect to mobile robots or AUVs, it may not be reasonable to assume that the vehicle operates at constant speeds. The authors in [29] define $\eta(s)$ as a linear function of s , whose boundary conditions are given by:

$$\eta(0) = \|\dot{p}(0)\|, \quad \eta(s_f) = \|\dot{p}(t_f)\|. \quad (2.18)$$

The simple choice of

$$\eta(s) = \eta_0 + \frac{\eta_f - \eta_0}{s_f} s, \quad (2.19)$$

where $\eta_0 = \eta(0)$, $\eta_f = \eta(s_f)$ implies that simultaneous time of arrival is guaranteed. Indeed, since by definition,

$$t = \int_0^s \frac{1}{\eta(\tau)} d\tau, \quad (2.20)$$

where t denotes time, it follows that

$$\begin{aligned} t_f &= \int_0^{s_f} \frac{1}{\eta(\tau)} d\tau \\ &= \int_0^{s_f} \frac{s_f}{\eta_0 s_f + (\eta_f - \eta_0)\tau} d\tau \\ &= \begin{cases} \frac{1}{\eta_0} s_f & \text{if } \eta_f = \eta_0, \\ \frac{s_f \ln(\eta_f/\eta_0)}{\eta_f - \eta_0} & \text{otherwise.} \end{cases} \end{aligned}$$

Solving for s_f gives

$$s_f = \begin{cases} \eta_0 t_f, & \text{if } \eta_f = \eta_0, \\ \frac{\eta_f - \eta_0}{\ln(\eta_f/\eta_0)} t_f, & \text{otherwise.} \end{cases} \quad (2.21)$$

Similarly, one can compute using (2.20) and (2.21) that

$$s = \begin{cases} \eta_0 t & \text{if } \eta_f = \eta_0, \\ \frac{\eta_0 t_f}{\ln(\eta_f/\eta_0)} \left(\left(\frac{\eta_f}{\eta_0} \right)^{\frac{t}{t_f}} - 1 \right) & \text{otherwise.} \end{cases} \quad (2.22)$$

Table 2.1: Examples of computation of the coefficients of 5th and 6th order polynomial paths

5 th order		
Boundary conditions	$x_i(0), x'_i(0), x''_i(0), x_i(s_f), x'_i(s_f), x''_i(s_f)$	
d_0/d_f	2/2	
N^*/N	5/5	
Linear algebraic matrix equation to solve for the coefficients a_{ik}	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 1 & s_f & s_f^2 & s_f^3 & s_f^4 & s_f^5 \\ 0 & 1 & 2s_f & 3s_f^2 & 4s_f^3 & 5s_f^4 \\ 0 & 0 & 2 & 6s_f & 12s_f^2 & 20s_f^3 \end{bmatrix}$	$\begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \end{bmatrix} = \begin{bmatrix} x_i(0) \\ x'_i(0) \\ x''_i(0) \\ x_i(s_f) \\ x'_i(s_f) \\ x''_i(s_f) \end{bmatrix}$
6 th order		
Boundary conditions	$x_i(0), x'_i(0), x''_i(0), x'''_i(0), x_i(s_f), x'_i(s_f), x''_i(s_f)$	
d_0/d_f	3/2	
N^*/N	5/6	
Linear algebraic matrix equation to solve for the coefficients a_{ik}	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 1 & s_f & s_f^2 & s_f^3 & s_f^4 & s_f^5 & s_f^6 \\ 0 & 1 & 2s_f & 3s_f^2 & 4s_f^3 & 5s_f^4 & 6s_f^5 \\ 0 & 0 & 2 & 6s_f & 12s_f^2 & 20s_f^3 & 30s_f^4 \end{bmatrix}$	$\begin{bmatrix} a_{i0} \\ a_{i1} \\ a_{i2} \\ a_{i3} \\ a_{i4} \\ a_{i5} \\ a_{i6} \end{bmatrix} = \begin{bmatrix} x_i(0) \\ x'_i(0) \\ x''_i(0) \\ x'''_i(0) \\ x_i(s_f) \\ x'_i(s_f) \\ x''_i(s_f) \end{bmatrix}$

Finally, one can compute

$$\frac{s}{s_f} = \begin{cases} \frac{t}{t_f}, & \text{if } \eta_f = \eta_0, \\ \frac{\eta_0}{\eta_f - \eta_0} \left(\left(\frac{\eta_f}{\eta_0} \right)^{\frac{t}{t_f}} - 1 \right), & \text{otherwise,} \end{cases} \quad (2.23)$$

and verify that $s = s_f$ when $t = t_f$. Satisfying this condition for every vehicle guarantees simultaneous time of arrival. In this setting, the number of optimization parameters is decreased to a single variable, t_f for any number of vehicles. If t_f is defined, the expression in (2.21) yields the final arc lengths s_j for $j = 1, \dots, n$. Vehicle j 's trajectory $p_j(s_j)$ is computed from (2.14) for $s_j \in [0, s_{f_j}]$.

Remark 1 *The method presented in [29] is in contrast to [42, 43, 44], where s_{f_j} 's are the optimization variables. Instead of n optimization parameters, the only optimization parameter is $t_f \in [t_1, t_2]$. Furthermore, the choice of $\eta_j(s)$ in (2.19) incorporates the rendezvous problem into the path generation, simplifying the coordination task. Further robustness in the coordination task can be introduced by optimizing the arrival margin as in [43].*

In this dissertation, the method in [29] is utilized and extended to include obstacle avoidance. At the path planning stage, it is assumed that the vehicle has preliminary information about obstacle locations. This information is incorporated into the path planning by including penalty terms in the cost function that must be minimized in the optimization. This is formally done by utilizing a cost function that obtains larger values closer to obstacles than farther away. More sophisticated methods for OA can be applied at the planning stage at the cost of additional computational expense and increased complexity. However, since initial information about obstacle locations may be inaccurate, it may not be cost effective to implement these methods at the path planning stage. Penalizing the cost function is a computationally inexpensive way to include OA in the path planning. The resulting trajectory must be updated as more accurate information is obtained from the visual sensors

onboard the vehicle throughout the mission. Chapter 3 deals with online OA by deforming the initial trajectory.

With the definition of $\eta_j(s_j)$ in (2.19), one can compute that the temporal and spatial derivatives of p_j satisfy

$$\begin{aligned}\dot{p}_j(t) &= \eta_j(s_j)p'_j(s_j), \\ \ddot{p}_j(t) &= \eta_j(s_j)\eta'_j(s_j)p'_j(s_j) + \eta_j^2(s_j)p''_j(s_j).\end{aligned}\tag{2.24}$$

The boundary conditions for $\eta_{0_j} \neq 0, \eta_{f_j} \neq 0$ are computed as:

$$\begin{aligned}p'_j(0) &= \frac{\dot{p}_j(0)}{\eta_{0_j}}, & p'_j(s_{f_j}) &= \frac{\dot{p}_j(t_f)}{\eta_{f_j}}, \\ p''_j(0) &= \frac{\ddot{p}_j(0) - \eta_{0_j}\eta'_{0_j}p'_j(0)}{\eta_{0_j}^2}, & p''_j(s_{f_j}) &= \frac{\ddot{p}_j(t_f) - \eta_{f_j}\eta'_{f_j}p'_j(s_{f_j})}{\eta_{f_j}^2},\end{aligned}\tag{2.25}$$

where

$$\eta'_j(s_j) = \eta'_j = \frac{\eta_{f_j} - \eta_{0_j}}{s_{f_j}}.\tag{2.26}$$

If $\eta_{0_j} = 0, p'_j(0) = p''_j(0) = 0$ and $\eta_{f_j} = 0$, then this implies that $p'_j(s_{f_j}) = p''_j(s_{f_j}) = 0$.

Define a *feasible trajectory* as one that can be tracked by a vehicle without having it exceed pre-specified bounds on the vehicles's speed $v_j(t)$ along the corresponding path as well as on the total acceleration. Let v_{\min}, v_{\max} and a_{\max} denote pre-specified bounds on the vehicle's speed $\|\dot{p}_j(t)\|$ and total acceleration $\|\ddot{p}_j(t)\|$, respectively. For a given $\eta_j(s_j)$, the temporal speed and acceleration along the path $p_j(s_j)$ are computed as:

$$v_j(s_j) = \eta_j(s_j)\sqrt{x_{1_j}^{\prime 2}(s_j) + x_{2_j}^{\prime 2}(s_j)} = \eta_j(s_j)\|p'_j(s_j)\|,\tag{2.27a}$$

$$a_j(s_j) = \|p''_j(s_j)\eta_j^2(s_j) + p'_j(s_j)\eta'_j\eta_j(s_j)\|.\tag{2.27b}$$

A feasible path must satisfy

$$v_{\min} \leq v_j(s_j) \leq v_{\max}, \quad a_j(s_j) \leq a_{\max}\tag{2.28}$$

for all $s_j \in [0, s_{f_j}]$, $j = 1, \dots, n$.

The optimization problem that solves path generation for a fleet of n vehicles tasked for simultaneous arrival is stated formally as that of computing

$$t_f^{\text{opt}} = \min_{t_f \in [t_1, t_2]} \sum_{j=1}^n w_j J_j, \quad (2.29)$$

subject to spatial deconfliction

$$\|p_j(t) - p_k(t)\| > \Xi, \quad \forall t \in [0, t_f^{\text{opt}}], \quad (2.30)$$

where $j, k \in \{1, \dots, n\}$ with $j \neq k$, and speed and acceleration constraints

$$v_{\min} \leq \eta_j(s_j) \|p'_j(s_j)\|, \quad \|p''_j(s_j) \eta_j^2(s_j) + p'_j(s_j) \eta'_j \eta_j(s_j)\| \leq a_{\max}, \quad (2.31)$$

where w_j are positive weights for $j = 1, 2, \dots, n$, t_1, t_2 are upper and lower bounds on t_f , and J_j is an associated cost function. The spatial deconfliction parameter Ξ is a conservative bound on the desired separation between vehicles. The value of Ξ will be determined in the obstacle avoidance of Chapter 3 to ensure that vehicles do not collide while avoiding an obstacle. The criterion J_j is minimized in the optimization and may include total energy consumption along a trajectory and OA, for example. For flight vehicles, instantaneous power required for vehicle maneuvering is proportional to the cube of speed, and the energy consumption cost function is

$$J_{e_j} = \int_0^{t_f} \|\dot{p}_j(t)\|^3 dt = \int_0^{s_{f_j}} \|p'_j(s_j)\|^3 \eta_j(s_j)^3 ds_j. \quad (2.32)$$

To account for any obstacle location information obtained *a priori*, let P_q denote the location of the q th obstacle. The cost function includes the contribution

$$J_{o_j} = \int_0^{t_f} \left(\sum_{q=1}^l \frac{1}{\|p_j(t) - P_q\|} \right) dt = \int_0^{s_{f_j}} \left(\sum_{q=1}^l \frac{1}{\|p_j(s_j) - P_q\|} \right) ds_j, \quad (2.33)$$

where l is the total number of obstacles with known locations. The cost function that optimizes energy consumption and includes OA is given by

$$J_j = J_{e_j} + J_{o_j}. \quad (2.34)$$

The optimization problem can be solved in real-time using optimization techniques such as Hooke-Jeeves pattern direct-search algorithm, Nelder-Mead downhill simplex algorithm, or Strongin's information-statistical method as discussed in [97]. Figure 2.3 (a) shows admissible (solid) and inadmissible (dashed) paths for two vehicles. The dashed circles in Figure 2.3 (a) represent obstacles at known locations $P_x = [0 \ 10 \ 50]$, $P_y = [10 \ -8 \ 40]$ and $l = 3$. The cost function choice in (2.34) will determine which of the paths in Figure 2.3 (a) is the sub-optimal path that will avoid obstacles and maximize fuel efficiency. Figure 2.3 (b) shows the spatial separation between the two vehicles and Figure 2.4 shows admissible speed and acceleration profiles for two vehicles, where $v_{\min} = 0.25$ m/sec, $v_{\max} = 3.5$ m/sec, and $a_{\max} = 0.5$ m/sec².

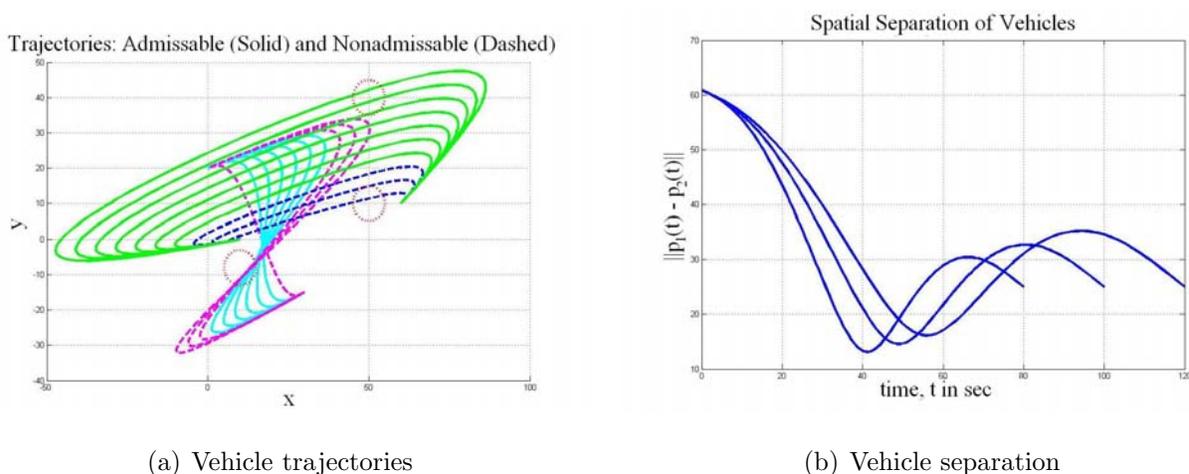


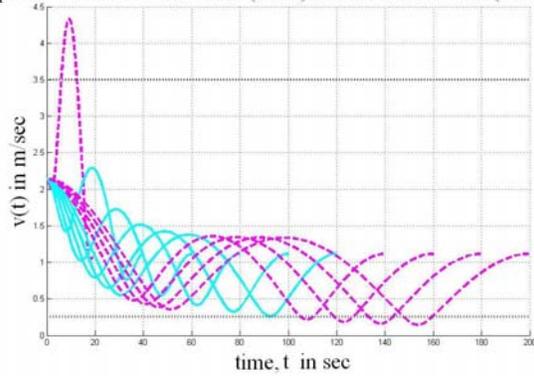
Figure 2.3: Path generation for two vehicles.

2.2.3 Analysis of Virtual Speed for a Single Vehicle

For this part of the analysis, the subscript j denoting the vehicle number is dropped for notational convenience and the analysis on $\eta(s)$ is completed for a single vehicle. By definition $\eta(s) = ds/dt$. This implies that

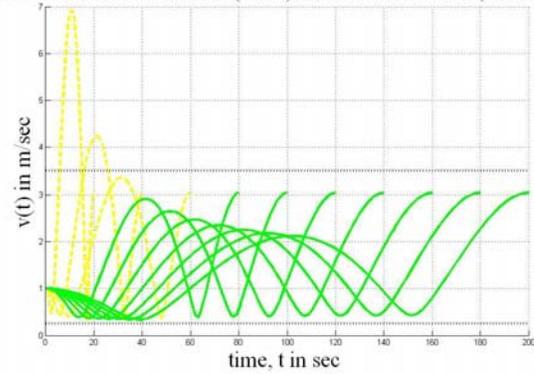
$$t = \int_0^s \frac{1}{\eta(s)} ds. \quad (2.35)$$

Speed: Vehicle 1 Admissible (Solid) and Nonadmissible (Dashed)



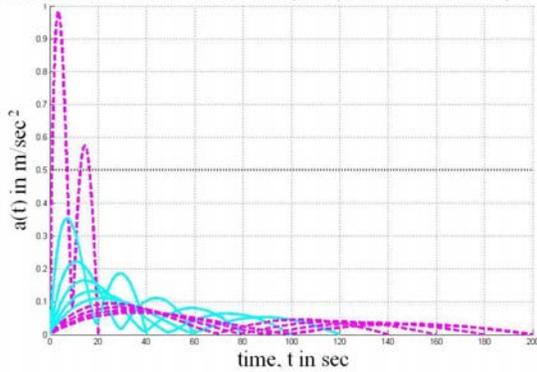
(a) Speed profiles: vehicle 1

Speed: Vehicle 2 Admissible (Solid) and Nonadmissible (Dashed)



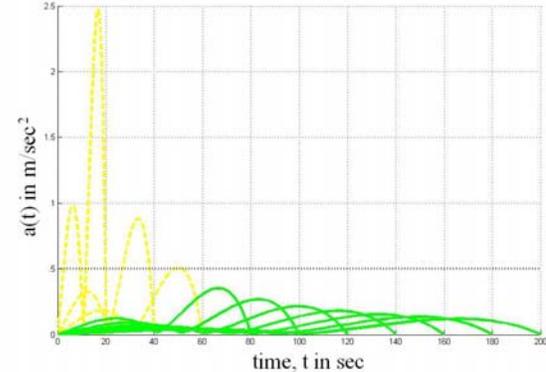
(b) Speed profiles: vehicle 2

Acceleration: Vehicle 1 Admissible (Solid) and Nonadmissible (Dashed)



(c) Acceleration profiles: vehicle 1

Acceleration: Vehicle 2 Admissible (Solid) and Nonadmissible (Dashed)



(d) Acceleration profiles: vehicle 2

Figure 2.4: Admissible vehicle speed and acceleration profiles.

Consider the simple choice when $\eta(s) = \eta \neq 0$ is a constant. Then

$$t = \frac{1}{\eta}s \quad \text{or} \quad s = \eta t$$

and the coordination objective is trivially satisfied since $t = t_f$ implies that

$$\frac{s}{s_f} = \frac{t}{t_f} = 1.$$

The following Proposition holds.

Proposition 1 *If $\eta(s) = \eta$ is a constant, then the N th-order path generated by $s = \eta t$ is identical to the N th-order path generated by $s = t$.*

Proof. Let $\eta(s) = \eta$ be a constant. The boundary conditions (2.25) reduce to

$$\frac{d^k}{ds^k} (p(0)) = \left(\frac{1}{\eta^k}\right) \frac{d^k}{dt^k} (p(0)), \quad \frac{d^k}{ds^k} (p(s_f)) = \left(\frac{1}{\eta^k}\right) \frac{d^k}{dt^k} (p(t_f)) \quad (2.36)$$

for $k = 0, 1, 2, \dots$, where the zeroth-derivative of a function is the function itself. Let d_0 and d_f denote the highest-order of the spatial-derivatives of the path that must meet pre-specified boundary conditions at the initial and final points, respectively. Consider two paths $p_1(t) = (x_1(t), x_2(t))$ (i.e. $s = t$) and $p_2(s) = (y_1(s), y_2(s))$, where $s = \eta t$, whose components are

$$x_i(t) = \sum_{k=0}^N a_{ik} t^k, \quad y_i(s) = \sum_{k=0}^N b_{ik} s^k, \quad i = 1, 2, \quad (2.37)$$

where $N \geq d_0 + d_f + 1$. From the definition of the polynomial paths and the boundary conditions (2.25), it follows that

$$a_{ik} = \frac{x_i^{(k)}(0)}{k!}, \quad b_{ik} = \frac{x_i^{(k)}(0)}{k! \eta^k}, \quad \text{for } k = 0, 1, 2, \dots, d_0, \quad (2.38)$$

where $x_i^{(k)}$ denotes the k th derivative of x_i with respect to t . For $p_2(s)$ and any $m \in \{0, 1, 2, \dots, d_f\}$, one has the corresponding linear equation for the m th spatial-derivative terminal boundary condition:

$$\sum_{l=0}^N \frac{d^m}{ds_f^m} (s_f^l) b_{il} = \frac{x_i^{(m)}(s_f)}{\eta^m}. \quad (2.39)$$

Equation (2.39) can be broken into two components as

$$\sum_{l=0}^{d_0} \frac{d^m}{ds_f^m} (s_f^l) b_{il} + \sum_{l=d_0+1}^N \frac{d^m}{ds_f^m} (s_f^l) b_{il} = \frac{x_i^{(m)}(s_f)}{\eta^m}. \quad (2.40)$$

Using the definition of b_{ik} in (2.25), one can compute

$$\sum_{l=0}^{d_0} \frac{d^m}{ds_f^m} (s_f^l) \frac{x_i^{(l)}(0)}{l! \eta^l} + \sum_{l=d_0+1}^N \frac{d^m}{ds_f^m} (s_f^l) b_{il} = \frac{x_i^{(m)}(s_f)}{\eta^m}. \quad (2.41)$$

Consider the case when $m \leq d_0$. This implies that (2.41) becomes

$$\sum_{l=m}^{d_0} \frac{s_f^{l-m} x_i^{(l)}(0)}{(l-m)! \eta^l} + \sum_{l=d_0+1}^N l \cdot (l-1) \cdot \dots \cdot (l-(m-1)) s_f^{l-m} b_{il} = \frac{x_i^{(m)}(s_f)}{\eta^m}. \quad (2.42)$$

Substituting $s_f = \eta t_f$ into (2.42) yields

$$\sum_{l=m}^{d_0} \frac{t_f^{l-m} x_i^{(l)}(0)}{(l-m)! \eta^m} + \sum_{l=d_0+1}^N l \cdot (l-1) \cdot \dots \cdot (l-(m-1)) \eta^{l-m} t_f^{l-m} b_{il} = \frac{x_i^{(m)}(s_f)}{\eta^m}. \quad (2.43)$$

Multiplying both sides of (2.43) by η^m gives

$$\sum_{l=m}^{d_0} \frac{t_f^{l-m} x_i^{(l)}(0)}{(l-m)!} + \sum_{l=d_0+1}^N l \cdot (l-1) \cdot \dots \cdot (l-(m-1)) \eta^l t_f^{l-m} b_{il} = x_i^{(m)}(s_f). \quad (2.44)$$

Similarly for the case when $s = t$, it can be verified that the linear equation corresponding to the m th spatial-derivative terminal boundary condition is

$$\sum_{l=m}^{d_0} \frac{t_f^{l-m} x_i^{(l)}(0)}{(l-m)!} + \sum_{l=d_0+1}^N l \cdot (l-1) \cdot \dots \cdot (l-(m-1)) t_f^{l-m} a_{il} = x_i^{(m)}(s_f). \quad (2.45)$$

Comparing (2.44) with (2.45), it follows that

$$a_{il} = b_{il} \eta^l, \quad \forall l = d_0 + 1, d_0 + 2, \dots, N. \quad (2.46)$$

If $m > d_0$, the first terms of (2.44) and (2.45) disappear, implying that

$$\sum_{l=m}^N l \cdot (l-1) \cdot \dots \cdot (l-(m-1)) \eta^l t_f^{l-m} b_{il} = \sum_{l=d_0+1}^N l \cdot (l-1) \cdot \dots \cdot (l-(m-1)) t_f^{l-m} a_{il} \quad (2.47)$$

and

$$a_{il} = b_{il} \eta^l \quad \forall l = m, \dots, N, \quad (2.48)$$

where $m \in \{d_0 + 1, \dots, d_f\}$ is arbitrary. Hence, (2.38), and (2.46) or (2.48) imply that

$$a_{il} = b_{il} \eta^l, \quad \forall l = 0, 1, \dots, N.$$

From (2.37), one can see that

$$x_i(t) = \sum_{k=0}^N a_{ik} t^k = \sum_{k=0}^N b_{ik} \eta^k t^k = \sum_{k=0}^N b_{ik} s^k, \quad i = 1, 2, \quad (2.49)$$

which completes the proof. \square

In general, $\eta(s)$ can be chosen as a polynomial of arbitrary order of the form

$$\eta(s) = (C_{1_p}s + C_{2_p})^p, \quad (2.50)$$

where $p = 2, 3, 4, \dots$ and $C_{1_p}, C_{2_p} \neq 0$ are constants that satisfy the boundary conditions $\eta(0) = \eta_0, \eta(s_f) = \eta_f$, and still satisfy the coordination objective $s/s_f = 1$ when $t = t_f$. Since

$$t = \int_0^s \frac{1}{(C_{1_p}\tau + C_{2_p})^p} d\tau = -\frac{1}{C_{1_p}(p-1)(C_{1_p}s + C_{2_p})^{p-1}} + \frac{1}{C_{1_p}(p-1)C_{2_p}^{p-1}}, \quad (2.51)$$

one can obtain

$$s = \frac{C_{2_p} \left(1 - (1 - C_{1_p}(p-1)C_{2_p}^{p-1}t)^{\frac{1}{p-1}}\right)}{C_{1_p}(1 - C_{1_p}(p-1)C_{2_p}^{p-1}t)^{\frac{1}{p-1}}}. \quad (2.52)$$

Hence,

$$\frac{s}{s_f} = \left(\frac{C_{2_p} \left(1 - (1 - C_{1_p}(p-1)C_{2_p}^{p-1}t)^{\frac{1}{p-1}}\right)}{C_{1_p}(1 - C_{1_p}(p-1)C_{2_p}^{p-1}t)^{\frac{1}{p-1}}} \right) \left(\frac{C_{1_p}(1 - C_{1_p}(p-1)C_{2_p}^{p-1}t_f)^{\frac{1}{p-1}}}{C_{2_p} \left(1 - (1 - C_{1_p}(p-1)C_{2_p}^{p-1}t_f)^{\frac{1}{p-1}}\right)} \right)$$

implies that $s/s_f = 1$ when $t = t_f$.

Proposition 2 If $\eta(s) = (C_{1_p}s + C_{2_p})^p$ and $\eta(0) = \eta_0, \eta(s_f) = \eta_f$, where $p = 2, 3, 4, \dots$, then

$$C_{2_p} = \eta_0^{\frac{1}{p}}, \quad C_{1_p} = \frac{\eta_f^{\frac{p-1}{p}} - \eta_0^{\frac{p-1}{p}}}{(p-1)\eta_0^{\frac{p-1}{p}}\eta_f^{\frac{p-1}{p}}t_f}, \quad (2.53)$$

and

$$s_f = \frac{(p-1)\eta_0^{\frac{p-1}{p}}\eta_f^{\frac{p-1}{p}}(\eta_f^{\frac{1}{p}} - \eta_0^{\frac{1}{p}})t_f}{\eta_f^{\frac{p-1}{p}} - \eta_0^{\frac{p-1}{p}}}, \quad (2.54)$$

$$s = \left(\frac{(p-1)\eta_0\eta_f^{\frac{p-1}{p}}t_f}{\eta_f^{\frac{p-1}{p}} - \eta_0^{\frac{p-1}{p}}} \right) \frac{\eta_f^{\frac{1}{p}}t_f^{\frac{1}{p-1}} - \left(\eta_f^{\frac{p-1}{p}}t_f - t(\eta_f^{\frac{p-1}{p}} - \eta_0^{\frac{p-1}{p}}) \right)^{\frac{1}{p-1}}}{\left(\eta_f^{\frac{p-1}{p}}t_f - t(\eta_f^{\frac{p-1}{p}} - \eta_0^{\frac{p-1}{p}}) \right)^{\frac{1}{p-1}}}.$$

Consequently,

$$\eta(t) = \left(\frac{\eta_0^{\frac{1}{p}} \eta_f^{\frac{1}{p}} t^{\frac{1}{p-1}}}{\left(\eta_f^{\frac{p-1}{p}} t_f - t(\eta_f^{\frac{p-1}{p}} - \eta_0^{\frac{p-1}{p}}) \right)^{\frac{1}{p-1}}} \right)^p = \frac{\eta_0 \eta_f t_f^{\frac{p}{p-1}}}{\left(\eta_f^{\frac{p-1}{p}} t_f - t(\eta_f^{\frac{p-1}{p}} - \eta_0^{\frac{p-1}{p}}) \right)^{\frac{p}{p-1}}}. \quad (2.55)$$

Proof. Consider $\eta(s) = (C_{1_p}s + C_{2_p})^p$, where C_{1_p}, C_{2_p} and s_f are given in (2.53) and (2.54), respectively, and $p \in 2, 3, 4, \dots$ is a constant. One can show that $\eta(s)$ satisfies the initial conditions since $s = 0$ trivially implies $\eta(0) = C_{2_p}^p = \eta_0$. Next, if $s = s_f$, one obtains

$$\begin{aligned} (C_{1_p}s + C_{2_p})^p &= \left(\left(\frac{\frac{p-1}{p} \eta_f^{\frac{p-1}{p}} - \frac{p-1}{p} \eta_0^{\frac{p-1}{p}}}{(p-1)\eta_0^{\frac{p-1}{p}} \eta_f^{\frac{p-1}{p}} t_f} \right) \left(\frac{(p-1)\eta_0^{\frac{p-1}{p}} \eta_f^{\frac{p-1}{p}} (\eta_f^{\frac{1}{p}} - \eta_0^{\frac{1}{p}}) t_f}{\eta_f^{\frac{p-1}{p}} - \eta_0^{\frac{p-1}{p}}} \right) + \eta_0^{\frac{1}{p}} \right)^p \\ &= \left(\eta_f^{\frac{1}{p}} - \eta_0^{\frac{1}{p}} + \eta_0^{\frac{1}{p}} \right)^p \\ &= \eta_f. \end{aligned}$$

Hence, $\eta(s)$ satisfies the boundary conditions $\eta(0) = \eta_0, \eta(s_f) = \eta_f$. Furthermore, one can verify using the definition of s in (2.52) that

$$s = \frac{C_{2_p} \left(1 - (1 - C_{1_p}(p-1)C_{2_p}^{p-1} t)^{\frac{1}{p-1}} \right)}{C_{1_p} \left(1 - C_{1_p}(p-1)C_{2_p}^{p-1} t \right)^{\frac{1}{p-1}}} \quad (2.56)$$

$$\begin{aligned} &= \frac{\eta_0^{\frac{1}{p}} \left(1 - \left(1 - \frac{\frac{p-1}{p} \eta_f^{\frac{p-1}{p}} - \frac{p-1}{p} \eta_0^{\frac{p-1}{p}}}{(p-1)\eta_0^{\frac{p-1}{p}} \eta_f^{\frac{p-1}{p}} t_f} (p-1)\eta_0^{\frac{p-1}{p}} t \right)^{p-1} \right)}{\frac{\frac{p-1}{p} \eta_f^{\frac{p-1}{p}} - \frac{p-1}{p} \eta_0^{\frac{p-1}{p}}}{(p-1)\eta_f^{\frac{p-1}{p}} \eta_0^{\frac{p-1}{p}} t_f} \left(1 - \frac{\frac{p-1}{p} \eta_f^{\frac{p-1}{p}} - \frac{p-1}{p} \eta_0^{\frac{p-1}{p}}}{(p-1)\eta_0^{\frac{p-1}{p}} \eta_f^{\frac{p-1}{p}} t_f} (p-1)\eta_0^{\frac{p-1}{p}} t \right)^{p-1}} \\ &= \frac{\eta_0^{\frac{1}{p}} \left(1 - \left(\frac{\frac{p-1}{p} \eta_f^{\frac{p-1}{p}} t_f + t(\eta_f^{\frac{p-1}{p}} - \eta_0^{\frac{p-1}{p}})}{\eta_f^{\frac{p-1}{p}}} t_f \right)^{\frac{1}{p-1}} \right)}{\frac{\frac{p-1}{p} \eta_f^{\frac{p-1}{p}} - \frac{p-1}{p} \eta_0^{\frac{p-1}{p}}}{(p-1)\eta_f^{\frac{p-1}{p}} \eta_0^{\frac{p-1}{p}} t_f} \left(\frac{\frac{p-1}{p} \eta_f^{\frac{p-1}{p}} t_f + t(\eta_f^{\frac{p-1}{p}} - \eta_0^{\frac{p-1}{p}})}{\eta_f^{\frac{p-1}{p}}} t_f \right)^{\frac{1}{p-1}}} \end{aligned} \quad (2.57)$$

$$= \left(\frac{(p-1)\eta_0\eta_f^{\frac{p-1}{p}}t_f}{\eta_f^{\frac{p-1}{p}} - \eta_0^{\frac{p-1}{p}}} \right) \frac{\left(\eta_f^{\frac{1}{p}}t_f^{\frac{1}{p-1}} - \left(\eta_f^{\frac{p-1}{p}}t_f + t(\eta_f^{\frac{p-1}{p}} - \eta_0^{\frac{p-1}{p}}) \right)^{\frac{1}{p-1}} \right)}{\left(\eta_f^{\frac{p-1}{p}}t_f + t(\eta_f^{\frac{p-1}{p}} - \eta_0^{\frac{p-1}{p}}) \right)^{\frac{1}{p-1}}},$$

which matches the definition of s given in (2.54). Substituting the definition of C_{1p}, C_{2p} , and s given in (2.53) and (2.54), respectively, yields the expression in (2.55). \square

Figures 2.6 and 2.7 show ten cases of $\eta(s)$ for $p = 1, 2, \dots, 10$. Figure 2.5 shows the parameter s_f as p increases. The blue solid lines represent the case when $\eta_0 > \eta_f$ with $\eta_0 = 2, \eta_f = 1$ and the red dashed lines represent the case when $\eta_0 < \eta_f$ with $\eta_0 = 1, \eta_f = 2$. One can see that increasing the order of $\eta(s)$ yields insignificant changes to the path for this example. The speed and acceleration profiles yield small, but noticeable changes (Figure 2.7). The sign of $(\eta_0 - \eta_f)$ changes the slope of $\eta(s)$, reflecting the resulting paths and profiles. It is of interest to note the behavior of $\eta(s)$, and hence, the path $p(t)$, as p and the order of $\eta(s)$ increases. As p increases, limits are obtained on $\eta(s)$, s_f , and s .

Proposition 3 Given $\eta(s) = (C_{1p}s + C_{2p})^p$, where C_{1p}, C_{2p} are given in (2.53), and s_f and s are defined in (2.54), the following limits are obtained:

$$\begin{aligned} \lim_{p \rightarrow \infty} \eta(s) &= \frac{\eta_0\eta_f t_f}{\eta_f t_f - t(\eta_f - \eta_0)}, \\ \lim_{p \rightarrow \infty} s_f &= \frac{\ln\left(\frac{\eta_f}{\eta_0}\right)\eta_0\eta_f t_f}{\eta_f - \eta_0}, \\ \lim_{p \rightarrow \infty} s &= \ln\left(\frac{\eta_f t_f}{\eta_f t_f - t(\eta_f - \eta_0)}\right) \frac{\eta_0\eta_f t_f}{\eta_f - \eta_0}. \end{aligned} \tag{2.58}$$

To illustrate the effect of p on the path generation and the results of Propositions 1 - 3, a simulation example is considered. The black dashed line in Figure 2.5 shows the limit on s_f as $p \rightarrow \infty$. The value of s_f is seen to converge to this limit as p increases for $p \in [1, 20]$. Next, Figure 2.6 (a) shows $\eta(s)$ for $p \in [1, 20]$ for two sets of initial conditions. The first is when $\eta_0 < \eta_f$ with $\eta_0 = 1, \eta_f = 2$. This is shown in Figure 2.6 (a) by the blue solid lines. The red dashed lines depict the case when $\eta_0 > \eta_f$ with $\eta_0 = 2, \eta_f = 1$, respectively.

Changing the sign of $\eta_0 - \eta_f$ changes the slope of $\eta(s)$, reflecting the image of $\eta(s)$. The limits on $\eta(s)$ as $p \rightarrow \infty$ for both cases are shown by the black dotted lines. For comparison, the cases for constant $\eta(s)$, that is, $\eta(s) = \eta_0 = 1$ and $\eta(s) = \eta_0 = 2$ are shown and labeled in Figure 2.6 (a). As the order of $\eta(s)$ increases, the difference between any two consecutive orders of $\eta(s)$ decrease for this example. Figure 2.6 (b) shows the generated paths for the same scenarios. From Proposition 1, $\eta = 1$ and $\eta = 2$ will generate the same paths, which is shown by the green dash-dot line. The path generated by constant $\eta(s)$ differs from the higher order cases, but further changes in the order of $\eta(s)$ cause small changes to the path $p(s)$ for this simulation example (Figure 2.6 (b)).

The limits on the associated speed and acceleration profiles of the path are shown in black dashed lines in Figures 2.6 - 2.7. Again, blue solid lines show the case when $\eta_0 = 1, \eta_f = 2$ and red dashed lines represent $\eta_0 = 2, \eta_f = 1$. The constant $\eta(s)$ case is indicated by the green dash-dot line. There is a large difference between the constant $\eta(s)$ cases, but the changes to the path and its resulting speed and acceleration profiles decrease with increasing p for this example. The path and speed and acceleration profiles converge to their respective limits as defined by Proposition 3.

2.2.4 Path Generation and Bezier Curves

It is useful to note that (2.14) is not the only polynomial choice that can be used to represent the path. For instance, Bezier curves are known to be an inexpensive way to approximate unknown functions and have been suggested to be advantageous over other approximation functions [77]. Recall that a Bezier curve is of the form:

$$B(\underline{s}) = \sum_{k=0}^N \binom{N}{k} (1 - \underline{s})^{N-k} \underline{s}^k b_k, \quad (2.59)$$

where the b_k 's are the coefficients that ensure the path meets the boundary conditions by construction and $\underline{s} \in [0, 1]$. A choice for \underline{s} is $\underline{s} = s/s_f$, for example. This implies that \underline{s} and s are related by a constant multiplier $1/s_f$. The following Corollary of Proposition 1 holds.

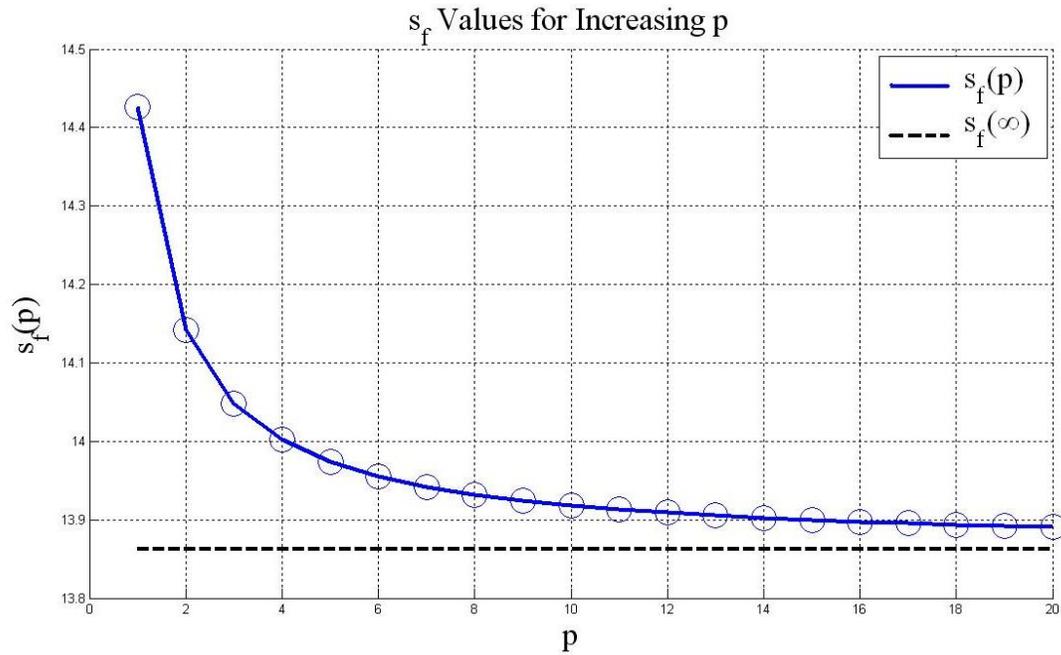
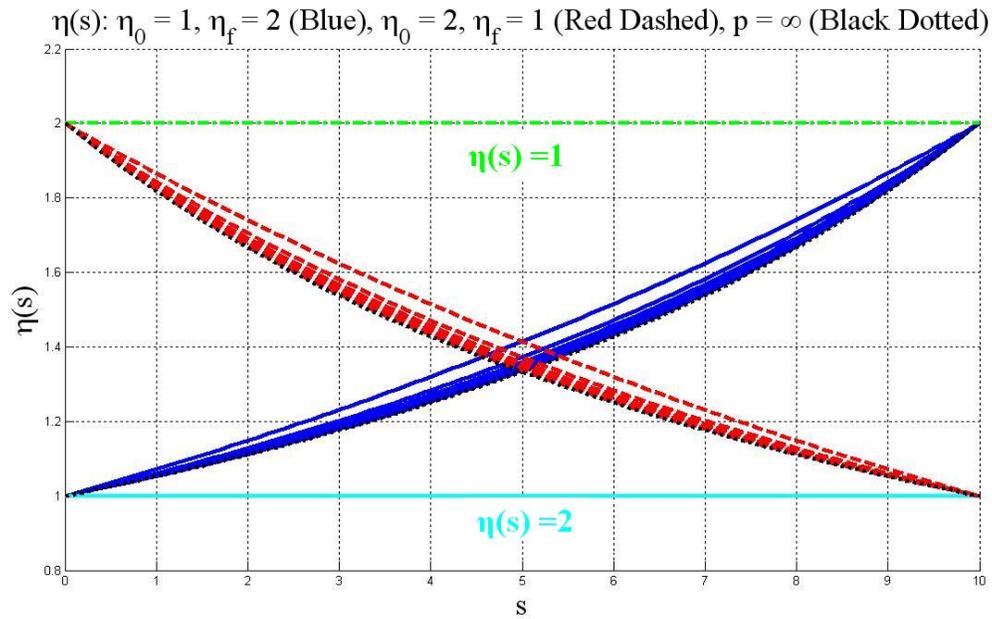


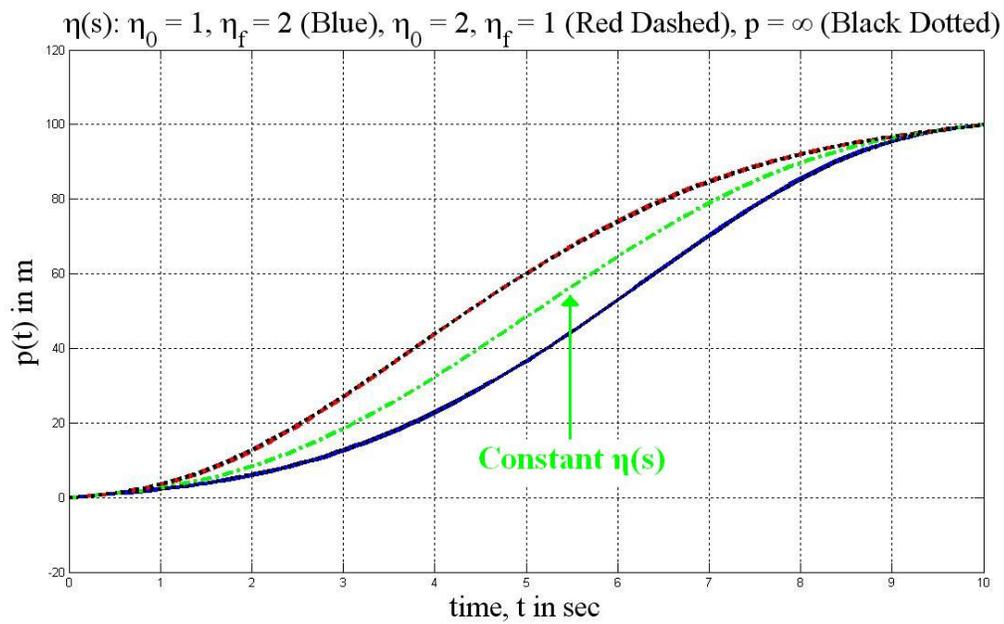
Figure 2.5: Final value of the parameter s_f as p increases.

Corollary 1 *If $\underline{s} = s/s_f$, where s is the virtual arc, then the path generated by an N th-order Bezier curve of the form (2.59) is equivalent to the N th-order path generated by the polynomial in (2.14) for any choice of s .*

Indeed, the virtual speeds of the two parameters $\underline{s} = s/s_f$ and s are related by a constant so that Proposition 1 holds and the paths generated will be identical. Hence, the Bezier curves in (2.59) will generate the same trajectories as the polynomials in (2.14) for any s . As a specific example, consider $\underline{s} = t/t_f$. It follows that $\underline{\eta} = 1/t_f$, which is another representation of the case when $\underline{s} = t$. Using Bezier curves for path generation with $\underline{s} = t$ is another representation of this simpler case when the polynomials in (2.14) are used with $\eta = 1/t_f$.

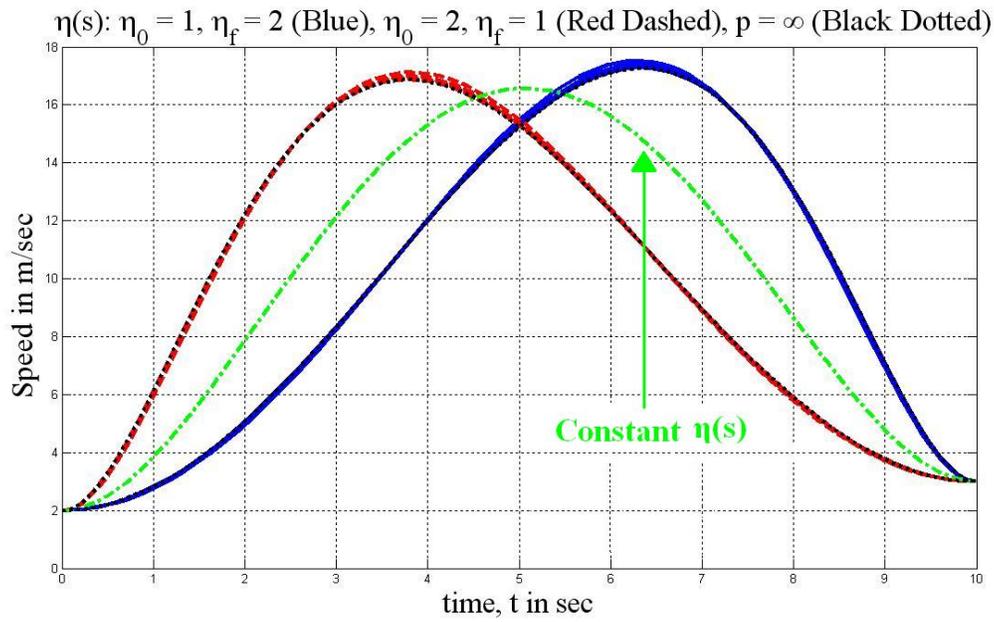


(a) Function $\eta(s)$ with increasing p

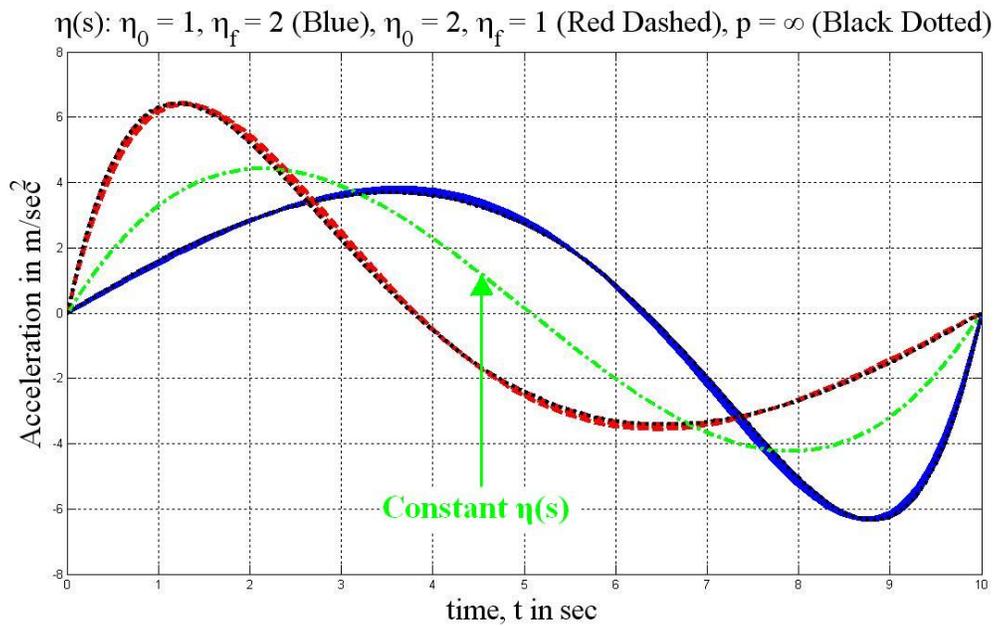


(b) Vehicle trajectories with increasing p

Figure 2.6: Comparison of vehicle trajectories with increasing p .



(a) Speed profiles with increasing p



(b) Acceleration profiles with increasing p

Figure 2.7: Comparison of speed and acceleration profiles with increasing p.

The minimum order of the Bezier curve corresponds to the minimum order N^* of the polynomials in (2.14). Hence,

$$x_{B_i}(\underline{s}) = \sum_{k=0}^N \binom{N}{k} (1 - \underline{s})^{N-k} \underline{s}^k b_{ik}. \quad (2.60)$$

The coefficients b_{ik} can be computed with the boundary conditions as the solution to a linear system of equations. With a slight abuse of notation, $p'_B(\underline{s}), p''_B(\underline{s})$ denote the first and second derivatives with respect to \underline{s} in the following. Table 2.2 shows how to solve the coefficients b_{ik} for 5th and 6th degree Bezier paths.

Let $\underline{\eta} = 1/t_f = \underline{\eta}(0) = \underline{\eta}(1)$ since $\underline{\eta}$ is constant. The shorthand notation $\underline{\eta}(0) = \underline{\eta}_0$ and $\underline{\eta}(1) = \underline{\eta}_f$ is introduced for readability. It follows that the temporal and spatial derivatives satisfy

$$\begin{aligned} \dot{p}(t) &= \underline{\eta}(\underline{s}) p'_B(\underline{s}) \\ \ddot{p}(t) &= \underline{\eta}(\underline{s}) \underline{\eta}'(\underline{s}) p'_B(\underline{s}) + \underline{\eta}^2(\underline{s}) p''_B(\underline{s}), \\ &= \underline{\eta}^2(\underline{s}) p''_B(\underline{s}), \end{aligned} \quad (2.61)$$

and the boundary conditions are computed as:

$$\begin{aligned} p'_B(0) &= \frac{\dot{p}(0)}{\underline{\eta}_0}, & p'_B(1) &= \frac{\dot{p}(t_f)}{\underline{\eta}_f}, \\ p''_B(0) &= \frac{\ddot{p}(0)}{\underline{\eta}_0^2}, & p''_B(1) &= \frac{\ddot{p}(t_f)}{\underline{\eta}_f^2}. \end{aligned} \quad (2.62)$$

The optimization problem that solves path generation for the rendezvous problem for n vehicles using Bezier curves is stated formally as that of computing

$$t_{B_f}^{\text{opt}} = \min_{t_f \in [t_1, t_2]} \sum_{j=1}^n w_{B_j} J_{B_j}, \quad (2.63)$$

subject to spatial deconfliction

$$\|p_{B_j}(t) - p_{B_k}(t)\| > \underline{\Xi}, \quad \forall t \in [0, t^{\text{opt}}], \quad (2.64)$$

Table 2.2: Examples of computation of the coefficients of 5th and 6th order Bezier paths

5 th order		
Boundary conditions	$x_{B_i}(0), x_{B_i}(1), x''_{B_i}(0), x_{B_i}(1), x'_{B_i}(1), x''_{B_i}(1)$	
d_0/d_f	2/2	
N^*/N	5/5	
Linear algebraic matrix equation to solve for the coefficients b_{ik}	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -5 & 5 & 0 & 0 & 0 & 0 \\ 20 & -40 & 20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -5 & 5 \\ 0 & 0 & 0 & 20 & -40 & 20 \end{bmatrix}$	$\begin{bmatrix} b_{i0} \\ b_{i1} \\ b_{i2} \\ b_{i3} \\ b_{i4} \\ b_{i5} \end{bmatrix} = \begin{bmatrix} x_{B_i}(0) \\ x'_{B_i}(0) \\ x''_{B_i}(0) \\ x_{B_i}(1) \\ x'_{B_i}(1) \\ x''_{B_i}(1) \end{bmatrix}$
6 th order		
Boundary conditions	$x_{B_i}(0), x'_{B_i}(0), x''_{B_i}(0), x'''_{B_i}(0), x_{B_i}(1), x'_{B_i}(1), x''_{B_i}(1)$	
d_0/d_f	3/2	
N^*/N	5/6	
Linear algebraic matrix equation to solve for the coefficients b_{ik}	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -5 & 5 & 0 & 0 & 0 & 0 \\ 20 & -40 & 20 & 0 & 0 & 0 \\ -60 & 180 & -180 & 60 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -5 & 5 \\ 0 & 0 & 0 & 20 & -40 & 20 \end{bmatrix}$	$\begin{bmatrix} b_{i0} \\ b_{i1} \\ b_{i2} \\ b_{i3} \\ b_{i4} \\ b_{i5} \\ b_{i6} \end{bmatrix} = \begin{bmatrix} x_{B_i}(0) \\ x'_{B_i}(0) \\ x''_{B_i}(0) \\ x'''_{B_i}(0) \\ x_{B_i}(1) \\ x'_{B_i}(1) \\ x''_{B_i}(1) \end{bmatrix}$

and speed and acceleration constraints

$$v_{\min} \leq \underline{\eta}_j(s) \|p'_{B_j}(\underline{s}_j)\| \leq v_{\max}, \quad \|p''_{B_j}(\underline{s}_j) \underline{\eta}_j^2(\underline{s}_j)\| \leq a_{\max}, \quad (2.65)$$

where w_{B_j} are positive weights for $i = 1, 2, \dots, n$ and J_{B_j} is the associated cost function that optimizes energy consumption and includes OA given by

$$J_{B_j} = \int_0^1 \left(\|p'_{B_j}(\underline{s}_j)\|^3 \underline{\eta}_j(\underline{s}_j)^3 + \left(\sum_{q=1}^l \frac{1}{\|p_{B_j}(\underline{s}_j) - P_q\|} \right) \right) d\underline{s}_j. \quad (2.66)$$

Figure 2.8 shows the sub-optimal path generated by the polynomials in (2.14) using a linear η function and constant η function compared to the optimal solution obtained from Pontryagin's Maximum Principle [49]. Figure 2.9 shows the associated speed and acceleration profiles, respectively. The sub-optimal path generated by the polynomials in (2.14) with linear $\eta(s)$ is higher in cost than the sub-optimal path generated by using constant $\eta(s)$. Note that both polynomial paths generated are more time-efficient than the optimal solution at the price of a higher cost. The polynomial paths require higher fuel consumption and approach the nearby obstacle at a closer distance as compared to the optimal solution. The optimal trajectory generates a cost function of $J_o = 525.9273$, the polynomials in (2.14) for a linear η and a constant η generate $J = 647.7364$ and $J = 602.4862$, respectively. It can be seen that the path generation method provides a good approximation to the optimal solution for this example. For this case, keeping η constant and letting $s = t$ yields the path that is closest to optimal. Since $s = t$ imposes the speed profile (2.11), keeping η as a general polynomial function allows for non-constrained speed profiles and affords additional flexibility.

In order to follow the desired paths generated by the polynomials in (2.14), a path following controller must be implemented. This path following controller defines desired commands for the vehicle that guarantee path following and is determined by the dynamics of the vehicle. To obtain the dynamics of the vehicle, Lagrangian methods [34] are employed, which is described in the next Section.

Remark 2 The polynomials (2.14) and Bezier curve (2.59) are two of many polynomial functions that can be used in the path generation method outlined above. The two above were chosen due to their simple forms and ease of computation. More complicated types of polynomials functions include Chebyshev polynomials, Legendre polynomials, Hermite polynomials, and Dickson polynomials, for example [1].

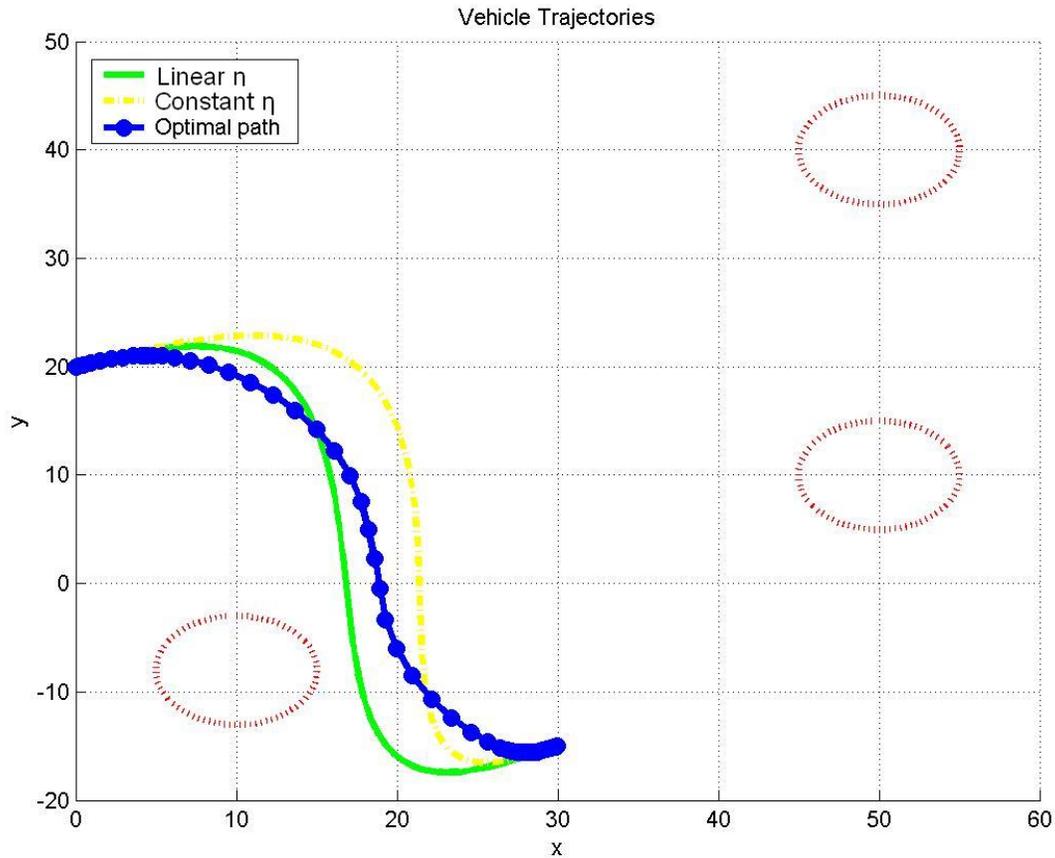
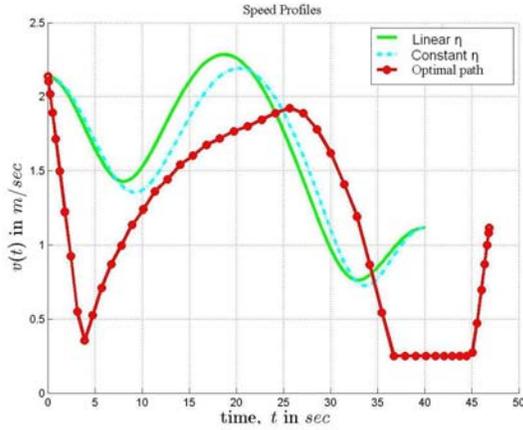
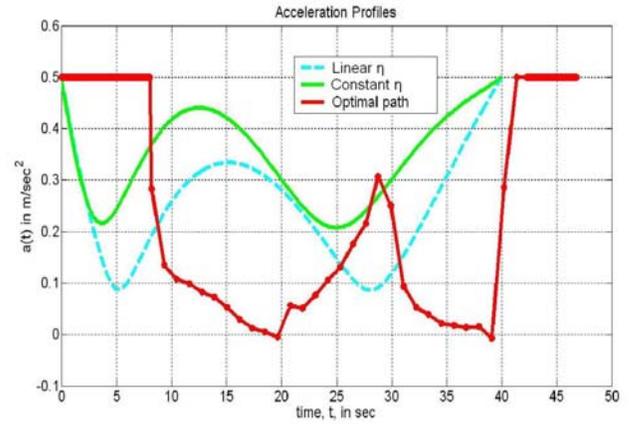


Figure 2.8: Trajectory comparison of polynomial path and optimal path.



(a) Speed profiles



(b) Acceleration profiles

Figure 2.9: Speed and acceleration profile comparison of polynomial path and optimal path.

2.3 Mobile Robot Equations of Motion

Consider a mechanical system with n -dimensional generalized coordinates q which are subject to w nonholonomic (i.e. nonintegrable) independent constraints of the form

$$C(q)\dot{q} = 0, \quad (2.67)$$

where $C(q) \in \mathbb{R}^{w \times n}$ is full rank [34]. For example, many wheeled vehicles are subject to constraints of rolling without slipping. Buses, cars, and mobile robots are examples of nonholonomic systems. Let $d_1(q), d_2(q), \dots, d_{n-w}(q)$ be a set of smooth, linearly independent vectors in $\mathcal{N}(C)$, the null space of $C(q)$. That is,

$$C(q)d_i(q) = 0, \quad i = 1, 2, \dots, n - w. \quad (2.68)$$

The number of degrees of freedom is usually defined by the difference $n - w$ [70]. For example, a unicycle is a 2-degree of freedom system [64]. Let $D(q) \in \mathbb{R}^{n \times n-w}$ be the full rank matrix defined by $D(q) = [d_1(q) \ d_2(q) \ \dots \ d_{n-w}(q)]$ such that

$$C(q)D(q) = 0. \quad (2.69)$$

It follows that $\dot{q} \in \Delta$, where

$$\Delta = \text{span}\{d_1(q), d_2(q), \dots, d_{n-w}(q)\}. \quad (2.70)$$

Using the Euler-Lagrange formulation [34], the dynamic equation of the mechanical system with nonholonomic constraints (2.67) is given by:

$$M(q)\ddot{q} + V(q, \dot{q}) = E(q)\mu - C^\top(q)\lambda, \quad (2.71)$$

where $M(q) \in \mathbb{R}^{n \times n}$ is the symmetric positive definite inertia matrix, $V(q, \dot{q}) \in \mathbb{R}^n$ is a vector of the centripetal and Coriolis force term, $E(q) \in \mathbb{R}^{n \times m}$ is the full-rank input transformation matrix, μ is the m -dimensional control input vector, and λ is the w -vector of Lagrange multipliers corresponding to the constraints. In the case of the wheeled mobile robot, the torque vector is applied to driving wheels. It is assumed that the mechanical system is not underactuated so that $m = n - w$. The constraints (2.67) and (2.69) imply that there exists an $(n - w)$ -dimensional vector z such that

$$\dot{q} = D(q)z. \quad (2.72)$$

Differentiating (2.72), one obtains

$$\ddot{q} = \dot{D}(q)z + D(q)\dot{z}. \quad (2.73)$$

Substituting this expression for \ddot{q} in (2.71) and pre-multiplying by $D^\top(q)$ yields

$$D^\top(q) \left(M(q)\dot{D}(q)z + M(q)D(q)\dot{z} + V(q, \dot{q}) \right) = D^\top(q)E(q)\mu - (C(q)D(q))^\top \lambda. \quad (2.74)$$

From (2.69), it can be seen that $(C(q)D(q))^\top = 0$ so that the Lagrange multiplier terms in (2.74) disappear. This gives the dynamic equation which has embedded in it the nonholonomic constraints in (2.67) as

$$D^\top(q) \left(M(q)\dot{D}(q)z + M(q)D(q)\dot{z} + V(q, \dot{q}) \right) = D^\top(q)E(q)\mu. \quad (2.75)$$

Pre-multiplying (2.75) by $(D^\top(q)E(q))^{-1} \in \mathbb{R}^{m \times m}$, which is well defined since $D^\top(q)$ and $E(q)$ are both full rank, gives

$$H(q)\dot{z} + G(q, z) = \mu, \quad (2.76)$$

where $H(q) \in \mathbb{R}^{m \times m}$ and $G(q, z) \in \mathbb{R}^m$ are as follows:

$$\begin{aligned} H(q) &= (D^\top(q)E(q))^{-1}D^\top(q)M(q)D(q), \\ G(q, z) &= (D^\top(q)E(q))^{-1}D^\top(q) \left(M(q)\dot{D}(q)z + V(q, \dot{q}) \right). \end{aligned} \quad (2.77)$$

The vehicle considered in this analysis is a unicycle-type that has two identical parallel, non-deformable rear wheels and a steering front wheel (Figure 2.10). The rear wheels are powered by a motor which generates a control torque. This allows generation of a control force and control torque to be used for path following. Since path following is concerned with the position and orientation of the vehicle, the generalized coordinates q are

$$q = [x_I(t) \quad y_I(t) \quad \psi(t)]^\top. \quad (2.78)$$

The coordinates $x_I(t), y_I(t)$ are expressed in a fixed inertial frame and $\psi(t)$ is the yaw angle of the vehicle. It is assumed that the contact between the wheels is pure rolling and non-slipping, which is expressed as the constraint

$$\dot{y}_I(t) \cos \psi(t) - \dot{x}_I(t) \sin \psi(t) = 0. \quad (2.79)$$

This implies that $n = 3, w = 1, m = n - w = 2$ and $z \in \mathbb{R}^2$. Let $z = [v(t) \quad r(t)]^\top$, where $v(t)$ is the speed of the vehicle and $r(t)$ is its rate of rotation. It is easily computed that

$$C(q) = [-\sin \psi(t) \quad \cos \psi(t) \quad 0], \quad D(q) = \begin{bmatrix} \cos \psi(t) & 0 \\ \sin \psi(t) & 0 \\ 0 & 1 \end{bmatrix}. \quad (2.80)$$

From (2.72), the simplified vehicle kinematic equations of a mobile robot are

$$\begin{aligned} \dot{x}_I(t) &= v(t) \cos \psi(t), \\ \dot{y}_I(t) &= v(t) \sin \psi(t), \\ \dot{\psi}(t) &= r(t). \end{aligned} \quad (2.81)$$

The vehicle is assumed to have access to information collected by sensors onboard so that the quantities $\psi(t), v(t), r(t)$ are available for feedback. For the kinematics of (2.81) with

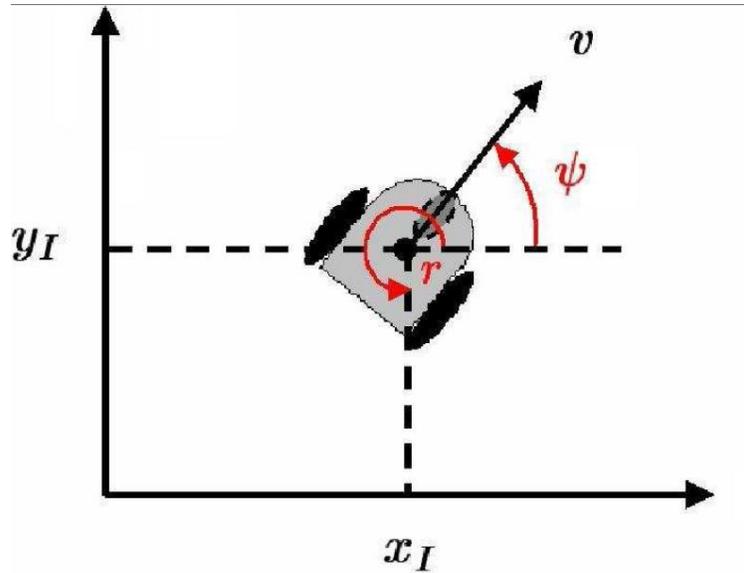


Figure 2.10: Unicycle-type mobile robot.

respect to the nonholonomic form in (2.2), $n = 3, m = 2$, and $p(s) = [p_x(s) \ p_y(s) \ p_\psi(s)]^\top$, where $p_x(s), p_y(s)$ are the x and y -coordinates of the path and $p_\psi(s)$ is the desired orientation along the path. The control vector fields are

$$G_1(s) = \begin{bmatrix} \cos(p_\psi(s)) \\ \sin(p_\psi(s)) \\ 0 \end{bmatrix}, \quad G_2(s) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \quad (2.82)$$

Remark 3 Note that the dimension of the path is equal to the number of states, n , and that the desired orientation $p_\psi(s)$ is not required in order to define a desired path. In fact, $p_\psi(s)$ can be derived from $p_x(s)$ and $p_y(s)$ directly via the algebraic relationship

$$p_\psi(s) = \arctan\left(\frac{p'_y(s)}{p'_x(s)}\right), \quad (2.83)$$

where the prime in $p'_y(s), p'_x(s)$ denotes derivation with respect to s . The importance and necessity of $p_\psi(s)$ will be made clear in the OA component in Chapter 3.

2.4 2D Path Following

Let $p(s)$ denote the desired path that the vehicle has to follow and let $\{F\}$ be a Frenet-Serret frame which is attached to a generic point O on the path. The point O plays the role of the center of mass of a “virtual” vehicle to be followed [64]. In Ref. [64], this point was simply defined as the projection of the vehicle to the path and led to local convergence results due to singularity problems. This is in contrast to the method in Ref. [87] for the two-dimensional case, in which O evolves according to an extra “virtual” control law $\dot{s}(t)$, where $s(t)$ is the path length along $p(t)$. Adding this extra degree of freedom to point O eliminates the singularity problem in Ref. [64] and allows for global convergence results for path following. This strategy was first exploited in Ref. [87] and has been applied in several cases to coordinated path following in Refs. [26, 28, 27, 43, 44], to name a few. Since $s(t)$ depends upon the motion of O along $p(s)$, then $p(s) = p(s(t)) = p(t)$. Following Ref. [28], the path following kinematic error dynamics are

$$\text{Kinematics : } \begin{cases} \dot{x}_e(t) = (y_e(t)c_c(s(t)) - 1)\dot{s}(t) + v(t) \cos \psi_e(t), \\ \dot{y}_e(t) = -x_e(t)c_c(s(t))\dot{s}(t) + v(t) \sin \psi_e(t), \\ \dot{\psi}_e(t) = r(t) - c_c(s(t))\dot{s}(t), \end{cases} \quad (2.84)$$

where $x_e(t), y_e(t)$ denote the x and y -coordinates of the path following errors between the vehicle and the virtual vehicle whose center of mass is located at point O , and $c_c(t)$ is the path curvature at O determined by $p(t)$ (Figure 2.11). That is, if point O has the coordinates $(x_F(t), y_F(t))$ in the inertial frame, then $x_e(t) = x_I(t) - x_F(t)$ and $y_e(t) = y_I(t) - y_F(t)$. If the tangent of the path at point O makes an angle $\psi_F(t)$ with respect to the inertial frame, then $\psi_e(t) = \psi(t) - \psi_F(t)$ denotes the path following angular error.

The inputs to the kinematic system in (2.84) are $v(t)$ and $r(t)$, which evolve according to

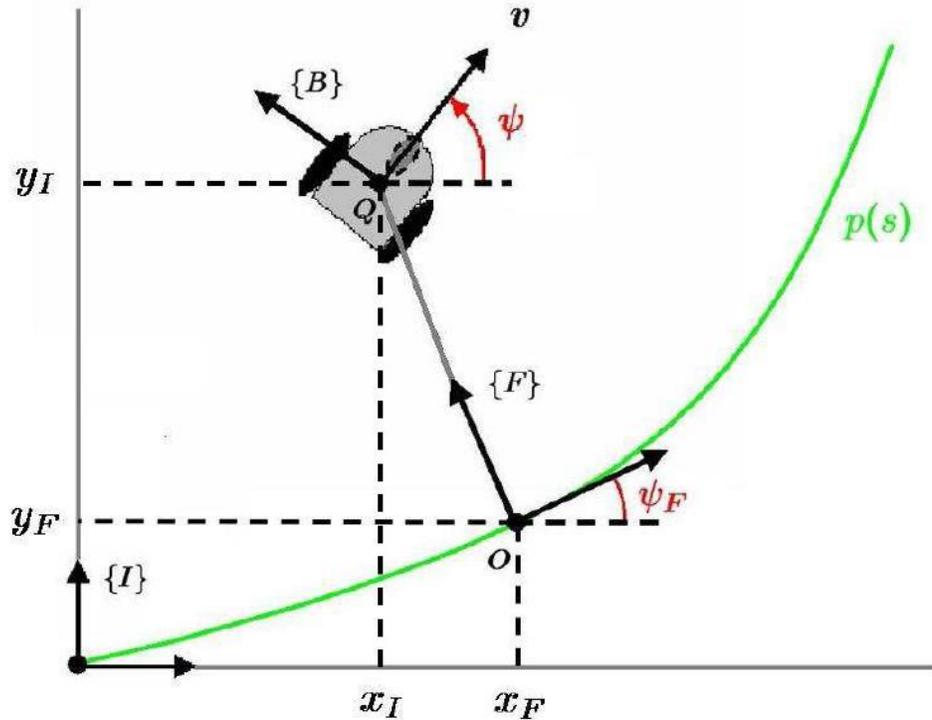


Figure 2.11: Body-fixed and Frenet-Serret frames.

the simplified dynamics [28]

$$\text{Dynamics : } \begin{cases} \dot{v}(t) = \frac{F(t)}{m}, \\ \dot{r}(t) = \frac{N(t)}{I}, \end{cases} \quad (2.85)$$

where m denotes the mass of the robot, I is the moment of inertia about the vertical body-axis, and F and N are the total force and torque, respectively, applied to the vehicle. Without loss of generality, let $m = I = 1$ in the appropriate units. The objective of the path following controller in the absence of obstacles is to drive the vehicle to the path by forcing the path following errors $x_e(t)$, $y_e(t)$, $\psi_e(t)$ and the speed error $v(t) - v_d(t)$ to zero, where $v_d(t)$ is a desired speed profile. To ensure that the shape of the robot's trajectory while approaching the path is within the capability of the vehicle [64], consider the shaping function for $\psi_e(t)$:

$$\sigma(t) = -\text{sign}(v(t)) \sin^{-1} \left(\frac{k_2 y_e(t)}{|y_e(t)| + \epsilon_0} \right), \quad (2.86)$$

where $\epsilon_0 > 0$ and $k_2 \in (0, 1]$ are constants. The following shaping function is introduced for $r(t)$:

$$\phi(t) = c_c(s(t))\dot{s}(t) + \dot{\sigma}(t) - k_3(\psi_e(t) - \sigma(t)) - v(t)y_e(t)\delta(t), \quad (2.87)$$

where k_3 is a positive constant and

$$\delta(t) = \begin{cases} 1 & \text{if } \psi_e(t) = \sigma(t), \\ \frac{\sin \psi_e(t) - \sin \sigma(t)}{\psi_e(t) - \sigma(t)} & \text{otherwise.} \end{cases}$$

Define the error states to be

$$\zeta(t) \triangleq [x_e(t) \quad y_e(t) \quad \psi_e(t) - \sigma(t) \quad r(t) - \phi(t) \quad v(t) - v_d(t)]^\top \quad (2.88)$$

and consider the following control laws for $N(t)$, $F(t)$, and $\dot{s}(t)$:

$$N_{PF}(t) = \dot{\phi}(t) - k_4(r(t) - \phi(t)) - (\psi_e(t) - \sigma(t)), \quad (2.89a)$$

$$F_{PF}(t) = \dot{v}_d(t) - k_5(v(t) - v_d(t)), \quad (2.89b)$$

$$\dot{s}_{PF}(t) = v(t) \cos \psi_e(t) + k_1 x_e(t), \quad (2.89c)$$

where $k_1, k_4, k_5 > 0$. The two terms $\sigma(t)$ and $\phi(t)$ define desired profiles for $\psi_e(t)$ and $r(t)$, respectively, that is determined at the kinematic level [87]. Figure 2.12 shows the graphical representation of $\sigma(t)$ and $\delta(t)$. From Figure 2.12 (a), one can see that $\sigma(t)$ is a function of $y_e(t)$ that approaches zero at a rate that is dependent on the value of $y_e(t)$. If $\psi_e(t)$ tracks $\sigma(t)$, then $\psi_e(t)$ also approaches zero at the same rate. Furthermore, $\sigma(t)$ acts as a saturation function for $\psi_e(t)$ to ensure that the vehicle's approach angle is within the capability of the vehicle. In a similar way, the choices of $\delta(t)$ and $\sigma(t)$ combine to form a part of the shaping function $\phi(t)$, so that $r(t)$ respects the limits of the vehicle turn rates.

It follows from Ref. [28] that the combined kinematic and dynamic systems can be stabilized by the functions in (2.89) as stated in the following lemma.

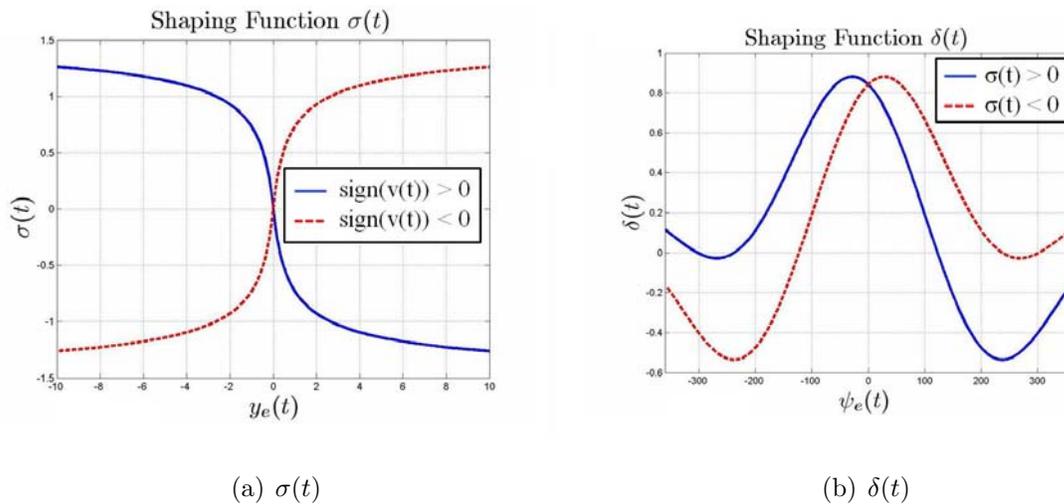


Figure 2.12: Graphical representation of shaping functions.

Remark 4 *The proof of Proposition is derived in [28] using Barbalat's lemma instead of the La-Salle Yoshizawa lemma. Applying the La-Salle Yoshizawa lemma simplifies the proof as compared to the proof in [28].*

Proposition 4 *If $v(t)$ is uniformly continuous and $\lim_{t \rightarrow \infty} v(t) \neq 0$, then the closed-loop system in (2.84), (2.85), and (2.89) is globally asymptotically stable [28].*

Proof. Consider the Lyapunov function candidate

$$V_{pf}(t) = \frac{1}{2} \zeta^\top(t) \zeta(t), \quad (2.90)$$

where the subscript $_{pf}$ denotes path following. Its time derivative along the trajectories

(2.84) and (2.85) is:

$$\begin{aligned}
\dot{V}_{pf} &= x_e((y_e c c - 1)\dot{s} + v \cos \psi_e) + y_e(-x_e c c \dot{s} + v \sin \psi_e) + (\psi_e - \sigma)(r - c c \dot{s} - \dot{\sigma}) + (r - \phi)(N - \dot{\phi}) \\
&\quad + (v - v_d)(F - \dot{v}_d) \\
&= -k_1 x_e^2 + y_e v \sin \psi_e + (\psi_e - \sigma)(r - c c \dot{s} - \dot{\sigma}) - k_4 (r - \phi)^2 - (r - \phi)(\psi_e - \sigma) - k_5 (v - v_d)^2 \\
&= -k_1 x_e^2 + y_e v \sin \psi_e - k_4 (r - \phi)^2 - v \sin \psi_e \nabla(\psi_e - \sigma) - k_3 (\psi_e - \sigma)^2 - k_5 (v - v_d)^2 \\
&= -k_1 x_e^2 - k_4 (r - \phi)^2 + v \sin \psi_e \sin \sigma - k_3 (\psi_e - \sigma)^2 - k_5 (v - v_d)^2 \\
&= -k_1 x_e^2 - k_4 (r - \phi)^2 - |v| \frac{k_2 y_e^2}{|y_e| + \epsilon_0} - k_3 (\psi_e - \sigma)^2 - k_5 (v - v_d)^2 \\
&\triangleq -W(\zeta(t)) \leq 0.
\end{aligned} \tag{2.91}$$

Thus, $\zeta = 0$ is a globally stable equilibrium point for the closed-loop system. Straightforward application of La-Salle Yoshizawa lemma further implies that $\lim_{t \rightarrow \infty} W(\zeta(t)) = 0$. Hence, $x_e(t)$, $|v(t)|y_e^2(t)$, $\psi_e(t) - \sigma(t)$, $r(t) - \phi(t)$, and $v(t) - v_d(t)$ must vanish as $t \rightarrow \infty$. Since $V_{pf}(t)$ is bounded below by zero and non-increasing, it follows that

$$\lim_{t \rightarrow \infty} y_e(t) = y_{\text{lim}},$$

where y_{lim} is a finite number. If $\lim_{t \rightarrow \infty} v(t) \neq 0$, then $\lim_{t \rightarrow \infty} |v(t)|y_e^2(t) = 0$ guarantees that $y_{\text{lim}} = 0$ so that $\zeta(t) = 0$ is globally asymptotically stable. This completes the proof.

□

2.5 Summary

This chapter reviews the path generation method using single polynomial functions in [97, 42, 43, 44, 29]. Special attention is given to the case when the virtual speed $\eta(s)$ is itself a polynomial function. An analysis of $\eta(s)$ and its order is provided, as well as its corresponding limits with increasing order in the context of the path generation algorithm. Furthermore, the constant $\eta(s)$ case is shown to correspond to $s = t$. A comparison is made between

the polynomial path generation method, and a generation method using Bezier curves. It is shown that for the special case when the Bezier curve is parameterized by $\underline{s} = s/s_f$, the two polynomials generate identical paths. As an illustrative example, simulation results are presented that compare the generated path with the optimal solution. The method is extended to include obstacle avoidance at a basic level by introducing additional terms in the cost function. Since preliminary data regarding obstacle location may be inaccurate, minimal effort is taken at this stage for obstacle avoidance. As vehicles obtain visual information that give a more accurate estimate of obstacle size and locations, more precautions can be taken to avoid obstacles. The 2-D equations of motion for a mobile robot are derived and a set of path following dynamics are computed. The proof of stability for the path following controller in [28] is simplified upon application of La-Salle Yoshizawa Lemma.

Chapter 3

Nonholonomic Path Deformation for Obstacle Avoidance

In the absence of obstacles, a vehicle will follow the desired path $p(t)$. In the presence of obstacles, a prescribed collision criterion is introduced which depends on information from the visual sensors processed through the vision-based estimation schemes in Chapter 4. The information required by the collision criterion is the range to an obstacle and the size of the obstacle. When the collision criterion is violated, the vehicle veers from its original path $p(t)$ and follows a new path until it is deemed safe for the vehicle to revert back to its original path.

It is assumed that each vehicle can only detect obstacles within a distance R_s to itself, where the exact value of R_s depends on the specifications of the onboard camera. The subscript $_s$ of R_s denotes “sensor range”. In this dissertation, the case of a single obstacle for each vehicle is considered. That is, each vehicle encounters *at most* one obstacle along its path. Adding additional obstacles is a topic of future research and requires extension of the methods described below. The unknown obstacle is assumed to be shaped like a circle with unknown radius L , which is upper bounded by $L \leq L_{\max}$, where L_{\max} is a conservative upper bound on the size of the obstacle. An additional “danger radius”, R_d , surrounds the

obstacle and must be avoided to ensure safety of the vehicle. The constant R_d is a number that is larger than L by a pre-specified amount. That is, $R_d = L + d$, where d is an *a priori* specification. It will be seen in the following formulation that the minimum distance between vehicles, Ξ as defined in the path generation of Chapter 2, will be at least two times greater than the conservative upper bound $L_{\max} + d$. Hence, vehicles will not “share” obstacles. That is, each vehicle will encounter a maximum of one obstacle, which is not an obstacle for another vehicle. Any two vehicles will be separated by a distance of greater than $2(L_{\max} + d)$ as guaranteed by the path generation method in Chapter 2. This ensures that no two vehicles will encounter the same obstacle. The next section is formulated for a single vehicle. Multiple vehicle deconfliction and the choice of Ξ will be addressed in Section 3.2. The indexing denoting vehicle number is dropped for simplicity.

Let (P_x, P_y) be the location of the obstacle in the inertial frame. Define the points

$$P_1(t) = -(P_x - x_I(t)) \sin \psi(t) + (P_y - y_I(t)) \cos \psi(t) + R_d(t), \quad (3.1a)$$

$$P_2(t) = -(P_x - x_I(t)) \sin \psi(t) + (P_y - y_I(t)) \cos \psi(t) - R_d(t), \quad (3.1b)$$

and let $z(t)$ denote the measured distance from the vehicle to the obstacle. The points $P_1(t) = (P_{1_x}(t), P_{1_y}(t))$ and $P_2(t) = (P_{2_x}(t), P_{2_y}(t))$ lie on a line that intersects the location of the obstacle (P_x, P_y) and is perpendicular to the vehicle’s velocity vector [82] (see Figure 3.1). The speed $v(t)$ in (2.84) is the magnitude of the vehicle’s velocity vector with the orientation defined by $\psi(t)$. Hence, $P_1(t)$ and $P_2(t)$ are the extremities of the projected edge of danger radius surrounding the obstacle in the vehicle’s body-fixed frame. The collision criterion is given as follows.

Collision Criterion:

If $z(t) \leq R_s$ and $\text{sign}(P_{1_y}(t)) \neq \text{sign}(P_{2_y}(t))$, then the collision criterion is violated. (3.2)

The insight behind the above collision criterion is that if a line extending from the vehicle’s

velocity vector intersects with any obstacle boundary, the vehicle is on collision-course with the obstacle and the path must be modified to avoid this collision (Figure 3.1). Once the collision criterion is violated at some time t_1 , the vehicle must be given a new path to avoid the detected obstacle.

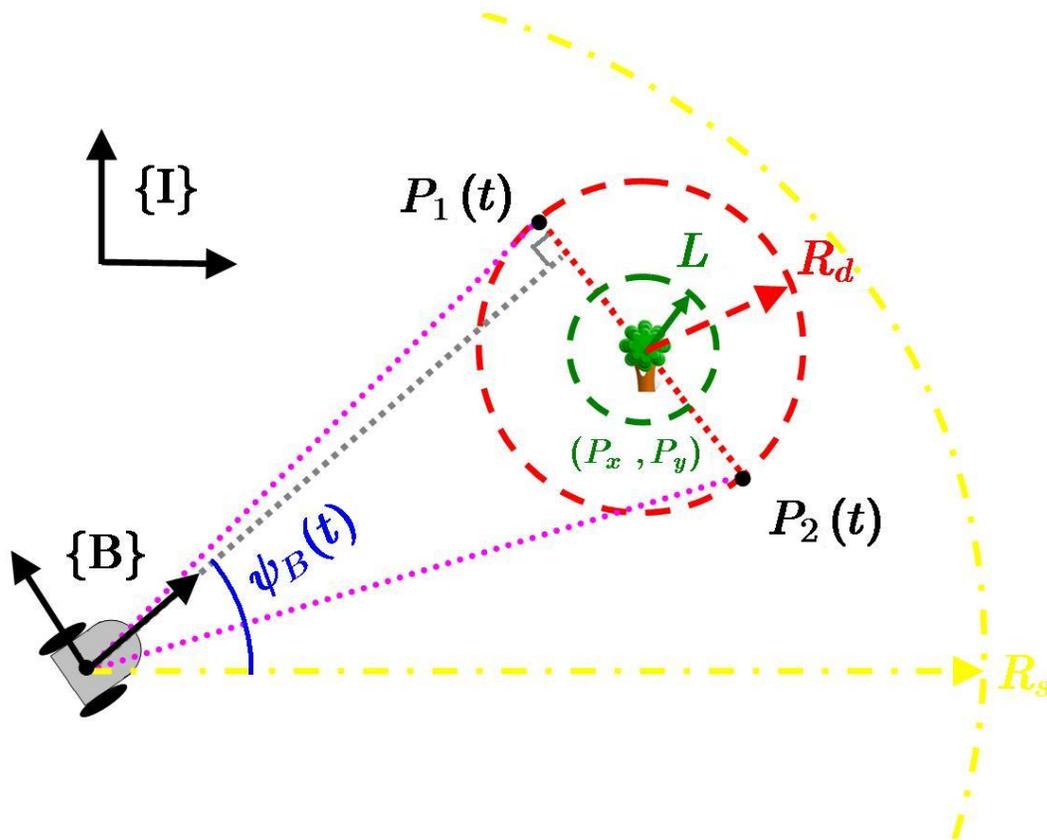


Figure 3.1: Points $P_1(t)$ and $P_2(t)$ are the projected edges of $R_d(t)$ expressed in the vehicle's body-fixed frame.

To compute a new path, we adopt a reactive OA method first appearing in Ref. [52] that deforms the original path on a kinematic level (i.e. with respect to the kinematics in (2.81)) in real-time in order to move away from the obstacle and ensures that the new path is collision-free. The path deformation algorithm is reviewed in this section. For more details,

please refer to Refs. [52, 57]. Since path deformation is not dependent on the evolution of a point along the path, we revert back to the notation $p(s)$ to denote the path in this section, where s is related to time via

$$\eta(s) \triangleq \frac{ds}{dt}.$$

Section 3.2 reformulates the path deformation algorithm for the mobile robot case with the kinematics in (2.84) first for a single vehicle. To account for a fleet of vehicles and to ensure that the obstacle avoidance does not cause any vehicles to collide, a deconfliction law which dictates the choice of the spatial separation variable Ξ in the path generation of Chapter 2 is introduced. Choosing a conservative value of Ξ guarantees that the vehicles will not collide for any number of vehicles avoiding a single obstacle.

3.1 Path Deformation

The main idea of the path deformation algorithm in Ref. [52] is to iteratively deform a path $p(s)$ by perturbing the kinematic inputs $u_j(s)$ introduced in (2.7) as the solution to the algebraic expression

$$\frac{d}{ds}(p(s)) = \sum_{j=1}^m u_j(s)G_j(p(s)), \quad \forall s \in \Omega,$$

where Ω is a compact set. To index these iterations, a parameter $\tau \in [0, +\infty)$ is introduced. The original path $p(s)$ then corresponds to the case when $\tau = 0$ and is rewritten as $p(s) = p(s, 0)$. The path deformation process is modelled as a mapping from a subset of $[0, S] \times [0, +\infty)$ to \mathbb{R}^n composed entirely of admissible paths. That is, there exist m -dimensional smooth mappings $u_1(s, \tau), u_2(s, \tau), \dots, u_m(s, \tau)$ defined over $[0, S] \times [0, +\infty)$ such that

$$\frac{\partial}{\partial s}(p(s, \tau)) = \frac{1}{\eta(s)} \sum_{j=1}^m u_j(s, \tau)G_j(p(s, \tau)), \quad \forall (s, \tau) \in [0, S] \times [0, +\infty). \quad (3.3)$$

From (3.3), one can see the reasoning behind having the dimension of the path $p(s)$ be equal to the number of states in light of Remark 3. Path deformation will be completed

by perturbing or altering the kinematic control inputs and every state must be included in order to fully define the deformation. Differentiating (3.3) with respect to τ , we get

$$\frac{\partial^2}{\partial s \partial \tau} (p(s, \tau)) = \frac{1}{\eta(s)} \sum_{j=1}^m \left(\frac{\partial u_j}{\partial \tau}(s, \tau) G_j(p(s, \tau)) + u_j(s, \tau) \frac{\partial G_j}{\partial p}(p(s, \tau)) \frac{\partial p}{\partial \tau}(s, \tau) \right), \quad (3.4)$$

where the second term on the right-hand side is derived upon application of the Chain Rule. The expression (3.4) explicitly defines a relationship between the input variations $\partial u_j / \partial \tau$, for $j = 1, \dots, m$, and the path variations $\partial p / \partial \tau$. Let $\nu(s, \tau) : [0, S] \times [0, +\infty) \mapsto \mathbb{R}^m$ and $\kappa(s, \tau) : [0, S] \times [0, +\infty) \mapsto \mathbb{R}^n$ denote the *input perturbations* and the *direction of deformation*, respectively, defined by

$$\nu(s, \tau) \triangleq \frac{\partial u}{\partial \tau}(s, \tau), \quad \kappa(s, \tau) \triangleq \frac{\partial p}{\partial \tau}(s, \tau), \quad (3.5)$$

where $u \triangleq [u_1 \ u_2 \ \dots \ u_m]^\top$. The concept of the direction of deformation for a single obstacle is illustrated in Figure 3.2. The direction of deformation must be chosen in a direction that moves the path away from the obstacle. Note that $\nu(s, \tau)$ belongs to the infinite-dimensional space of smooth vector-valued functions C^∞ . The path deformation is computed in a finite-dimensional space, which will be addressed in the following.

With the definitions in (3.5), equation (3.4) can be rewritten as:

$$\frac{\partial}{\partial s} (\kappa(s, \tau)) = A(s, \tau) \kappa(s, \tau) + B(s, \tau) \nu(s, \tau), \quad \kappa(0, \tau) = 0, \quad (3.6)$$

where

$$\begin{aligned} A(s, \tau) &= \frac{1}{\eta(s)} \sum_{j=1}^m u_j(s, \tau) \frac{\partial G_j}{\partial p}(p(s, \tau)) \in \mathbb{R}^{n \times n}, \\ B(s, \tau) &= \frac{1}{\eta(s)} [G_1(p(s, \tau)) \ G_2(p(s, \tau)) \ \dots \ G_m(p(s, \tau))] \in \mathbb{R}^{n \times m}. \end{aligned} \quad (3.7)$$

The direction of deformation $\kappa(s, \tau)$ is related to $\nu(s, \tau)$ through a linear dynamical system as seen in (3.6), where $\kappa(s, \tau)$ is the state of the system and $\nu(s, \tau)$ is the input to the system evolving with respect to τ . If $\nu(s, \tau)$ is specified, equation (3.6) can be integrated with respect to τ to get the direction of deformation $\kappa(s, \tau)$. In implementation, τ is discretized

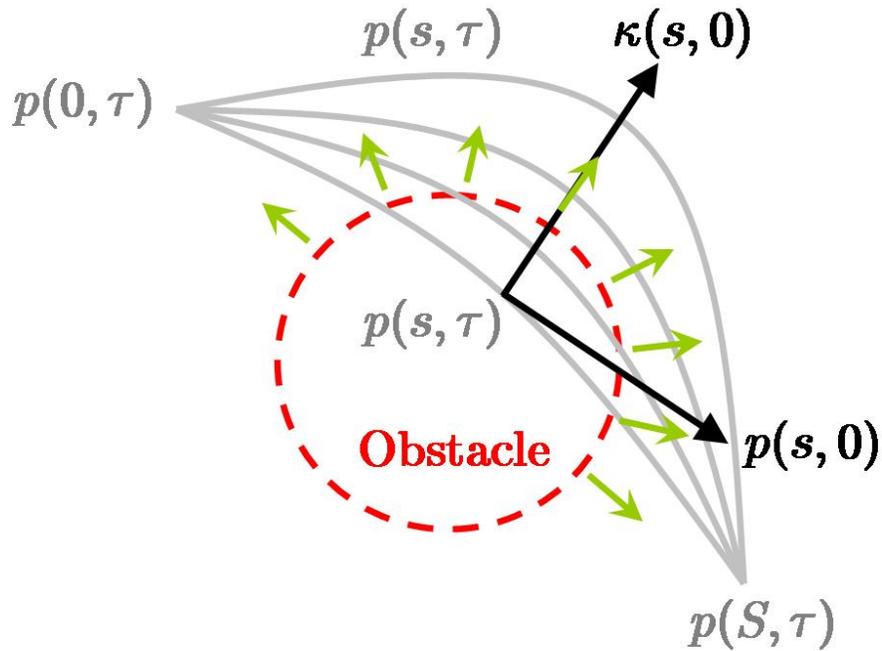


Figure 3.2: The iterative path deformation procedure. Obstacles impose repellant forces on the path so that the direction of deformation of the path ensures that an OA potential function $Y(\tau)$ diminishes.

for iterations of the path deformation process and new paths are numerically approximated according to the update law.

$$p(s, \tau + \Delta\tau) \approx p(s, \tau) + \Delta\tau(\tau)\kappa(s, \tau), \quad (3.8)$$

where $\Delta\tau(\tau)$ is the discretization step that depends on τ . Note that (3.8) is a first-order approximation with respect to τ . It follows that $\Delta\tau(\tau)\kappa(s, \tau)$ should be small in order to keep the approximation more accurate. To keep $\kappa(s, \tau)$ small, let κ_{\max} be a fixed small number. If $\Delta\tau(\tau)\|\kappa(s, \tau)\|_{\infty} \leq \kappa_{\max}$ is satisfied, where $\|\kappa(s, \tau)\|_{\infty} \triangleq \max_{s \in [0, S]} \|\kappa(s, \tau)\|$,

then the term $\Delta\tau(\tau)\kappa(s, \tau)$ remains small. The choice of

$$\Delta(\tau) = \begin{cases} \frac{\kappa_{\max}}{\|\kappa(s, \tau)\|_{\infty}}, & \text{if } \|\kappa(s, \tau)\|_{\infty} > \kappa_{\max}, \\ 1, & \text{otherwise,} \end{cases} \quad (3.9)$$

ensures that $\Delta\tau(\tau)\|\kappa(s, \tau)\|_{\infty} \leq \kappa_{\max}$ holds.

Remark 5 *It is difficult to pinpoint the exact relationship between the value of κ_{\max} to the performance of the path deformation method. Intuitively, smaller values of κ_{\max} requires a higher number of iterations to avoid obstacles, implying longer processing time and computational expense. Larger values of κ_{\max} may cause numerical instability and lead to diverging paths. In practice, the choice of κ_{\max} is determined through tuning.*

To define $\nu(s, \tau)$, it is essential to restrict $\nu(s, \tau)$ to a finite-dimensional subspace of C^{∞} generated by q linearly independent basis functions $\{b_1(s), b_2(s), \dots, b_q(s)\}$, where $q > n$ and $b_j(s) \in \mathbb{R}^m$ for $j = \{1, 2, \dots, q\}$, and express $\nu(s, \tau)$ with respect to these bases:

$$\nu(s, \tau) = \sum_{j=1}^q \lambda_j(\tau) b_j(s), \quad (3.10)$$

where the λ_j 's are parameters that uniquely define $\nu(s, \tau)$. The choice of $q > n$ is to ensure that the dimension of an $n \times q$ matrix, yet to be defined, will have a null space with $q - n > 0$ dimension. This will be made clear in the following. The basis functions can be, for example, finite Fourier series or finite power series. Let $E_j(s, \tau)$ denote the directions of deformation that correspond to the input perturbations given by $b_j(s)$ for the system (3.6) for $j \in \{1, 2, \dots, q\}$. That is, $E_j(s, \tau)$ are computed as solutions to the following q differential equations:

$$\frac{\partial}{\partial s} E_j(s, \tau) = A(s, \tau) E_j(s, \tau) + B(s, \tau) b_j(s), \quad E_j(0, \tau) = 0, \quad \text{for } j \in \{1, 2, \dots, q\}. \quad (3.11)$$

It follows from (3.6) that the direction of deformation $\kappa(s, \tau)$ corresponding to $\nu(s, \tau)$ can be decomposed into linear combinations of the E_j 's as

$$\kappa(s, \tau) = \sum_{j=1}^q \lambda_j(\tau) E_j(s, \tau). \quad (3.12)$$

The choice of appropriate parameters $\lambda_j(\tau)$ depends upon a local potential field $U(p(s, \tau))$ [83] and a global potential function that sums up this potential field along the path

$$Y(\tau) = \int_0^S U(p(s, \tau)) ds. \quad (3.13)$$

For the purpose of OA, $Y(\tau)$ should be chosen to yield large values for paths close to obstacles and smaller values away from obstacles. For example, $U(\tau)$ can be defined as the sum of the inverse distances to obstacles and S is the minimum value of the virtual arc of the desired path $p(s)$ when obstacles are no longer nearby [83]. The choice of $\lambda_j(\tau)$ should ensure that as τ increases, the value of the potential function $Y(\tau)$ decreases. The gradient $\frac{\partial U}{\partial p} p(s, \tau)$ in (3.16) acts as a force in \mathbb{R}^n that repels the path away from an obstacle (Figure 3.2). In order to define $\lambda_j(\tau)$ explicitly, consider the relationship between the variation of the potential function and $\kappa(s, \tau)$ as given by:

$$\frac{dY}{d\tau}(\tau) = \int_0^S \frac{\partial U^\top}{\partial p}(p(s, \tau)) \kappa(s, \tau) ds. \quad (3.14)$$

If the expression (3.14) is negative, then the potential function $Y(\tau)$ decreases with each iteration of τ . Hence, the proper choice of $\kappa(s, \tau)$ as discussed in Remark 5 ensures that (3.14) decreases with increasing τ and satisfies the linear dynamics (3.6). Substituting (3.12) into the expression (3.14) yields

$$\frac{dY}{d\tau}(\tau) = \sum_{j=1}^q \lambda_j(\tau) \int_0^S \frac{\partial U^\top}{\partial p}(p(s, \tau)) E_j(s, \tau) ds. \quad (3.15)$$

If $\lambda_j(\tau)$ is chosen such that (3.15) is negative, then the path potential function decreases with each iteration of τ .

Let

$$\mu_j(\tau) = \int_0^S \frac{\partial U^\top}{\partial p}(p(s, \tau)) E_j(s, \tau) ds,$$

where $U(p(s, \tau))$ is a function that depends on the path $p(s, \tau)$. The choice of

$$\lambda_j^0(\tau) = -\mu_j(\tau) \quad (3.16)$$

ensures that the path potential is at least non-increasing since (3.15) becomes:

$$\frac{dY}{d\tau}(\tau) = \sum_{j=1}^q - \left(\int_0^S \frac{\partial U^\top}{\partial p}(p(s, \tau)) E_j(s, \tau) ds \right)^2 = -\mu_j(\tau)^2 \leq 0.$$

Remark 6 *The choice of (3.16) may be slow to move the path away from an obstacle. To make the path potential decrease at the fastest rate, the optimal value of $\lambda_j(\tau)$ must realize the minimum:*

$$\min_{\|\kappa(s, \tau)\|_\infty=1} \left(\frac{dY}{d\tau}(\tau) \right) = \min_{\|\sum_{j=1}^q \lambda_j E_j\|_\infty=1} \left(\sum_{j=1}^q \mu_j(\tau) \lambda_j(\tau) \right). \quad (3.17)$$

The solution of (3.17) is difficult to obtain since minimization occurs over an infinity-norm $\|\cdot\|_\infty$. The authors in [52] suggest an approximation to the optimal value of $\lambda_j(\tau)$ can be made by minimizing instead

$$\min_{\|\sum_{j=1}^q \lambda_j E_j\|_{L^2}=1} \left(\sum_{j=1}^q \mu_j(\tau) \lambda_j(\tau) \right), \quad (3.18)$$

where $\|\cdot\|_{L^2}$ denotes the L^2 norm defined by

$$\|f\|_{L^2} \triangleq \left(\int_0^S f^\top(s) f(s) ds \right)^{\frac{1}{2}}.$$

The relationship between the approximation obtained by solving (3.18) and the solution of (3.17) is not defined in [52] and does not appear straightforward.

In order to impose continuity of the path, the deformed paths are restricted to have the same endpoints as the initial path $p(s, 0)$ which is enforced by the following boundary conditions:

$$p(0, \tau) = p(0, 0), \quad (3.19a)$$

$$p(S, \tau) = p(S, 0). \quad (3.19b)$$

This is equivalent to

$$\kappa(0, \tau) = 0, \quad (3.20a)$$

$$\kappa(S, \tau) = 0. \quad (3.20b)$$

Constraint (3.20a) is satisfied by definition of $\kappa(s, \tau)$ in equation (3.6). The second constraint (3.20b), along with (3.12), can be expressed as a linear constraint over $\lambda(\tau) = [\lambda_1(\tau) \ \lambda_2(\tau) \ \dots \ \lambda_q(\tau)]^\top$ as:

$$E(S, \tau)\lambda(\tau) = 0, \quad (3.21)$$

where $E(S, \tau) = [E_1(S, \tau) \ E_2(S, \tau) \ \dots \ E_q(S, \tau)]$. That is, the set of vectors $\lambda(\tau)$ that satisfies the boundary conditions (3.20b) must be in the null space of $E(S, \tau)$. Since $E(S, \tau)$ is full rank by construction, the dimension of its null space is $q - n$ by the Rank-Nullity Theorem, justifying the choice of $q > n$. To obtain such a vector for the choice of $\lambda_j^0(\tau)$ in (3.16), consider the orthogonal projection of $\lambda^0(\tau) = [\lambda_1^0(\tau) \ \lambda_2^0(\tau) \ \dots \ \lambda_q^0(\tau)]^\top$ over the null space of $E(S, \tau)$:

$$\lambda(\tau) = (\mathbb{I}_q - E(S, \tau)^\dagger E(S, \tau))\lambda^0(\tau), \quad (3.22)$$

where \mathbb{I}_q denotes the $q \times q$ identity matrix, and

$$E(S, \tau)^\dagger = \lim_{\epsilon \rightarrow 0} (E(S, \tau)^\top E(S, \tau) + \epsilon)^{-1} E(S, \tau)^\top$$

denotes the Moore-Penrose pseudo-inverse. Since $E(S, \tau)$ has linearly independent rows, the Moore-Penrose pseudo-inverse simplifies to

$$E^\dagger(S, \tau) = E^\top(S, \tau) (E(S, \tau)E^\top(S, \tau))^{-1}.$$

It is verified in Ref. [52] (Proposition 1, page 971) that the direction of deformation corresponding to the choice of $\lambda(\tau)$ ensures that the path potential $Y(\tau)$ decreases. Using the definition of $\lambda(\tau)$ in (3.22), $\kappa(s, \tau)$ can be computed from the relationship in (3.12).

Given a potential field $U(p(s, \tau))$ and an initial path $p(s, 0)$, the path deformation algorithm iterates with respect to τ until the potential function no longer decreases from further iterations and all collisions with obstacles have been avoided. A larger number of iterations implies that the OA will be more successful at diverting the vehicle from obstacles than fewer iterations. The number of iterations that can be performed in real-time is dependent on the

Table 3.1: Path deformation algorithm

1. Given a path $p(s, \tau)$, compute $u(s, \tau)$ via the relationship in (3.3) from iteration to iteration.
2. Compute matrices $A(s, \tau)$ and $B(s, \tau)$ as defined in (3.7), where the computed $u(s, \tau)$ from Step 1 is used.
3. Compute $E_j(s, \tau)$ by integrating (3.11).
4. Compute the gradient of $U(p(s, \tau))$ for the current path using the locations of obstacles.
5. Compute $\lambda_j^0(\tau)$ from the expression in (3.16).
6. Project $\lambda^0(\tau)$ over the null space of $E(S, \tau)$ as computed in equation (3.22). Note that this uniquely defines the input perturbations $\nu(s, \tau)$.
7. Compute $\kappa(s, \tau)$ using (3.12).
8. Compute the approximate updated path $p(s, \tau + \Delta\tau) \approx p(s, \tau) + \Delta\tau\kappa(s, \tau)$, where $\Delta\tau$ is the iteration step-size which evolves according to (3.9).
9. Check to see if the updated path $p(s, \tau + \Delta\tau)$ is equal to the current path $p(s, \tau)$.
 - If $p(s, \tau + \Delta\tau) \neq p(s, \tau)$, \rightarrow repeat steps 1 – 11 with $p(s, \tau) = p(s, \tau + \Delta\tau)$.
 - If $p(s, \tau + \Delta\tau) = p(s, \tau)$, \rightarrow stop.

onboard processor, which must be sufficiently fast. Integration over s is approximated using numerical approximation methods such as Euler integration or higher-order Runge-Kutta methods, for example. Table 3.1 outlines the path deformation algorithm.

In summary, the path deformation algorithm in Table 3.1 is a local optimization algorithm that can be viewed as a dynamical system, the state of which is a path. This system takes as input the kinematic model of the vehicle, an interval $[0, S]$ for which the OA is to take place, the original path $p(s) = p(s, 0)$ over this interval, and the obstacle location. The output of the system is a new admissible path $p(s, \tau)$. The control inputs for the path deformation algorithm are the input perturbations $\nu(s, \tau) = \frac{\partial u}{\partial p}(p(s, \tau))$ and a direction of deformation

$\kappa(s, \tau) = \frac{\partial p}{\partial \tau}(s, \tau)$. These control inputs are chosen based on the current path and the location of obstacles. Based on these control inputs, the τ -derivative of the path is uniquely defined and can be integrated with respect to τ to obtain a deformed path that meets the obstacle avoidance objective. Figure 3.3 shows a flow chart of the path deformation algorithm. In Section 3.2, we will apply this method towards a nonholonomic unicycle-type mobile robot with the kinematics in (2.84) for the purpose of OA.

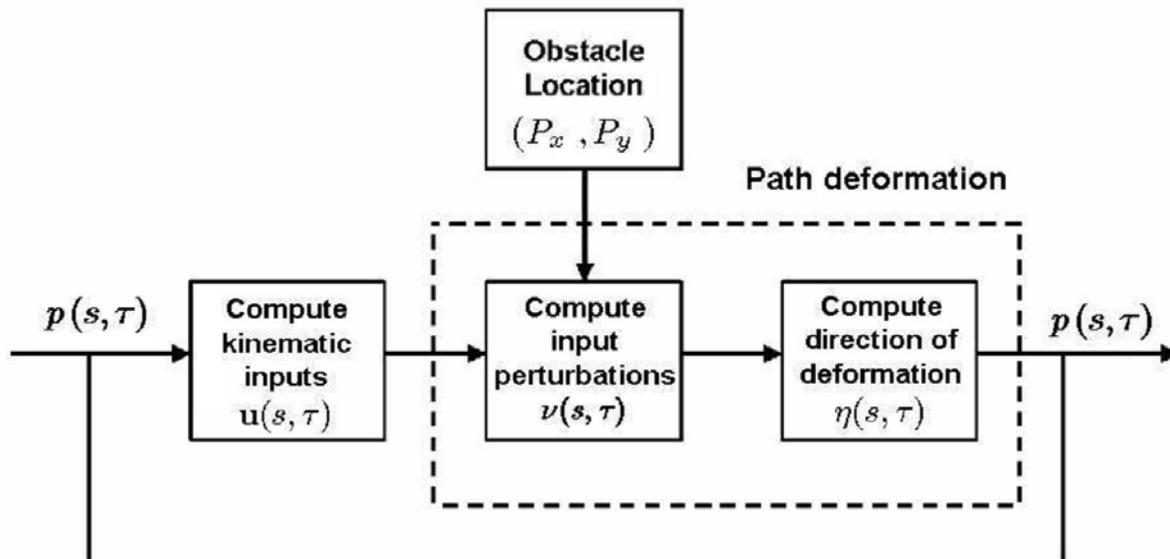


Figure 3.3: Path deformation process modelled as a dynamic control system evolving on a τ time scale.

3.2 Path Deformation for Two-Wheeled Mobile Robots in the Presence of Unknown Obstacles

To define the OA path, a portion of the original path $p(s)$ is deformed in the direction that minimizes a given potential function chosen to achieve the specific goal of obstacle avoidance.

One such choice of a potential function in (3.13) is

$$U(s, \tau) = \begin{cases} \frac{1}{\bar{z}(s, \tau) + d_0} + \frac{\bar{z}(s, \tau)}{(d_0 + \hat{R}_d(t_1))^2} & \text{if } 0 \leq \bar{z}(s, \tau) \leq \hat{R}_d(t_1), \\ \frac{1}{\hat{R}_d(t_1) + d_0} + \frac{\hat{R}_d(t_1)}{(d_0 + \hat{R}_d(t_1))^2} & \text{otherwise,} \end{cases} \quad (3.23a)$$

where t_1 is the time the collision criterion in (3.2) is violated, $0 < d_0 \ll 1$ is a constant, and

$$\bar{z}(s, \tau) = \sqrt{\left(p_x(s, \tau) - \hat{P}_x(t_1)\right)^2 + \left(p_y(s, \tau) - \hat{P}_y(t_1)\right)^2} \quad (3.24)$$

is the distance between the path and $(\hat{P}_x(t_1), \hat{P}_y(t_1))$, the estimated position of the obstacle (Figure 3.4). The parameters $\hat{R}_d(t)$, $\hat{P}_x(t_1)$, $\hat{P}_y(t_1)$ are used by the vehicle as the estimates of R_d , $P_x(t_1)$, and $P_y(t_1)$ which are related to the estimation of L , the derivation of which will be clarified in Chapter 4. Defining the potential function in this way yields high values when the vehicle is close to an obstacle and lower values when farther away. That is, a lower potential function value will result if the vehicle follows a path that stays farther away from an obstacle. Figure 3.4 shows $U(s, \tau)$ for an obstacle with $\hat{R}_d(t_1) = 3$ m and $d_0 = 0.001$. As $\bar{z}(s, \tau)$ gets closer to zero, $U(s, \tau)$ gets larger, up to a maximum value of $1/d_0$. As $\bar{z}(s, \tau)$ approaches and surpasses the value of $\hat{R}_d(t_1)$, the value of $U(s, \tau)$ becomes constant.

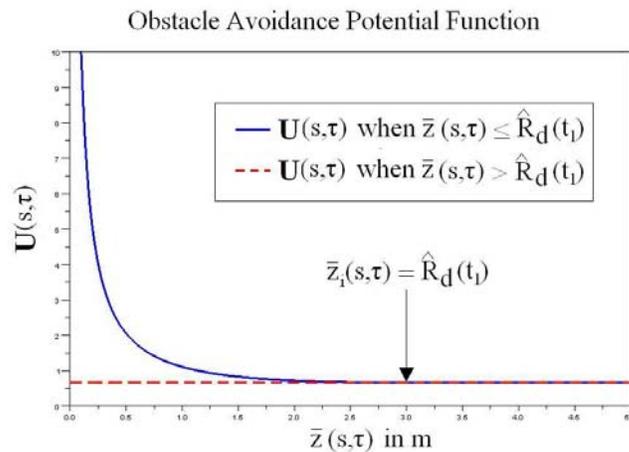


Figure 3.4: Plot of $U(s, \tau)$ for an obstacle with $\hat{R}_d(t_1) = 3$, $d_0 = 0.001$.

The gradient $\frac{\partial U}{\partial p}(s, \tau)$ is computed as

$$\begin{aligned} \frac{\partial U}{\partial p}(s, \tau) &= \left[\frac{\partial U}{\partial p_x}(s, \tau) \quad \frac{\partial U}{\partial p_y}(s, \tau) \quad \frac{\partial U}{\partial p_\psi} \right]^\top \\ &= \begin{cases} \begin{bmatrix} \frac{-(p_x(s, \tau) - \hat{P}_x(t_1))}{\bar{z}(s, \tau)(\bar{z}(s, \tau) + d_0)^2} + \frac{p_x(s, \tau) - \hat{P}_x(t_1)}{\bar{z}(s, \tau)(d_0 + \hat{R}_d(t_1))^2} \\ \frac{-(p_y(s, \tau) - \hat{P}_y(t_1))}{\bar{z}(s, \tau)(\bar{z}(s, \tau) + d_0)^2} + \frac{p_y(s, \tau) - \hat{P}_y(t_1)}{\bar{z}(s, \tau)(d_0 + \hat{R}_d(t_1))^2} \\ 0 \end{bmatrix}, & \text{if } \bar{z}(s, \tau) \leq \hat{R}_d(t_1), \\ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, & \text{otherwise.} \end{cases} \end{aligned} \quad (3.25)$$

Remark 7 Note that the definition of the potential field U in (3.23) implies that the path will deform until $\bar{z}(s, \tau) > \hat{R}_d(t_1)$ for all $s \geq 0$. Indeed, $\bar{z}(s, \tau) > \hat{R}_d(t_1)$ everywhere implies that $U(s, \tau) = U$ is a constant:

$$\frac{\partial U}{\partial p} = \mathbf{0}.$$

From (3.16), $\lambda^0(\tau) = 0$ for all future iterations. Hence, the direction of deformation $\kappa(s, \tau) = 0$ so that no further updates will be made on the path.

Remark 8 It can be seen from (3.23) the importance of the choice of R_d , which directly affects the value of $\hat{R}_d(t)$. The original undeformed path may be computed based on a set of optimality criteria and diverting away from this path decreases the optimality of the path. In this sense, smaller values of R_d restrict the path to deform minimally away from the original path and an obstacle, while larger values increase the distance between the vehicle and the obstacle. Hence, there exists a trade-off between increasing the safety margin for the vehicle and retaining the desired characteristics of the original path.

Using the potential field in kinematics in (3.13) and (3.23) and the kinematics in (2.81), we apply the path deformation process described in Ref. [52]. Let $S = s_f - s_0$, where $[s_0, s_f]$ is the interval over which the OA is to take place. Define s_0 as $s(t_1)$, where t_1 is the time the collision criterion is violated and $s(t)$ is the path length along $p(s)$ which evolves according to the dynamics $\dot{s}(t)$ as defined in (2.89c). Let $s_f = \hat{P}_x(t_1) + \hat{R}_d$ for a single obstacle. Note that multiple obstacles or clusters of obstacles are not considered by defining s_f in this way. Future work in this direction would be to define s_f in such a way as to incorporate a more general class of obstacles. In the implementation, the parameter τ is discretized as τ_k . One can compute

$$A(s, \tau_k) = \frac{1}{\eta(s)} \sum_{j=1}^2 u_j(s, \tau_k) \frac{\partial G_j}{\partial p}(p(s, \tau_k)) = \frac{1}{\eta(s)} \begin{bmatrix} 0 & 0 & -u_1(s, \tau_k) \sin(p_\psi(s, \tau_k)) \\ 0 & 0 & u_1(s, \tau_k) \cos(p_\psi(s, \tau_k)) \\ 0 & 0 & 0 \end{bmatrix}, \quad (3.26)$$

$$B(s, \tau_k) = \frac{1}{\eta(s)} [G_1(s, \tau_k) \quad G_2(s, \tau_k)],$$

where $G_1(s, \tau_k), G_2(s, \tau_k)$ are defined in (2.82). The basis functions chosen to restrict $\nu(s, \tau_k)$ to a finite-dimensional subspace of polynomial functions are:

$$\begin{aligned} b_1(s) &= [1 \quad 0]^\top, & b_2(s) &= [0 \quad 1]^\top, \\ b_3(s) &= \left[\frac{s}{S} \quad 0\right]^\top, & b_4(s) &= \left[0 \quad \frac{s}{S}\right]^\top, \\ b_5(s) &= \left[\frac{s^2}{S^2} \quad 0\right]^\top, & b_6(s) &= \left[0 \quad \frac{s^2}{S^2}\right]^\top, \\ &\vdots & &\vdots \\ b_{2q_1+1}(s) &= \left[\frac{s^{q_1}}{S^{q_1}} \quad 0\right]^\top, & b_{2q_1+2}(s) &= \left[0 \quad \frac{s^{q_1}}{S^{q_1}}\right]^\top, \end{aligned} \quad (3.27)$$

where q_1 is the maximal order of the polynomial basis and $q = 2q_1 + 2$.

It is important to show that the path deformation process, viewed as a dynamical system evolving on time-scale τ as shown in Figure (3.3), does not lead to paths that diverge. Towards that end, recall that

$$\bar{z}(s, \tau) = \sqrt{(p_x(s, \tau) - \hat{P}_x(t_1))^2 + (p_y(s, \tau) - \hat{P}_y(t_1))^2}$$

is the distance between the path and the estimated position of the obstacle. The parameters $\hat{P}_x(t_1), \hat{P}_y(t_1)$ are the estimates of P_x, P_y at a fixed time and are constant with respect to τ .

From the definition of $\kappa(s, \tau)$ in (3.12), the dynamics of $p(s, \tau)$ with respect to τ are

$$\frac{\partial p}{\partial \tau}(s, \tau) = \kappa(s, \tau) = \sum_{j=1}^q \lambda_j(\tau) E_j(s, \tau).$$

It can be computed that

$$\frac{\partial}{\partial \tau} (\bar{z}(s, \tau)) = \begin{bmatrix} \frac{p_x(s, \tau) - \hat{P}_x(t_1)}{\bar{z}(s, \tau)} & \frac{p_y(s, \tau) - \hat{P}_y(t_1)}{\bar{z}(s, \tau)} & 0 \end{bmatrix} \sum_{j=1}^q \lambda_j(\tau) E_j(s, \tau), \quad (3.28)$$

where $\lambda_j(\tau)$ are the components of $\lambda(\tau)$ as defined in (3.22). To keep the denominator bounded away from zero, let

$$\bar{Z}(s, \tau) = \sqrt{(p_x(s, \tau) - \hat{P}_x(t_1))^2 + (p_y(s, \tau) - \hat{P}_y(t_1))^2 + d_0},$$

where d_0 is the small positive constant introduced in (3.23). Boundedness of $\bar{Z}(s, \tau)$ is equivalent to boundedness of $\bar{z}(s, \tau)$ since d_0 is a constant. One can obtain that

$$\frac{\partial}{\partial \tau} (\bar{Z}(s, \tau)) = \begin{bmatrix} \frac{p_x(s, \tau) - \hat{P}_x(t_1)}{\bar{Z}(s, \tau)} & \frac{p_y(s, \tau) - \hat{P}_y(t_1)}{\bar{Z}(s, \tau)} & 0 \end{bmatrix} \sum_{j=1}^q \lambda_j(\tau) E_j(s, \tau). \quad (3.29)$$

For fixed s , this system is a nonautonomous system with independent variable τ and dependent variable \bar{Z} . If $\bar{Z}(s, \tau)$ is bounded, it follows that $\bar{z}(s, \tau)$ and the components of the deformed paths $p_x(s, \tau)$, $p_y(s, \tau)$, and hence, $p_\psi(s, \tau)$ are bounded. The proof of boundedness of $\bar{z}(s, \tau)$ which results in boundedness of $p_x(s, \tau)$, $p_y(s, \tau)$, and $p_\psi(s, \tau)$ is given next.

Using the definition of $\lambda(\tau)$ in (3.16) and (3.22), $\lambda_j(\tau)$ can be expressed as

$$\begin{aligned} \lambda_j(\tau) &= L_{j1}(\tau) \lambda_1^0(\tau) + L_{j2}(\tau) \lambda_2^0(\tau) + \dots + L_{jq}(\tau) \lambda_q^0(\tau) \\ &= -L_{j1}(\tau) \int_0^S \frac{\partial U^\top}{\partial p}(s, \tau) E_1(s, \tau) ds - \dots - L_{jq}(\tau) \int_0^S \frac{\partial U^\top}{\partial p}(s, \tau) E_q(s, \tau) ds \quad (3.30) \\ &= - \int_0^S \frac{\partial U^\top}{\partial p}(s, \tau) \left(L_{j1}(\tau) E_1(s, \tau) + \dots + L_{jq}(\tau) E_q(s, \tau) \right) ds, \end{aligned}$$

where $j = \{1, 2, \dots, q\}$ and $L_{jk}(\tau)$ is the j th row, k th column component of the matrix

$$(\mathbb{I}_q - E(S, \tau)^\dagger E(S, \tau)).$$

One can see from (3.30) that $\lambda_j(\tau)$ is equal to zero when the gradient $\frac{\partial U}{\partial p}(s, \tau)$ in (3.25) is identically equal to zero for all $s \in [0, S]$. That is,

$$\bar{Z}(s, \tau) \geq \sqrt{\hat{R}_d^2(t_1) + d_0}$$

or $\bar{z}(s, \tau) \geq \hat{R}_d(t_1)$ for all $s \in [0, S]$, where $\hat{R}_d(t_1)$ is the estimate of the danger radius R_d for the obstacle.

The following Proposition states general uniform boundedness results for a nonautonomous system which is similar to the result of Lemma 4.18 in [46] for uniform ultimate boundedness.

Proposition 5 *Let $D \subset \mathbb{R}^n$ be a domain that contains the origin and $V : [0, \infty) \times D \rightarrow \mathbb{R}$ be a continuously differentiable function such that*

$$\alpha_1(\|x\|) \leq V(t, x) \leq \alpha_2(\|x\|) \quad (3.31)$$

$$\frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} f(t, x) \leq 0, \quad \forall \|x\| \geq \mu > 0, \quad (3.32)$$

for all $t \geq 0$ and $x \in D$, where α_1, α_2 are class \mathcal{K} functions. Take $r > 0$ such that $B_r \triangleq \{x \in D \mid \|x\| \leq r\} \subset D$ and suppose that

$$\mu < \alpha_2^{-1}(\alpha_1(r)).$$

If every initial state $x(t_0)$ satisfies $\|x(t_0)\| \leq \alpha_2^{-1}(\alpha_1(r))$, then

$$\|x(t)\| \leq r, \quad \forall t \geq t_0.$$

Proof. Let $\Omega_1 = \{x \in B_r \mid V(t, x) \leq \alpha_1(r)\}$ and let $\Omega_2 = \{x \in B_r \mid V(t, x) \leq \alpha_2(\mu)\}$. Then $B_\mu \triangleq \{x \in B_r \mid \|x\| \leq \mu\} \subset \Omega_2$ and $\Omega_1 \subset B_r$. Furthermore, $\mu < \alpha_2^{-1}(\alpha_1(r))$ and making use of α_2 being a class \mathcal{K} function implies that $\alpha_2(\mu) \leq \alpha_1(r)$ and $\Omega_2 \subset \Omega_1$. If $\|x(t_0)\| \leq \alpha_2^{-1}(\alpha_1(r))$, then $\|x(t_0)\| \in \Omega_1$. Indeed, $\dot{V}(t, x)$ is non-positive on the boundary of Ω_1 and a solution in Ω_1 must remain in Ω_1 . Hence, $\|x(t)\| \in \Omega_1 \subset B_r$ and $\|x(t)\| \leq r$. It is important to note that if $\|x(t_0)\| \in \Omega_2$, then $\|x(t)\|$ cannot leave the set Ω_2 because

$\dot{V}(t, x)$ is non-positive on the boundary of Ω_2 . For this case, one can use the fact that $\Omega_2 \subset \{x \in B_r \mid \alpha_1(\|x\|) \leq \alpha_2(\mu)\}$ to obtain:

$$\|x(t)\| \leq \alpha_1^{-1}(\alpha_2(\mu)).$$

Figure 3.5 illustrates the proof. □

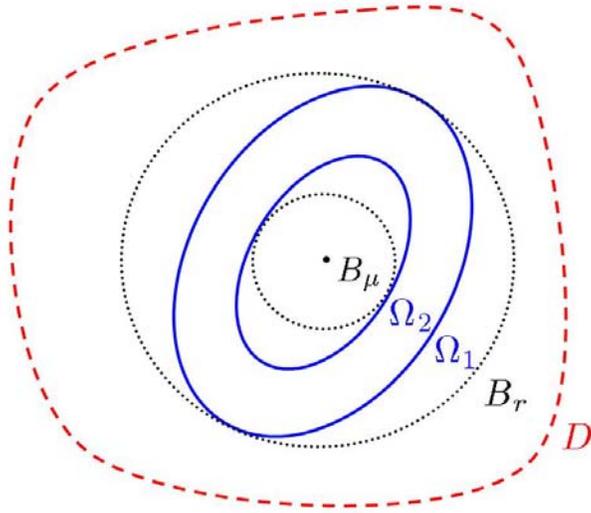


Figure 3.5: Representation of the sets B_r, B_μ (dotted) and Ω_1, Ω_2 (solid).

Finally, consider the following corollary for the dynamical system in (3.29) as a result of Proposition 5.

Corollary 2 *The dynamical system in (3.29), $p_x(s, \tau)$, $p_y(s, \tau)$, and $p_\psi(s, \tau)$ are uniformly bounded with respect to τ .*

Proof. Consider the candidate Lyapunov function

$$V(\bar{Z}(s, \tau)) = \frac{1}{2} \bar{Z}^2(s, \tau).$$

It can be computed from (3.30) that the τ - derivative of $V(\bar{Z}(s, \tau))$ with respect to the

trajectories (3.29) is

$$\begin{aligned} \frac{dV}{d\tau}(s, \tau) = & \begin{bmatrix} p_x(s, \tau) - \hat{P}_x(t_1) & p_y(s, \tau) - \hat{P}_y(t_1) & 0 \end{bmatrix} \\ & \sum_{j=1}^q \left(- \int_0^s \frac{\partial U^\top}{\partial p}(s, \tau) \left(L_{j1}(\tau) E_1(s, \tau) + \dots + L_{jq}(\tau) E_q(s, \tau) \right) ds \right) E_j(s, \tau). \end{aligned} \quad (3.33)$$

If $\|\bar{Z}(s, \tau)\| \geq \sqrt{\hat{R}_d^2(t_1) + d_0} > 0$, one can obtain

$$\frac{\partial U}{\partial p}(s, \tau) = 0$$

and

$$\frac{dV}{d\tau}(s, \tau) = 0.$$

Let $\mu = \sqrt{\hat{R}_d^2(t_1) + d_0}$ and $\alpha_1(\|\bar{Z}\|) = \alpha_2(\|\bar{Z}\|) = \|\bar{Z}\|^2$. Applying Proposition 5, one can conclude that the state $\bar{Z}(s, \tau)$ is uniformly bounded. Hence, $\bar{z}(s, \tau)$, $p_x(s, \tau)$, $p_y(s, \tau)$, and $p_\psi(s, \tau)$ are uniformly bounded with respect to τ . \square

Corollary 2 only implies boundedness of the deformed paths and does not guarantee that the path deformation will successfully divert the path from an obstacle. Intuitively, if the path deformation algorithm ensures that the origin is an unstable equilibrium of the system (3.28), which corresponds to $\bar{z}(s, \tau) = 0$ or collision with an obstacle, then the path will be guaranteed to avoid obstacles as long as $\bar{z}(s, 0) \neq 0$. That is, instability of the origin implies that the path is deformed away from obstacles (i.e. near the origin) until the path is outside the compact set $\bar{\Omega} = \{\bar{z} \mid \bar{z} \leq \bar{R}_d(t_1)\}$ as shown by Corollary 2. These observations motivate the following (unproven) conjecture.

Conjecture 1 *If $\bar{z}(s, 0) \neq 0$ and the origin $\bar{z}(s, \tau) = 0$ is an unstable equilibrium of the dynamical system (3.28) with respect to τ , then the path deformation method guarantees that the obstacle is avoided and $\bar{z}(s, \tau) \geq \hat{R}_d(t_1)$ after a sufficient number τ -iterations.*

The boundedness result in Corollary 2 and the proof of Conjecture 1 imply that the path deformation algorithm is well-defined by guaranteeing that the deformed path remains bounded and avoids an obstacle.

Note that boundedness results for $p(s, \tau)$ can be obtained for the path deformation dynamical system without using Lyapunov analysis. If the first component of the kinematic control inputs $u(s, \tau)$ is bounded, then uniform continuity results imply continuity of $p(s, \tau)$ as can be seen in the following alternative to Proposition 5.

Proposition 6 *If $u_1(s, \tau)$ is bounded for any $\tau \geq 0$, then $p(s, \tau)$ is bounded for the path deformation system which evolves according to the dynamics*

$$\frac{dp}{d\tau}(s, \tau) = \sum_{j=1}^q \left(- \int_0^s \frac{\partial U^\top}{\partial p}(s, \tau) \left(L_{j1}(\tau)E_1(s, \tau) + \dots + L_{jq}(\tau)E_q(s, \tau) \right) ds \right) E_j(s, \tau),$$

where $E_j(s, \tau)$ is defined in (3.11) for $j \in [0, q]$.

Proof. First, it is necessary to show that $E_j(s, \tau)$ is bounded for a fixed τ . From (3.11) and (3.26), one can see that

$$\frac{dE_j}{ds}(s, \tau) = \begin{bmatrix} 0 & 0 & -u_1(s, \tau) \sin p_\psi(s, \tau) \\ 0 & 0 & u_1(s, \tau) \cos p_\psi(s, \tau) \\ 0 & 0 & 0 \end{bmatrix} E_j(s, \tau) + \begin{bmatrix} \cos p_\psi(s, \tau) & 0 \\ \sin p_\psi(s, \tau) & 0 \\ 0 & 1 \end{bmatrix} b_j(s), \quad (3.34)$$

where $b_j(s)$ is the j th basis function introduced in (3.27). Let $E_{1j}(s, \tau), E_{2j}(s, \tau), E_{3j}(s, \tau)$ represent the three components of $E_j(s, \tau)$ and let $b_{1j}(s), b_{2j}(s)$ be the first and second components of $b_j(s)$. It follows that

$$\frac{dE_j}{ds}(s, \tau) = \begin{bmatrix} -u_1(s, \tau) \sin p_\psi(s, \tau) E_{3j}(s, \tau) + b_{1j} \cos p_\psi(s, \tau) \\ u_1(s, \tau) \cos p_\psi(s, \tau) E_{3j}(s, \tau) + b_{1j}(s) \sin p_\psi(s, \tau) \\ b_{2j}(s) \end{bmatrix}. \quad (3.35)$$

Using the definition of $b_j(s)$ in (3.27), one can compute that

$$E_{3j}(s, \tau) = \begin{cases} 0 & \text{if } j = 2r + 1, \\ \frac{s^{r+1}}{(r+1)S^r} & \text{if } j = 2r + 2, \end{cases} \quad \text{for } r = \{0, 1, \dots, q-1\}. \quad (3.36)$$

Hence, $E_{3_j}(s, \tau)$ is bounded. Using well known results [85], it can be concluded that $\frac{dE}{ds}(s, \tau)$ is bounded for all $j \in [0, q]$ and $E : [0, S] \rightarrow \mathbb{R}^3$ is uniformly continuous in s . Furthermore, uniform continuity of $E_j(s, \tau)$ implies that $E_j(s, \tau)$ is bounded over the compact set $[0, S]$ for all $j \in \{1, \dots, q\}$.

In order to show boundedness of $p(s, \tau)$ over each iteration of τ , let $H_j(s, \tau)$ be a continuous function whose s -derivative is

$$\frac{dH}{ds}(s, \tau) = \frac{\partial U^\top}{\partial p}(s, \tau) E_j(s, \tau).$$

Using the fact that $E_j(s, \tau)$ is bounded yields

$$\begin{aligned} \frac{\partial U^\top}{\partial p}(s, \tau) E_j(s, \tau) &\leq \left\| \frac{\partial U^\top}{\partial p}(s, \tau) \right\| \|E_j(s, \tau)\| \\ &= \left(\frac{1}{(\bar{z}(s, \tau) + d_0)^2} - \frac{1}{(\hat{R}_d(t_1) + d_0)^2} \right)^2 \|E_j(s, \tau)\|, \end{aligned} \quad (3.37)$$

which is bounded. For each τ , $H_j(s, \tau)$ is uniformly continuous and bounded over the compact set $[0, S]$. If $H_j(s, \tau)$ is bounded for all $s \in [0, S]$ and $j \in [0, q]$, then

$$H_j(S, \tau) - H_j(0, \tau) = \int_0^S \frac{\partial U^\top}{\partial p}(s, \tau) E_j(s, \tau) ds \quad (3.38)$$

is bounded. One can get from (3.38) that

$$\frac{dp}{ds}(s, \tau) = \sum_{j=1}^q \left(L_{j1}(\tau)(H_1(0, \tau) - H_1(S, \tau)) + \dots + L_{jq}(\tau)(H_q(0, \tau) - H_q(S, \tau)) \right) E_j(s, \tau)$$

is bounded. Finally, $p(s, \tau)$ is uniformly continuous with respect to τ and bounded for any τ belonging to a compact set. \square

Remark 9 *It is important to recognize that if (3.3) holds, then the path deformation method respects the nonholonomic constraints of the mobile robot (2.79). From (3.3) and (3.26), one*

can write the nonholonomic constraint (2.79) as

$$\begin{aligned}
& \dot{p}_y(s, \tau) \cos p_\psi(s, \tau) - \dot{p}_x(s, \tau) \sin p_\psi(s, \tau) \\
&= \eta(s) \left(\frac{dp_y}{ds}(s, \tau) \cos p_\psi(s, \tau) - \frac{dp_x}{ds}(s, \tau) \sin p_\psi(s, \tau) \right) \\
&= \eta(s) \left(u_1(s, \tau) \sin p_\psi(s, \tau) \cos p_\psi(s, \tau) - u_1(s, \tau) \sin p_\psi(s, \tau) \cos p_\psi(s, \tau) \right) \\
&= 0.
\end{aligned} \tag{3.39}$$

Judicious choice of κ_{\max} used in the first-order approximation (3.8) of τ ensures that the relationship in (3.3) holds with small error.

Next, the issue of multiple vehicles and their deconfliction with respect to the obstacle avoidance is discussed. In a multiple vehicle setting, only the case where each vehicle encounters at most one obstacle is being considered. It follows that the problem of deconfliction can be reduced to that of deconflicting two vehicles. The following choice of Ξ solves the obstacle avoidance deconfliction, which is illustrated in Figure 3.6.

Proposition 7 *The choice of*

$$\Xi = 2(L_{\max} + d) + 2\max(\kappa_{\max}, 1) + \epsilon_d, \tag{3.40}$$

where L_{\max} is a conservative upper bound on the size of obstacles, d is a positive constant, κ_{\max} defines the maximum step-size allowance defined in (3.9), and ϵ_d is an arbitrary positive constant, ensures that all vehicles are deconflicted during the path deformation process for their respective single obstacles.

Proof. Let vehicle i and vehicle j be any two arbitrary vehicles which are separated by the distance Ξ as chosen in (3.40). Since each vehicle can encounter at most one obstacle and the size of each obstacle is upper bounded by L_{\max} , it follows that the path deformation method will bring each vehicle at most $\Xi - 2(L_{\max} + d) + e_{PD}$ within each other, where e_{PD} is a path deformation error term by using the first-order approximation (3.8). This error term

is determined by the step size of the path deformation algorithm, which is given in (3.9) as $\max(\kappa_{\max}, 1)$. Hence, the minimum distance between the two vehicles becomes

$$\Xi - 2(L_{\max} + d) - \max(\kappa_{\max}, 1) = \epsilon_d.$$

This scenario is shown in Figure 3.6. In the case of single obstacles for each vehicle, guaranteeing two arbitrary vehicles do not collide ensures that all vehicles do not collide. \square

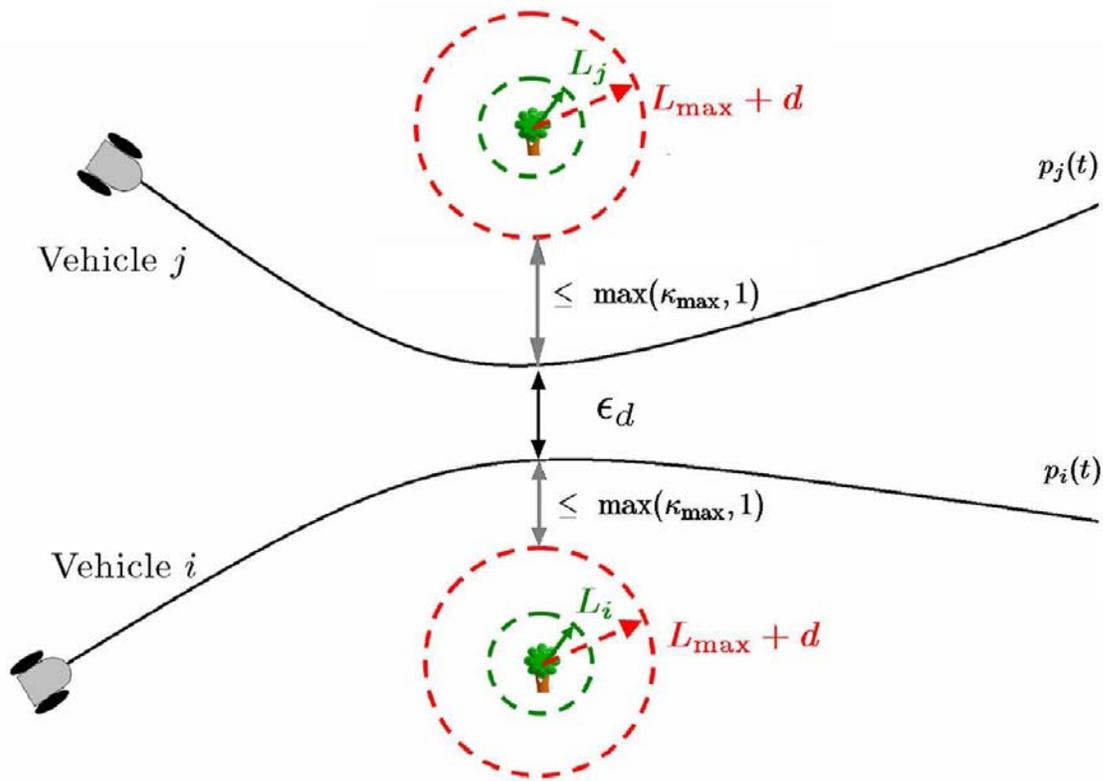


Figure 3.6: Vehicle i and vehicle j can move to within at most ϵ_d of each other by the choice of Ξ .

3.3 Path Deformation With Input Constraints Using Projection

The method in [52] does not incorporate saturation limits on vehicle speed and acceleration in the path deformation method. Since the kinematic control inputs $u_1(s, \tau), u_2(s, \tau)$ correspond to the speed and rate of rotation of the vehicle, these two quantities cannot exceed upper limits as imposed by the capabilities of the vehicle during path deformation. As an extension of the method in [52], the projection operator [78] is applied to the path deformation algorithm in order to respect input constraints and is used to keep the kinematic control inputs bounded. In [78], the projection operator is introduced in the context of adaptive control as a method to keep the adaptive parameter θ within a desired compact set to prevent parameter drift. That is, [78] considers an adaptive controller for a nonlinear system that depends linearly on some unknown parameter Θ . This unknown parameter is known to lie within a given set and the projection operator is used to ensure that the adaptive estimate θ of Θ stay in this set. While there is no need for adaptation in the context of path deformation, the projection operator can be used to keep the kinematic control inputs away from strict bounds in a continuous way as compared to other types of saturation functions. One contribution of this dissertation is to apply the projection operator towards keeping kinematic inputs bounded. With respect to the projection operator, recall the following definitions.

Definition 1 A set $\Omega_c \subset \mathbb{R}^m$ is a convex set if

$$cx + (1 - c)y \in \Omega_c, \quad \forall 0 \leq c \leq 1, \quad \forall x, y \in \Omega_c. \quad (3.41)$$

Definition 2 A function $f : \mathbb{R}^m \rightarrow \mathbb{R}$ is a convex function if

$$f(cx + (1 - c)y) \leq cf(x) + (1 - c)f(y), \quad \forall 0 \leq c \leq 1, \quad \forall x, y \in \mathbb{R}^m. \quad (3.42)$$

Intuitively, a convex set is a set such that every point on the straight line segment that joins any two points in the set is itself a member of the set. For example, a solid cube is convex,

but anything that is hollow or has a dent in it, for example, a crescent shape, is not convex. Similarly, a function is convex if the function lies below the straight line segment connecting any two points of the function on a given interval (Figure 3.7).

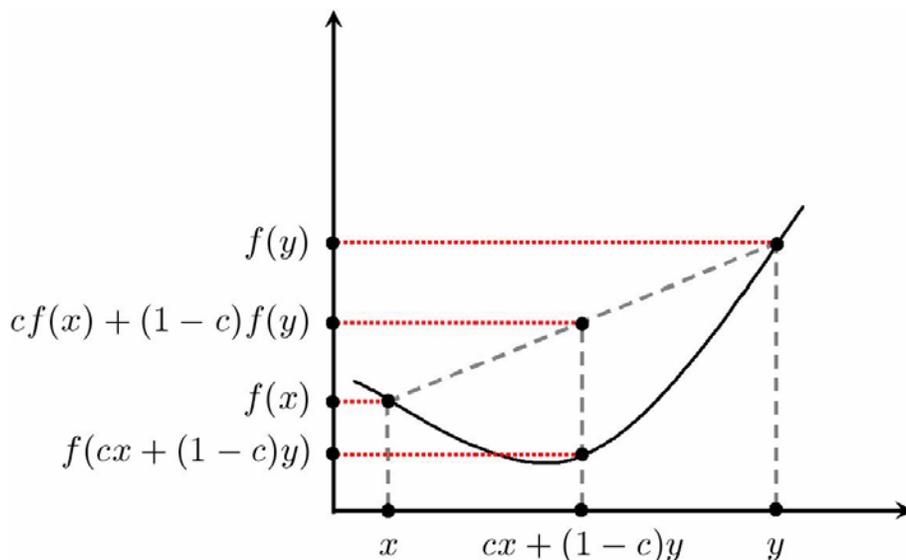


Figure 3.7: Pictorial representation of a convex function (solid line).

The projection operator exploits characteristics of convex sets and functions in order to keep a signal from leaving a pre-specified compact set. In the following, the general idea of using projection is first introduced in the context of [78]. The special case of applying projection to the problem of path deformation for a nonholonomic robot is given in the sequel. With that respect, the following Lemmas [78] will be useful in the construction of the projection operator and will be relied on to guarantee that the input constraints are met. The proofs of the Lemmas can be found in [78].

Lemma 1 *Let $f : \mathbb{R}^m \rightarrow \mathbb{R}$ be a convex function. For any constant $\delta > 0$, the set $\Omega_\delta = \{\theta \in \mathbb{R}^m | f(\theta) \leq \delta\}$ is convex.*

Lemma 2 *Let $f(\theta) : \mathbb{R}^m \rightarrow \mathbb{R}$ be a continuously differentiable convex function. Let $\delta > 0$ be a fixed constant and let $\Omega_\delta = \{\theta \in \mathbb{R}^m | f(\theta) \leq \delta\} \subset \mathbb{R}^m$ be a convex set. Furthermore, let*

$\theta, \theta^* \in \Omega_\delta$ with $f(\theta^*) < \delta$ and $f(\theta) = \delta$. That is, θ is a point that lies on the boundary of Ω_δ and θ^* is a strictly interior point of Ω_δ . The following inequality holds:

$$(\theta^* - \theta)^\top \nabla f(\theta) \leq 0, \quad (3.43)$$

where

$$\nabla f(\theta) \triangleq \left[\frac{\partial f(\theta)}{\partial \theta_1} \quad \frac{\partial f(\theta)}{\partial \theta_2} \quad \dots \quad \frac{\partial f(\theta)}{\partial \theta_m} \right]^\top \in \mathbb{R}^m$$

is the gradient of $f(\theta)$ evaluated at θ .

The importance of Lemma 2 is that it introduces a vector which always points out of the convex set, the negative of which is a vector that points inward towards the set. The gradient of a convex function evaluated at the boundary of the convex set is normal to the tangent of the convex set at the boundary point and makes an angle greater than 90° with the difference vector between the boundary point and any interior point of the convex set. Hence, their inner product is always negative. Figure 3.8 illustrates the main idea of Lemma 2. The projection operator exploits this idea by subtracting a vector that is parallel to the gradient of a convex function evaluated at the boundary of a convex set when a signal is wandering out of this set, smoothly transforming the original vector to an inward or tangent vector to keep the vector within the desired set. Next, the projection operator is formally defined in the context of [78] where the objective is to keep θ , which is the adaptive estimate of the unknown parameter Θ , in a given set.

Definition 3 Let $\Omega_\delta = \{\theta \in \mathbb{R}^m | f(\theta) \leq \delta\}$, $0 \leq \delta \leq 1$, where $f : \mathbb{R}^m \rightarrow \mathbb{R}$ is the smooth convex function

$$f(\theta) = \frac{\theta^\top \theta - \theta_{\max}^2}{\epsilon_\theta \theta_{\max}^2}, \quad (3.44)$$

where θ_{\max} is the norm bound imposed on the vector θ . The parameter ϵ_θ is a parameter that dictates the maximum tolerance that the norm of θ is allowed to exceed the maximum conservative value θ_{\max} . Let the true value of the adaptive parameter Θ be denoted as θ^* ,

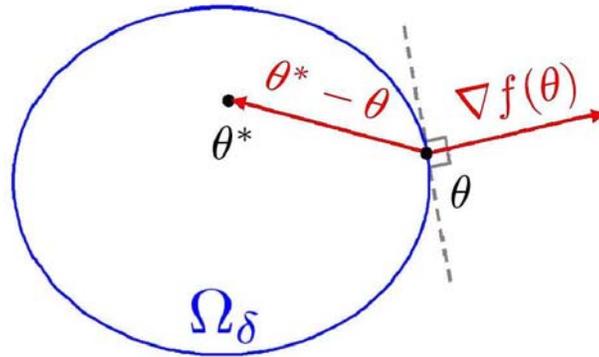


Figure 3.8: Pictorial representation of equation (8).

which belongs to $\Omega_0 = \{\Theta \in \mathbb{R}^m | f(\Theta) \leq 0\}$. The projection operator is

$$\text{Proj}(\theta, Z) \triangleq \begin{cases} Z & \text{if } f(\theta) < 0, \\ Z & \text{if } f(\theta) \geq 0 \text{ and } \nabla f^\top Z \leq 0, \\ Z - \frac{\nabla f}{\|\nabla f\|} \left\langle \frac{\nabla f^\top}{\|\nabla f\|}, Z \right\rangle f(\theta) & \text{if } f(\theta) \geq 0 \text{ and } \nabla f^\top Z > 0. \end{cases} \quad (3.45)$$

From the definition of the projection operator (3.45), Z is not altered if $\theta \in \Omega_0$ or if $0 < f(\theta) \leq 1$ and $\nabla f^\top Z \leq 0$. That is, if $\nabla f^\top Z \leq 0$, then Z is in a direction that points towards the interior of $\Omega_\delta = \{\theta \in \mathbb{R}^m | f(\theta) = \delta\}$ for $0 < \delta \leq 1$. If $0 < f(\theta) \leq 1$ and $\nabla f^\top Z > 0$, the projection operator subtracts a vector normal to Ω_δ to smoothly deflect Z in an Ω_δ -inward direction. Indeed, $\frac{\nabla f}{\|\nabla f\|}$ is the unit normal to Ω_δ and $\left\langle \frac{\nabla f^\top}{\|\nabla f\|}, Z \right\rangle f(\theta)$ is the projection of Z onto this unit normal scaled by $f(\theta)$ (Figure 3.9). Hence, if $\dot{\theta}(t) = \text{Proj}(\theta(t), y(t))$, then the adaptive parameter θ remains within the set $\Omega_1 = \{\theta \in \mathbb{R}^m | f(\theta) = 1\}$.

Note that $f(\theta) \leq 1$ implies that

$$\theta^\top \theta \leq (1 + \epsilon_\theta) \theta_{\max}^2.$$

It follows that the parameter ϵ_θ specifies the maximum tolerance the adaptive parameter is allowed to exceed θ_{\max} . The smaller the value of ϵ_θ , the more strict the projection operator will be in terms of keeping $\theta^\top \theta \leq \theta_{\max}^2$.

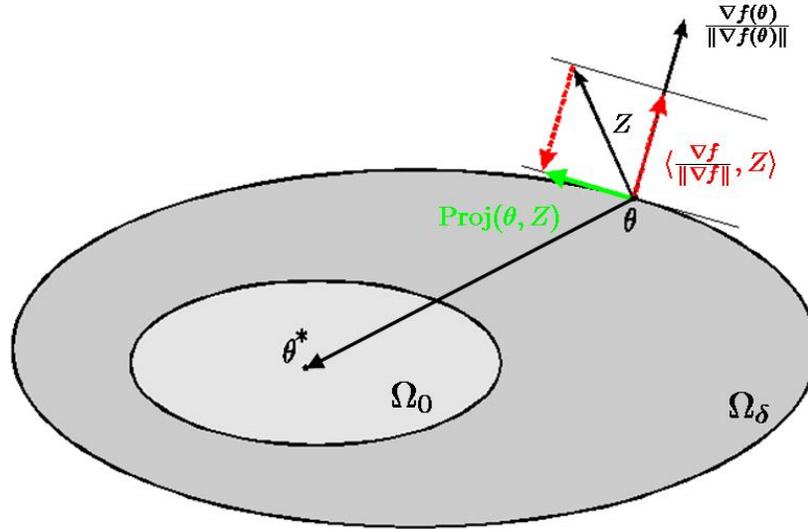


Figure 3.9: Pictorial representation of the projection operator.

Next, the projection operator is used to keep the kinematic inputs bounded in the path deformation process. In order to implement the projection operator, simple Euler integration is performed to approximate $u_j(s, \tau)$ at each s instead of solving for $u_j(s, \tau)$ as in Step 1 in Table 3.1. That is, Step 1 will be replaced by an approximation on $u(s, \tau)$ that incorporates projection. The smaller the step size of s in the discretization, the closer the solution obtained from Euler integration is to the solution obtained by implementing Step 1. As with the τ iterations, the size of each s -step is dictated by the capabilities of the available hardware.

Differentiating (3.3) with respect to s , one obtains

$$\frac{\partial^2}{\partial s^2} (p(s, \tau)) = \sum_{j=1}^m \frac{\partial u_j}{\partial s}(s, \tau) G_j(p(s, \tau)) + \sum_{j=1}^m u_j(s, \tau) \frac{\partial G_j}{\partial s}(p(s, \tau)), \quad (3.46)$$

which can be solved for $\frac{\partial u_j}{\partial s}(s, \tau)$. The solution that solves (3.46) is denoted as $\dot{u}_I(s, \tau)$ so that

$$\dot{u}_I(s, \tau) = \frac{\partial u_j}{\partial s}(s, \tau).$$

For a fixed τ , $u_j(s, \tau)$ evolves according to

$$u_j(s + \Delta s, \tau) = u_j(s, \tau) + \dot{u}_{I_j}(s, \tau)\Delta s + H.O.T., \quad (3.47)$$

where Δs is the s -step size, $\dot{u}_{I_j}(s, \tau)$ is the j th component of the solution $u_I(s, \tau)$ obtained by solving (3.46) for $u(s, \tau)$, and H.O.T. are higher-order terms resulting from the Taylor Series expansion of $u(s, \tau)$ computed from (3.3). These higher-order terms are small for small Δs and can be neglected so that $u_j(s, \tau)$ is approximated by the first-order approximation

$$u_j(s + \Delta s, \tau) = u_j(s, \tau) + \dot{u}_{I_j}(s, \tau)\Delta s. \quad (3.48)$$

The Euler integration is initiated by solving for $u_j(0, \tau)$ from the algebraic expression

$$\frac{\partial}{\partial s}(p(0, \tau)) = \sum_{j=1}^m u_j(0, \tau)G_j(p(0, \tau)). \quad (3.49)$$

Note that (3.48) is an approximation of the solution $u(s, \tau)$ obtained by solving (3.3) corresponding to Step 1 in Table 3.1. The idea of using the projection operator to keep the kinematic inputs $u(s, \tau)$ bounded is to replace the exact solution $\dot{u}_I(s, \tau)$ of (3.46) with a new derivative term $\dot{u}(s, \tau)$ that incorporates projection. Since $\dot{u}_I(s, \tau)$ corresponds to the derivative of the solution $u(s, \tau)$ as computed from (3.3), replacing $\dot{u}_I(s, \tau)$ with $\dot{u}(s, \tau)$ as defined by projection will further reduce the accuracy of the approximation (3.49) and decrease optimality of the path deformation method. Thus, there exists a trade-off by implementing projection and constraining input bounds with the optimality of the deformation algorithm.

Let u_{\max} denote the maximum bound on the norm of the kinematic control inputs $u(s, \tau)$. That is, for a fixed τ and for every $s \in [0, S]$, u must satisfy

$$\|u^\top(s, \tau)\| \leq u_{\max}.$$

The projection operator that keeps the kinematic control inputs bounded is defined formally.

Definition 4 Let $\bar{\Omega}_{\bar{\delta}} = \{u \in \mathbb{R}^m | \bar{f}(u) \leq \bar{\delta}\}$, $0 \leq \bar{\delta} \leq 1$, where $\bar{f} : \mathbb{R}^m \rightarrow \mathbb{R}$ is the smooth convex function

$$\bar{f}(u) = \frac{u^\top u - \bar{u}^2}{\epsilon_u \bar{u}^2}, \quad (3.50)$$

where $0 < \bar{u} < u_{\max}$ is a constant and

$$\epsilon_u = \frac{u_{\max}^2}{\bar{u}^2} - 1 > 0. \quad (3.51)$$

The projection operator that keeps $\|u\| \leq u_{\max}$ is

$$\text{Proj}(u, \dot{u}_I) \triangleq \begin{cases} \dot{u}_I & \text{if } \bar{f}(u) < 0, \\ \dot{u}_I & \text{if } \bar{f}(u) \geq 0 \text{ and } \nabla \bar{f}^\top \dot{u}_I \leq 0, \\ \dot{u}_I - \frac{\nabla \bar{f}}{\|\nabla \bar{f}\|} \left\langle \frac{\nabla \bar{f}}{\|\nabla \bar{f}\|}, \dot{u}_I \right\rangle \bar{f}(u) & \text{if } \bar{f}(u) \geq 0 \text{ and } \nabla \bar{f}^\top \dot{u}_I > 0. \end{cases} \quad (3.52)$$

Note from the definition of ϵ_u in (3.51) that if u remains in the set $\bar{\Omega}_1 = \{u \in \mathbb{R}^m \mid \bar{f}(u) \leq 1\}$, then

$$u^\top u \leq (1 + \epsilon_u) \bar{u}^2 = u_{\max}^2.$$

The parameter \bar{u} acts as a conservative upper bound on $u(s, \tau)$ and the set $\bar{\Omega}_0 = \{u \in \mathbb{R}^m \mid \bar{f}(u) \leq 0\}$ is the target set that $u(s, \tau)$ should remain close to. The closer \bar{u} is to u_{\max} , the smaller ϵ_u is and the stricter the projection operator is in terms of keeping $\|u\| \leq \bar{u}$. Indeed, a small ϵ_u implies that the maximum tolerance the projection operator will allow is small. Similarly, if \bar{u} is chosen to be small as compared to u_{\max} , then ϵ_u will be larger, allowing the kinematic control inputs to wander farther away from $\bar{\Omega}_0$. The following Proposition holds.

Proposition 8 *Let*

$$\dot{u}(s, \tau) = \text{Proj}(u(s, \tau), \dot{u}_I(s, \tau)), \quad (3.53)$$

where $\text{Proj}(u(s, \tau), \dot{u}_I(s, \tau))$ is defined in (3.52). Then, $u(s, \tau)$ never leaves $\bar{\Omega}_1$ and $\|u(s, \tau)\| \leq u_{\max}$ for all $s \in [0, S]$.

Proof. Let

$$u^* = \frac{\bar{u}}{\sqrt{m}} \mathbf{1}_m,$$

where m is the dimension of u and 1_m is the $m \times 1$ vector with every element equal to one. Note that u^* belongs to the set $\bar{\Omega}_0$ and $\|u^*\| < u_{\max}$. Consequently, u^* is a strictly interior point of $\bar{\Omega}_1$. From the definition of the projection operator (3.52), one can compute

$$(u^* - u)^\top (\dot{u}_I - \text{Proj}(u, \dot{u}_I)) = \begin{cases} 0 & \text{if } \bar{f}(u) < 0, \\ 0 & \text{if } \bar{f}(u) \geq 0 \text{ and } \nabla \bar{f}^\top \dot{u}_I \leq 0, \\ (u^* - u)^\top \frac{\nabla \bar{f}}{\|\nabla \bar{f}\|} \nabla \bar{f} \dot{u}_I \bar{f}(u) & \text{if } \bar{f}(u) \geq 0 \text{ and } \nabla \bar{f}^\top \dot{u}_I > 0. \end{cases}$$

From Lemma 2,

$$(u^* - u)^\top \frac{\nabla \bar{f}}{\|\nabla \bar{f}\|} \leq 0,$$

and $\nabla \bar{f}^\top \dot{u}_I > 0$, and $\bar{f}(u) > 0$ hold by definition. Hence, the term

$$(u^* - u)^\top \frac{\nabla \bar{f}}{\|\nabla \bar{f}\|} \nabla \bar{f} \dot{u}_I \bar{f}(u) \leq 0.$$

The following holds:

$$(u - u^*)^\top (\dot{u}_I - \text{Proj}(u, \dot{u}_I)) \geq 0, \quad \forall s \in [0, S].$$

This implies that \dot{u} is always pointed in a direction which is tangent to the boundary of $\bar{\Omega}_0$ or inward to $\bar{\Omega}_0$. Since $\bar{\Omega}_0 \subset \bar{\Omega}_1$, the signal u cannot leave $\bar{\Omega}_1$, which completes the proof. \square

Note that Proposition 8 implies that the kinematic control inputs remains bounded so that the assumption of Proposition 6 is satisfied. Furthermore, the choice of $\lambda(\tau)$ in (3.16) and (3.22) implies that $\lambda(\tau)$ is a function of $E_j(s, \tau)$, $E_j(S, \tau)$, $P(S, \tau)$ and $\partial U / \partial p(s, \tau)$ which makes the path potential decrease for any values of its arguments. Hence, the potential function $Y(\tau)$ is still guaranteed to decrease with increasing τ in the presence of the projection operator. The path deformation process is modified to include the use of the projection operator to keep the kinematic control inputs $u(s, \tau)$ bounded from iteration to iteration. The algorithm is initialized by the initial undeformed path $p(s, 0)$, the location of the obstacle (P_x, P_y) , and the corresponding kinematic control inputs $u_j(s, 0)$ that generate $p(s, 0)$ as obtained from the relationship in (3.49). To incorporate the use of the projection operator, Steps 2 - 11 in Table 3.1 remain unaltered but Step 1 becomes:

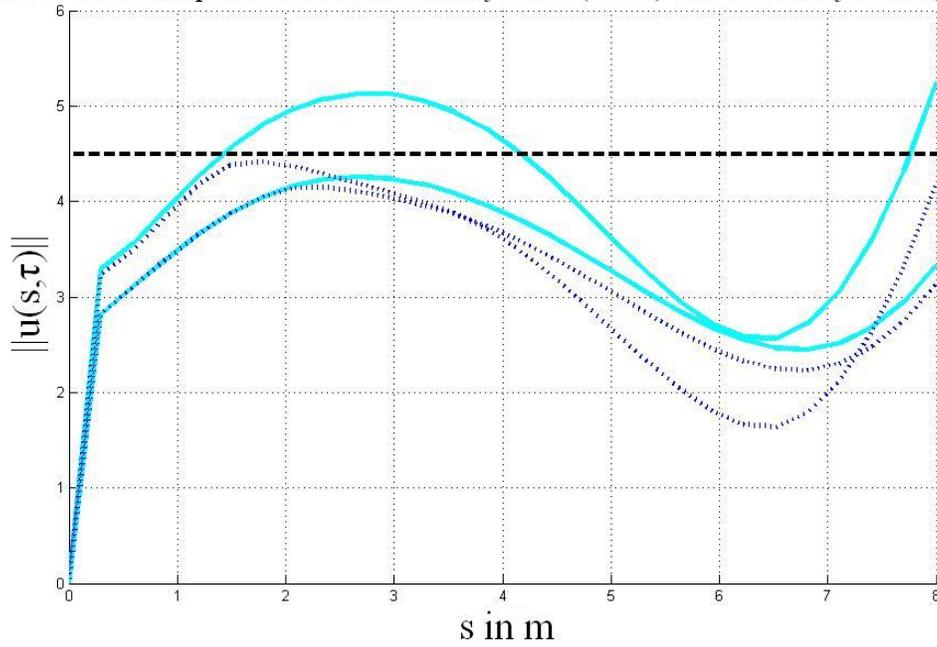
1. Compute $u(s, \tau)$ via Euler integration (3.48), where $\dot{u}(s, \tau)$ is chosen based on projection (3.52), (3.53).

Figure 3.10 shows a simulation of the path deformation algorithm with and without projection for a single obstacle over two iterations of τ . The parameters are chosen as $u_{\max} = 5$, $\bar{u} = 4.5$, $\epsilon_u = 0.2656$. One can see from Figure 3.10 (top) that the kinematic control inputs without projection (solid lines) violate the upper bound $u_{\max} = 5$ (black dashed line), which is corrected by applying projection (dotted lines). Figure 3.10 (bottom) shows the deformed paths around the obstacle with (dotted lines) and without (solid lines) projection. Since the projection operator is limiting the amount of kinematic control inputs available, the path is not able to deform as much as it would without projection, implying that a larger number of iterations may be required to completely avoid an object. Hence, there exists a trade-off between keeping the kinematic control inputs bounded and the number of path deformation iterations that must be computed as noted in the following Remark.

Remark 10 *The projection operator guarantees that kinematic control inputs remain bounded at the cost of increased iterations. Hence, paths generated by the path deformation with projection are sub-optimal as compared to paths generated by the path deformation method without projection. There exists a necessary trade-off between boundedness of kinematic control inputs and optimality with respect to the required number of iterations to avoid obstacles.*

Remark 11 *Equation (3.1) and the collision criterion (3.2) assume that the locations of the obstacle (P_x, P_y) and the distance between the vehicle and the obstacle is exactly known, which is a restrictive assumption. Estimating the value of $z(t)$ at each time step allows for an approximation of (P_x, P_y) to be used in these equations so that explicit knowledge of the obstacle's location is not required. Chapter 4 details the range identification problem and offers an implementable solution.*

Kinematic Control Input Norms Without Projection (Solid) and With Projection (Dotted)



Deformation Paths: Without Projection (Solid) and With Projection (Dashed)

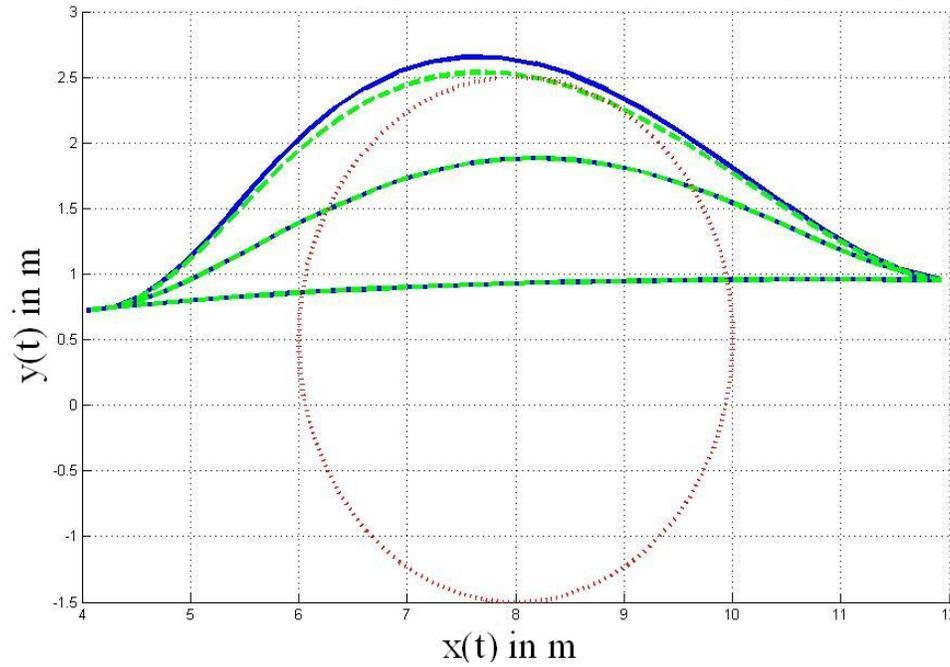


Figure 3.10: Path deformation with and without projection.

3.4 Summary

In this chapter, the path deformation method of [52] is reviewed. The method is then applied to the problem of obstacle avoidance for a fleet of mobile robots. The basis functions used in the path deformation are chosen to be polynomial basis functions to remain consistent with the path generation of Chapter 2. A potential function for obstacle avoidance is given, which satisfies the avoidance objective. The path deformation algorithm is shown to be a stable process that does not lead to diverging paths. Furthermore, the projection operator is used to ensure that kinematic input constraints on the vehicles are satisfied throughout the deformation. To account for multiple vehicles, a separation law between vehicles is defined which guarantees that the vehicles are deconflicted throughout the avoidance scheme.

Chapter 4

Vision-Based Range Identification

For mobile robots, the range identification problem is formulated in a two-dimensional setting [12]. In order to estimate the relative range between the vehicle and an obstacle, two approaches are utilized. The first is an existing identifier-based observer (IBO) [41] that provides exponential convergence of both the range and parameter estimates under a certain set of assumptions. The second is the fast adaptive estimator [61, 17, 62] that enables estimation of the unknown parameters in the system dynamics via fast adaptation (large adaptive gain) and a low-pass filter. Section 4.1 formulates the vision-based estimation problem of a static obstacle. Application of the IBO and the fast estimator is performed in Sections 4.2 and 4.3, respectively. Numerical comparison of the performance between these two methods and their effect on the obstacle avoidance component is provided in Section 4.4.

4.1 Problem Formulation

Consider the two-dimensional scenario of a vision-based range estimation problem as shown in Figure 4.1. It is assumed that a conventional pin-hole model for the camera is used and that image processing algorithms are available to extract the bearing angle $\beta(t)$ and the

subtended angle $\alpha(t)$. The objective is to estimate the relative range $z(t)$ between the vehicle and an obstacle using the visual measurements $\alpha(t)$ and $\beta(t)$, as well as $\psi(t), r(t), v(t)$ that are available from onboard sensors. This is known in the literature as the *range identification* problem [61], [16].

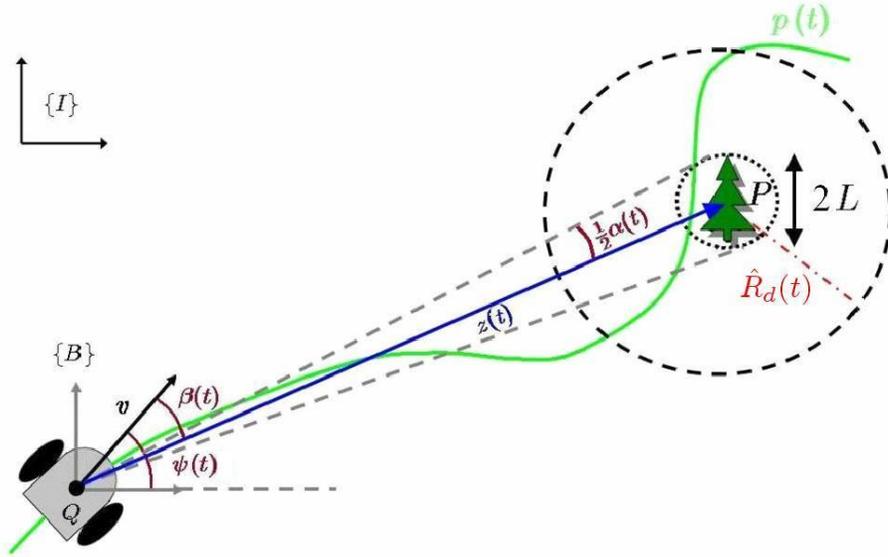


Figure 4.1: Visual measurements $\alpha(t)$ and $\beta(t)$.

Let $(z_x(t), z_y(t))$ be the vector of relative distance between the robot and the obstacle in the inertial frame so that $z(t) = \sqrt{z_x^2(t) + z_y^2(t)}$. The quantities $z_x(t), z_y(t), \alpha(t), \beta(t)$, and $z(t)$ are defined with respect to the inertial frame.

In the kinematic setting, the relative dynamics are given by:

$$\begin{aligned}\dot{z}_x(t) &= \dot{P}_x - \dot{x}_I(t) = -v(t) \cos \psi(t), \\ \dot{z}_y(t) &= \dot{P}_y - \dot{y}_I(t) = -v(t) \sin \psi(t), \\ \dot{\psi}(t) &= r(t).\end{aligned}\tag{4.1}$$

The bearing angle $\beta(t)$ is

$$\beta(t) = \psi(t) - \tan^{-1} \left(\frac{z_y(t)}{z_x(t)} \right),\tag{4.2}$$

and the subtended angle $\alpha(t)$ of the obstacle in the image plane is

$$\alpha(t) = 2 \tan^{-1} \left(\frac{L}{z(t)} \right). \quad (4.3)$$

Note from the definition of $\alpha(t)$ that $0 < \alpha(t) < \pi$ for all $t \geq 0$. If L is known, one can compute $z(t)$ from the following relationship

$$z(t) = \frac{L}{\tan(\alpha(t)/2)}. \quad (4.4)$$

Hence, knowledge of $z(t)$ is equivalent to knowledge of L since $\alpha(t)$ is measurable. From equations (4.2), (4.3), and the definition of $z(t)$, it is straightforward to get

$$\begin{aligned} z_x(\alpha(t), \beta(t), \psi(t)) &= \frac{L \cos(\psi(t) - \beta(t))}{\tan(\alpha(t)/2)}, \\ z_y(\alpha(t), \beta(t), \psi(t)) &= \frac{L \sin(\psi(t) - \beta(t))}{\tan(\alpha(t)/2)}. \end{aligned} \quad (4.5)$$

It can be computed from (4.1) and (4.5) that

$$\begin{aligned} \dot{z}_x(t) &= \frac{\partial z_x}{\partial \alpha}(t) \dot{\alpha}(t) + \frac{\partial z_x}{\partial \beta}(t) \dot{\beta}(t) + \frac{\partial z_x}{\partial \psi}(t) \dot{\psi}(t) = -v(t) \cos \psi(t), \\ \dot{z}_y(t) &= \frac{\partial z_y}{\partial \alpha}(t) \dot{\alpha}(t) + \frac{\partial z_y}{\partial \beta}(t) \dot{\beta}(t) + \frac{\partial z_y}{\partial \psi}(t) \dot{\psi}(t) = -v(t) \sin \psi(t). \end{aligned} \quad (4.6)$$

Solving (4.6) for $\dot{\alpha}(t)$ and $\dot{\beta}(t)$, one can obtain

$$\begin{bmatrix} \dot{\alpha}(t) \\ \dot{\beta}(t) \end{bmatrix} = \frac{1}{L} \begin{bmatrix} 2v(t) \sin^2 \left(\frac{\alpha(t)}{2} \right) \cos \beta(t) \\ v(t) \tan \left(\frac{\alpha(t)}{2} \right) \sin \beta(t) \end{bmatrix} + \begin{bmatrix} 0 \\ r(t) \end{bmatrix}. \quad (4.7)$$

Let

$$f(t) = \begin{bmatrix} f_1(\alpha(t), \beta(t), v(t)) \\ f_2(\alpha(t), \beta(t), v(t)) \end{bmatrix} = \begin{bmatrix} 2v(t) \sin^2 \left(\frac{\alpha(t)}{2} \right) \cos \beta(t) \\ v(t) \tan \left(\frac{\alpha(t)}{2} \right) \sin \beta(t) \end{bmatrix}, \quad \theta = \frac{1}{L}. \quad (4.8)$$

The system (4.7) can be rewritten as:

$$\begin{bmatrix} \dot{\alpha}(t) \\ \dot{\beta}(t) \end{bmatrix} = \theta \begin{bmatrix} f_1(\alpha(t), \beta(t), v(t)) \\ f_2(\alpha(t), \beta(t), v(t)) \end{bmatrix} + \begin{bmatrix} 0 \\ r(t) \end{bmatrix}. \quad (4.9)$$

The *estimation objective* is to design an adaptive estimator to estimate the value of the unknown parameter $\theta = \frac{1}{L}$, upon which the relationship (4.4) will yield the value of $\hat{z}(t)$, which is the estimate of the relative range between the vehicle and an obstacle. The problem formulation assumes that if an obstacle is within a vehicle's sensing range, an estimate is available via onboard visual sensors. In reality, poor lighting and occlusion of obstacles may occur which is not considered in this formulation. Future research can be directed at addressing such problems.

Remark 12 *If $\|v(t)\| > 0$ for all $t \geq 0$, the values of $f_1(t)$ and $f_2(t)$ cannot both be equal to zero at any time $t \geq 0$. If $v(t) = 0$, then there is no danger of collision with the obstacle and estimation of θ is irrelevant. Successful implementation of the obstacle avoidance algorithm in Chapter 3 implies that the vehicle will always remain at a minimum distance of $R_d > 0$ away from the obstacle. Hence, $\alpha(t) \leq 2 \tan^{-1} \left(\frac{L}{R_d} \right) < \pi$ for all $t \geq 0$. A finite sensor region R_s implies that $0 < 2 \tan^{-1} \left(\frac{L}{R_s} \right) \leq \alpha(t)$ and $\alpha(t) \in \left[2 \tan^{-1} \left(\frac{L}{R_s} \right), 2 \tan^{-1} \left(\frac{L}{R_d} \right) \right]$. It follows that at least one of $f_1(t)$ or $f_2(t)$ are invertible as long as $v(t) \neq 0$. It is also useful to note that if $v(t)$ is uniformly bounded, $f_1(t)$ and $f_2(t)$ are also uniformly bounded.*

4.2 Range Identification using IBO

Consider the state estimation problem for the dynamical system given in (4.9) where the unknown quantity is the constant θ . The system fits into the form to which the IBO can be applied directly. To remain consistent with the notation in [41], bold letters \mathbf{x} and \mathbf{u} are used to review the IBO.

The IBO is designed for a class of nonlinear systems with the following structure [41]:

$$\begin{aligned}\dot{\mathbf{x}}_1(t) &= \mathbf{w}_s^\top(\mathbf{x}_1(t), \mathbf{u}(t))\mathbf{x}_2(t) + \varphi(\mathbf{x}_1(t), \mathbf{u}(t)), \\ \dot{\mathbf{x}}_2(t) &= g_s(\mathbf{x}_1(t), \mathbf{x}_2(t), \mathbf{u}(t)), \\ \mathbf{y}(t) &= \mathbf{x}_1(t),\end{aligned}\tag{4.10}$$

where $\mathbf{x}_1(t) \in \mathbb{X}_1 \subset \mathbb{R}^{n_1}$, $\mathbf{x}_2(t) \in \mathbb{X}_2 \subset \mathbb{R}^{n_2}$ and $\mathbf{u}(t) \in \mathbb{U} \subset \mathbb{R}^k$. The $n_1 \times n_2$ matrix $w_s^\top(\mathbf{x}_1(t), \mathbf{u}(t))$ and the vector $g_s(\mathbf{x}_1(t), \mathbf{x}_2(t), \mathbf{u}(t))$ are general nonlinear functions of their parameters. Let $\mathbf{x}(t) = [\mathbf{x}_1^\top(t), \mathbf{x}_2^\top(t)]^\top$ and $n = n_1 + n_2$ so that $\mathbf{x}(t) \in \mathbb{X} \subset \mathbb{R}^n$, where $\mathbb{X} = \mathbb{X}_1 \oplus \mathbb{X}_2$.

Following Ref. [41], we introduce the following assumptions:

Assumption 1

1. Let $\mathbf{x}(t)$ be bounded: $\|\mathbf{x}(t)\| < M$, where $M > 0$ for every $t \geq 0$. Let $\Omega = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}(t)\| < M\}$. Further, for some fixed constant $\gamma > 1$, let $\Omega_\gamma = \{\mathbf{x}(t) \in \mathbb{R}^n : \|\mathbf{x}(t)\| < \gamma M\}$. Assume that the function $g_s(\mathbf{x}_1(t), \mathbf{x}_2(t), \mathbf{u}(t))$ is locally Lipschitz in Ω_γ with respect to $\mathbf{x}_2(t)$, i.e., there exists a positive constant ι such that

$$\|g_s(\mathbf{x}_1, \mathbf{x}_2, \mathbf{u}) - g_s(\mathbf{x}_1, \mathbf{z}_2, \mathbf{u})\| < \iota \|\mathbf{x}_2 - \mathbf{z}_2\|, \quad (4.11)$$

for all $\mathbf{x}_2(t), \mathbf{z}_2(t) \in \Omega_\gamma \cap \mathbb{X}_2$, uniformly in $\mathbf{x}_1(t) \in \Omega_\gamma \cap \mathbb{X}_1$ and $\mathbf{u}(t) \in \mathbb{U}$.

2. Let the regressor matrix $w_s^\top(\mathbf{x}_1(t), \mathbf{u}(t))$ and its first time derivative be piecewise smooth and uniformly bounded. Further, assume that there exist positive constants L_1, L_2, L_3 and L_4 such that

$$\|w_s^\top(\mathbf{x}_1, \mathbf{u})\| < L_1, \quad \left\| \frac{dw_s^\top(\mathbf{x}_1, \mathbf{u})}{dt} \right\| < L_2, \quad (4.12a)$$

$$\int_t^{t+L_4} w_s(\mathbf{x}_1(\tau), \mathbf{u}(\tau)) w_s^\top(\mathbf{x}_1(\tau), \mathbf{u}(\tau)) d\tau > L_3 \mathbb{I} \quad (4.12b)$$

for all $t \geq 0$, for all trajectories that originate in \mathbb{X} and for all $\mathbf{u}(t) \in \mathbb{U}$, while \mathbb{I} denotes the identity matrix of appropriate dimension.

Letting $\hat{\mathbf{x}}_1(t)$ and $\hat{\mathbf{x}}_2(t)$ be the estimates of $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$, respectively, the IBO is introduced as:

$$\begin{cases} \dot{\hat{\mathbf{x}}}_1(t) = G_1 A_m (\hat{\mathbf{x}}_1 - \mathbf{x}_1) + w_s^\top(\mathbf{x}_1, \mathbf{u}) \hat{\mathbf{x}}_2 + \varphi(\mathbf{x}_1, \mathbf{u}), \\ \dot{\hat{\mathbf{x}}}_2(t) = -G_1^2 w_s(\mathbf{x}_1, \mathbf{u}) P (\hat{\mathbf{x}}_1 - \mathbf{x}_1) + g_s(\mathbf{x}_1, \hat{\mathbf{x}}_2, \mathbf{u}), \\ \hat{\mathbf{x}}(t_i^+) = M \frac{\hat{\mathbf{x}}(t_i^-)}{\|\hat{\mathbf{x}}(t_i^-)\|}, \end{cases} \quad (4.13)$$

where G_1 is a scalar constant and A_m is an $n_1 \times n_1$ Hurwitz matrix. The matrix P is the positive definite solution of the Lyapunov equation $A_m^\top P + P A_m = -Q$ for some $Q > 0$. The sequence t_i is defined as follows:

$$t_i = \min\{t : t > t_{i-1} \text{ and } \|\hat{\mathbf{x}}(t)\| \geq \gamma M\}, \quad t_0 = 0, \quad (4.14)$$

where γ is a fixed constant.

Theorem 1 [41] *Subject to Assumption 1, there exists a positive constant G_0 such that the estimation errors $\mathbf{e}_1(t) = \mathbf{x}_1(t) - \hat{\mathbf{x}}_1(t)$ and $\mathbf{e}_2(t) = \mathbf{x}_2(t) - \hat{\mathbf{x}}_2(t)$ converge to zero exponentially, if the constant G_1 in (4.13) is chosen larger than G_0 .*

Returning to the system in (4.9), let

$$x_1(t) = \begin{bmatrix} \alpha(t) \\ \beta(t) \end{bmatrix}, \quad \varphi(r(t)) = \begin{bmatrix} 0 \\ r(t) \end{bmatrix}, \quad g_s(x_1(t), \theta, v(t), r(t)) = g_s \equiv 0, \quad (4.15)$$

and define the regressor vector $w_s(t)$ to be

$$w_s(x_1(t), v(t)) = \begin{bmatrix} f_1(x_1(t), v(t)) & f_2(x_1(t), v(t)) \end{bmatrix}. \quad (4.16)$$

The system in (4.7) can be rewritten as

$$\begin{aligned} \dot{x}_1(t) &= w_s^\top(x_1(t), v(t))\theta + \varphi(r(t)), \\ \dot{\theta} &= g_s. \end{aligned} \quad (4.17)$$

Note that $g_s(t) \equiv 0$ is globally Lipschitz. For application of the IBO, the following assumption needs to be satisfied for the system in (4.17).

Assumption 2 :

1. *Let the function $w_s(x_1(t), v(t))$ and its first time derivative be piecewise smooth and uniformly bounded. That is, suppose there exist positive constants L_1, L_2 such that*

$$\|w_s^\top(x_1(t), v(t))\| < L_1, \quad \left\| \frac{dw_s^\top(x_1(t), v(t))}{dt} \right\| < L_2. \quad (4.18)$$

2. There does not exist a nonzero constant ϖ such that

$$\varphi w_s^\top(x_1(\bar{t}), v(\bar{t})) = 0, \quad (4.19)$$

for all $\bar{t} \in [t, t + \rho]$, where $\rho > 0$ is a sufficiently small constant.

In light of Remark 12, it is straightforward to show that the system in (4.9) verifies the assumptions required for application of the IBO. Estimation of the unknown parameter θ , and hence the range $z(t)$, can be obtained by direct application of the IBO, as shown below.

Letting $\tilde{x}_1(t) = \hat{x}_1(t) - x_1(t)$ and $\tilde{\theta}(t) = \hat{\theta}(t) - \theta$, the following observer can be designed for the system in (4.17):

$$\begin{aligned} \dot{\hat{x}}_1(t) &= G_1 A_m \tilde{x}_1(t) + w_s^\top(x_1(t), v(t)) \hat{\theta}(t) + \varphi(r(t)), \\ \dot{\hat{\theta}}(t) &= -G_1^2 w_s(x_1(t), v(t)) P \tilde{x}_1(t) + g_s, \end{aligned} \quad (4.20)$$

where G_1 is a positive constant, A_m is a Hurwitz matrix, and P is the symmetric positive definite matrix that satisfies the Lyapunov equation $A_m^\top P + P A_m = -Q$ for a positive-definite symmetric matrix Q . The closed-loop error dynamics can be computed from (4.17) and (4.20) as

$$\begin{aligned} \dot{\tilde{x}}_1(t) &= G_1 A_m \tilde{x}_1(t) + w^\top(x_1(t), v(t)) \tilde{\theta}(t), \\ \dot{\tilde{\theta}}(t) &= -G_1^2 w(x_1(t), v(t)) P \tilde{x}_1(t) + g_s. \end{aligned} \quad (4.21)$$

According to Theorem 1, there exists a positive constant G_0 such that the estimation errors $[\tilde{x}_1(t) \ \tilde{\theta}(t)]^\top$ converge to zero exponentially if the constant G_1 in (4.20) is chosen greater than G_0 .

4.3 Range Identification using fast estimator

In this section, a recently-developed fast estimation scheme is applied for the range identification problem formulated in Section 4.1. The adaptive estimator in Refs. [61, 17, 62] allows

for fast adaptation of the unknown parameter via large adaptation gain and low-pass filter to guarantee the absence of high frequency signals in the control channel. With this respect, recall the following definitions:

Definition 1 For a signal $\xi(t)$, $t \geq 0$, $\xi(t) \in \mathbb{R}^n$, its \mathcal{L}_∞ norm is defined as

$$\|\xi(t)\|_{\mathcal{L}_\infty} = \max_{i=1,\dots,n} \left(\sup_{\tau \geq 0} |\xi_i(\tau)| \right), \quad (4.22)$$

where $\xi_i(t)$ is the i th component of $\xi(t)$.

Definition 2 The \mathcal{L}_1 gain of a stable proper single-input single-output system $G(s)$ is defined as:

$$\|G(s)\|_{\mathcal{L}_1} = \int_0^\infty |g(t)| dt, \quad (4.23)$$

where $g(t)$ is the impulse response of $G(s)$.

Definition 3 For a stable m -input n -output system $G(s)$, its \mathcal{L}_1 gain is defined as

$$\|G(s)\|_{\mathcal{L}_1} = \max_{i=1,\dots,n} \left(\sum_{j=1}^m \|G_{ij}(s)\|_{\mathcal{L}_1} \right), \quad (4.24)$$

where $G_{ij}(s)$ is the i th row, j th column component of $G(s)$.

A review of the fast estimator is given first for the following general system dynamics:

$$\dot{x}(t) = A_m x(t) + \omega(t), \quad x(0) = x_0, \quad (4.25)$$

where A_m is a known Hurwitz matrix, $x(t) \in \mathbb{R}^n$ is the (measurable) vector of states, and $\omega(t) \in \mathbb{R}^n$ is a vector of unknown time-varying signals or parameters which is assumed to belong to a known compact set Ω so that $\omega(t) \in \Omega$. The signal $\omega(t)$ is assumed to be uniformly bounded and continuously differentiable with uniformly bounded derivative. That is,

$$\|\omega(t)\| \leq b_1 < \infty \quad \text{and} \quad \|\dot{\omega}(t)\| \leq b_2 < \infty, \quad \forall t \geq 0, \quad (4.26)$$

where b_1 and b_2 are positive constants. The estimation of $\omega(t)$ can be obtained via the following steps:

State Predictor Consider the following state predictor

$$\dot{\hat{x}}(t) = A_m \hat{x}(t) + \hat{\omega}(t), \quad \hat{x}(0) = x_0, \quad (4.27)$$

which matches the structure of (4.25) with the estimate $\hat{\omega}(t)$ replacing the unknown vector $\omega(t)$.

Adaptive Law The adaptive estimates propagate according to the dynamics

$$\dot{\hat{\omega}}(t) = \Gamma_c \text{Proj}(\hat{\omega}(t), -P\tilde{x}(t)), \quad \hat{\omega}(0) = \hat{\omega}_0, \quad (4.28)$$

where $\Gamma_c \in \mathbb{R}^+$ is the adaptation gain, chosen sufficiently large, $\tilde{x}(t) = \hat{x}(t) - x(t)$ is the error signal between the state predictor and the actual state of the system, P is the solution of the Lyapunov equation $A_m^\top P + PA = -Q$ for some choice of $Q > 0$, and $\text{Proj}(\cdot, \cdot)$ denotes the projection operator.

Estimation The estimation of the unknown signal is generated by:

$$\omega_e(s) = C(s)\hat{\omega}(s), \quad (4.29)$$

where $C(s)$ is a diagonal matrix with its n th diagonal element $C_n(s)$ being a strictly proper stable transfer function with low-pass gain $C_n(0) = 1$. One such choice is

$$C_n(s) = \frac{c}{s + c}, \quad (4.30)$$

where $c > 0$.

The fast adaptive estimator ensures that $\omega_e(t)$ estimates the unknown signal $\omega(t)$ with the final precision after transient ($t \rightarrow \infty$) [61], [16]:

$$\|\omega_e - \omega\|_{\mathcal{L}_\infty} \leq \|\omega_e - \omega_r\|_{\mathcal{L}_\infty} + \|\omega_r - \omega\|_{\mathcal{L}_\infty} \leq \frac{\gamma_c}{\sqrt{\Gamma_c}} + \|1 - C(s)\|_{\mathcal{L}_1} \|\omega\|_{\mathcal{L}_\infty}. \quad (4.31)$$

The precision for all $t \geq 0$ is:

$$\|\omega_e - \omega\|_{\mathcal{L}_\infty} \leq \|\hat{\omega}_0 - \omega_0\|_{\mathcal{L}_\infty} e^{-ct} + \frac{\gamma_c}{\sqrt{\Gamma_c}} + \|1 - C(s)\|_{\mathcal{L}_1} \|\omega\|_{\mathcal{L}_\infty}, \quad \forall t \geq 0. \quad (4.32)$$

The proof of the convergence results for the \mathcal{L}_1 fast estimator relies on three steps. First, the tracking error $\tilde{x}(t)$ can be shown to be bounded via a Lyapunov approach with bound

$$\|\tilde{x}(t)\|_{\mathcal{L}_\infty} \leq \sqrt{\frac{\omega_m}{\lambda_{\min}(P)\Gamma_c}}. \quad (4.33)$$

Using (4.33), it can be shown through some algebraic manipulations that

$$\|\omega_e(t) - \omega_r(t)\|_{\mathcal{L}_\infty} \leq \frac{\gamma_c}{\sqrt{\Gamma_c}}, \quad (4.34)$$

where $\omega_r(t)$ is an intermediate signal defined for the purposes of the proof. The last step involves characterization of the performance bound to obtain

$$\|\omega_r(t) - \omega(t)\|_{\mathcal{L}_\infty} = \|1 - C(s)\|_{\mathcal{L}_1} \|\omega(t)\|_{\mathcal{L}_\infty}. \quad (4.35)$$

The final result (4.31) follows upon application of the triangle inequality. For details, please see Refs. [61, 17, 62].

Remark 13 *One can see from (4.31) and (4.32) that the performance of the adaptive estimator is dependent on the design parameters Γ_c and the bandwidth of $C(s)$. From the second term in (4.31), it can be seen that increasing the bandwidth of $C(s)$ minimizes the \mathcal{L}_1 gain of $1 - C(s)$ and leads to improved transient time and final estimation precision. However, increasing bandwidth of $C(s)$ increases γ_c , which in turn requires an increase of Γ_c to keep the performance bound $\frac{\gamma_c}{\sqrt{\Gamma_c}}$ small as seen in the first term of (4.31). Large adaptation gain Γ_c requires faster computation and smaller integration step and is limited by available hardware. Since the fast adaptive estimator assumes minimization of the \mathcal{L}_1 gain of $1 - C(s)$ for performance improvement, it is referred to as the \mathcal{L}_1 fast estimator.*

For the system of interest in (4.9), define

$$\Phi(t) = \begin{bmatrix} \Phi_1(t) \\ \Phi_2(t) \end{bmatrix} \triangleq \theta f(t) = \theta \begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix}. \quad (4.36)$$

The system in (4.7) can be rewritten as

$$\begin{bmatrix} \dot{\alpha}(t) \\ \dot{\beta}(t) \end{bmatrix} = \Phi(t) + \begin{bmatrix} 0 \\ r(t) \end{bmatrix}. \quad (4.37)$$

Treating $\Phi(t)$ as the unknown signal, the estimate of $\Phi(t)$, denoted by $\Phi_e(t)$, can be obtained using the fast estimator through the following steps:

- **State Predictor:**

$$\begin{bmatrix} \dot{\hat{\alpha}}(t) \\ \dot{\hat{\beta}}(t) \end{bmatrix} = A_m \begin{bmatrix} \tilde{\alpha}(t) \\ \tilde{\beta}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ r(t) \end{bmatrix} + \hat{\Phi}(t), \quad \begin{bmatrix} \hat{\alpha}(0) \\ \hat{\beta}(0) \end{bmatrix} = \begin{bmatrix} \alpha(0) \\ \beta(0) \end{bmatrix}, \quad (4.38)$$

where

$$\begin{bmatrix} \tilde{\alpha}(t) \\ \tilde{\beta}(t) \end{bmatrix} = \begin{bmatrix} \hat{\alpha}(t) \\ \hat{\beta}(t) \end{bmatrix} - \begin{bmatrix} \alpha(t) \\ \beta(t) \end{bmatrix}. \quad (4.39)$$

- **Adaptive Law (Γ_c chosen to be large):**

$$\dot{\hat{\Phi}}(t) = \Gamma_c \text{Proj} \left(\hat{\Phi}(t), -P \begin{bmatrix} \tilde{\alpha}(t) \\ \tilde{\beta}(t) \end{bmatrix} \right), \quad \hat{\Phi}(0) = \hat{\Phi}_0. \quad (4.40)$$

- **Estimation:**

$$\Phi_e(s) = C(s)\hat{\Phi}(s), \quad C(s) = \frac{c}{s+c}, \quad c > 0, \quad (4.41)$$

where $\Phi_e(t)$ is the filtered estimate of $\Phi(t)$.

- **Extraction:** Knowledge of $\Phi_e(t)$ implies that one can solve for $\hat{\theta}(t)$, the estimate of θ , using the relationship

$$\hat{\theta}(t) = \sqrt{\frac{\Phi_{e_1}(t)^2 + \Phi_{e_2}(t)^2}{f_1^2(t) + f_2^2(t)}}, \quad (4.42)$$

where $\Phi_e(t) = [\Phi_{e_1}(t) \ \Phi_{e_2}(t)]^\top$. If $v(t) = 0$, one cannot obtain an estimate for θ and no update can be made on $\hat{\theta}(t)$ as long as $v(t)$ remains zero. As a consequence of Remark 12, equation (4.42) is always well defined. Since the vehicle is following a desired trajectory with a desired speed profile, the speed of the vehicle cannot remain zero for all time and estimation will be achieved for any non-zero velocity.

- **Range Identification:** Once $\hat{\theta}(t)$ is available, the range between the vehicle and an obstacle can be computed via

$$\hat{z}(t) = \frac{\hat{\theta}^{-1}(t)}{\tan(\alpha(t)/2)}. \quad (4.43)$$

Remark 14 Since the parameter θ is a scalar and known to be positive, the compact set Ω can be chosen so that $\Omega \subset \mathbb{R}^+$. Hence, $\hat{\theta}(t) \in \Omega \subset \mathbb{R}^+$ for all $t \geq 0$ and (4.43) is well-defined. Knowledge of the conservative upper bound L_{\max} implies that Ω can be chosen to respect this bound so that $\hat{L}(t) \leq L_{\max}$ and $\hat{\theta} \geq \frac{1}{L_{\max}} > 0$, where $\hat{L}(t)$ is the estimate of L .

As noted in Remark 13, the estimation precision $\Phi_e(t) - \Phi(t)$ and the transient time to achieve this can be arbitrarily reduced by increasing the bandwidth of the low-pass filter $C(s)$. Increasing the bandwidth of $C(s)$ implies increasing the adaptation gain Γ_c , which is limited by available hardware. Estimation of the range $z(t)$ is finally obtained through the algebraic expressions (4.42) and (4.43) with the available measurements $v(t)$, $\alpha(t)$, and $\beta(t)$. The flow chart of the range identification via the \mathcal{L}_1 fast estimator is shown in Figure 4.2.

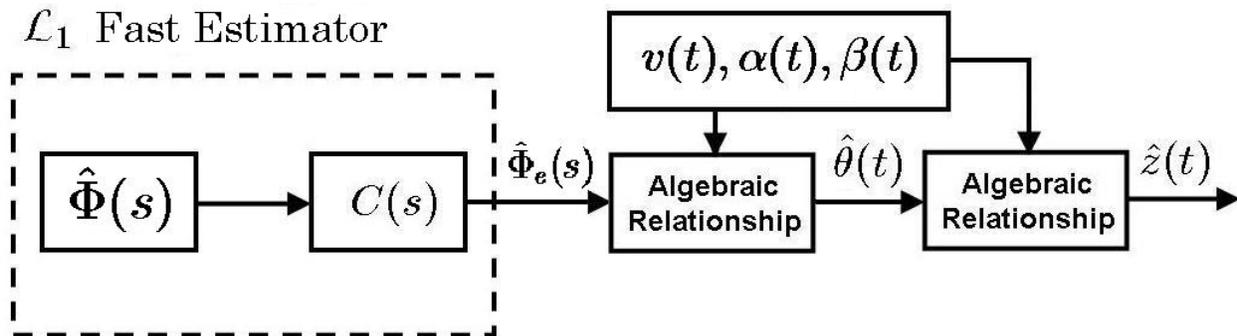


Figure 4.2: Flow chart of range identification using the \mathcal{L}_1 fast estimator.

Given an estimate for $z(t)$ and the available measurements for bearing angle $\beta(t)$ and yaw angle $\psi(t)$, equations (3.1) – (3.24) are redefined replacing the unknown position of the obstacle $(P_x(t), P_y(t))$ with its estimate $(\hat{P}_x(t), \hat{P}_y(t))$, given by:

$$(\hat{P}_x(t), \hat{P}_y(t)) = \left(\hat{z}(t) \cos(\psi(t) - \beta(t)), \hat{z}(t) \sin(\psi(t) - \beta(t)) \right). \quad (4.44)$$

Similarly, $z(t)$, which is the relative distance between the vehicle and the obstacle, is replaced by $\hat{z}(t)$.

Remark 15 *It is important to note that by replacing the original quantities $(P_x(t), P_y(t))$ and $z(t)$ with their estimates and generating new portions of the path via the obstacle avoidance algorithm in Chapter 3, the stability and convergence properties of the path following controller (2.89) are not lost. The path following controller utilizes the curvature of a path and feeds back errors between a path and the vehicle. If the path changes continuously, this is equivalent to a new set of initial conditions and curvature for a new path. The path following errors will be discontinuous whenever the path changes, but the stability results obtained in Chapter 2 Section 2.4 will hold independent of changes in the path as long as the path itself is not discontinuous, which is guaranteed by imposing the boundary conditions in (3.19).*

4.4 Effect of Estimation Schemes on Obstacle Avoidance

Proper estimation of the danger radius $\hat{R}_d(t)$ is essential to ensure obstacle avoidance. The value of $\hat{R}_d(t)$ is related to the estimated size of the obstacle L and to the performance of the estimation schemes. One can notice that for safety considerations, the estimate $\hat{R}_d(t)$ must assume larger values as compared to R_d . One could choose $\hat{R}_d(t) = \hat{R}_d = L_{\max} + d$, where L_{\max} is an upper bound on L and d is an *a priori* specified constant, which would guarantee that $\hat{R}_d(t) \geq R_d$ for all $t \geq 0$. However, this represents a “worst case” scenario and assumes the largest possible value for the obstacle. The new path for obstacle avoidance would lead the vehicle farther away from an obstacle than is needed, leading to additional fuel consumption and delay. To reduce this additional cost, $\hat{R}_d(t) \geq R_d$ can be derived for both cases when using the IBO and the fast estimator which takes into account each obstacle’s estimated size and leads to a less conservative estimate for R_d . Furthermore, this

estimate can be improved arbitrarily in the case of the fast estimator by proper choice of the design parameters.

Application of the IBO implies that the estimation error $\|\hat{\theta}(t) - \theta\|$ decays to zero exponentially. From the stability proof of the IBO in Ref. [41], there exists a positive constant ς such that

$$\|\hat{\theta}(t) - \theta\|_{\mathcal{L}_\infty} \leq \|\hat{\theta}(0) - \theta\|_{\infty} e^{-\varsigma t} \triangleq \gamma_1(t). \quad (4.45)$$

It follows from (4.8) that L is the inverse of θ , and accordingly, $\hat{L}(t)$ is the inverse of $\hat{\theta}(t)$. Straightforward algebraic manipulations show that

$$\|\hat{L} - L\|_{\mathcal{L}_\infty} = \|L\hat{L}(\hat{\theta}(t) - \theta)\|_{\mathcal{L}_\infty} \leq L_{\max} (L_{\max} + h(t)) \gamma_1(t), \quad (4.46)$$

where $h(t) = \|\hat{L}(0) - L_{\max}\| e^{-\varsigma t}$ is an exponentially decaying term that decays with the rate ς .

Proposition 9 *In the application of the IBO, if the estimate $\hat{R}_d(t)$ of R_d is chosen as:*

$$\hat{R}_{d\text{IBO}}(t) = \hat{L}(t) + d + L_{\max} (L_{\max} + h(t)) \gamma_1(t), \quad (4.47)$$

where L_{\max} is an upper bound on the size of the obstacle, then $\hat{R}_{d\text{IBO}}(t) \geq R_d$ for all $t \geq 0$.

Proof. One can compute

$$\begin{aligned} \hat{R}_{d\text{IBO}}(t) - R_d &= \hat{L}(t) + d + L_{\max} (L_{\max} + h(t)) \gamma_1(t) - (L + d) \\ &= \hat{L}(t) - L + L_{\max} (L_{\max} + h(t)) \gamma_1(t) \\ &\geq -\|\hat{L}(t) - L\|_{\mathcal{L}_\infty} + L_{\max} (L_{\max} + h(t)) \gamma_1(t) \\ &\geq 0. \end{aligned}$$

□

Remark 16 *From (4.47), it can be seen that at $t = 0$, the estimate $\hat{R}_d(t)$ is more conservative than setting $\hat{R}_d(t) = L_{\max} + d$. However, as t increases, the terms $h(t)$ and $\gamma_1(t)$*

decrease exponentially so that the estimate $\hat{R}_d(t)$ is close to the estimated size of the obstacle $\hat{L}(t)$ and is less than $L_{\max} + d$.

From (4.32), the error of the fast estimator is bounded above by

$$\|\Phi_e(t) - \Phi(t)\|_{\mathcal{L}_\infty} \leq \|\hat{\Phi}_0 - \Phi_0\|_\infty e^{-ct} + \frac{\gamma_c}{\sqrt{\Gamma_c}} + \|1 - C(s)\|_{\mathcal{L}_1} \|\Phi(t)\|_{\mathcal{L}_\infty} \triangleq \gamma_2(t), \quad (4.48)$$

where c is the bandwidth of the low-pass filter introduced in (4.41), $\Phi_0 = \Phi(0)$, and

$$\gamma_c = \|C(s)(s\mathbb{I}_2 - A_m)^{-1}\|_{\mathcal{L}_1} \sqrt{\frac{\omega_m}{\lambda_{\min}(P)}}, \quad (4.49)$$

where ω_m is a constant that depends on P, Q, Ω , and the bounds on $f_1(t), f_2(t)$ and their derivatives, respectively. To find the estimation bound for $\|\hat{\theta}(t) - \theta\|_{\mathcal{L}_\infty}$, it is computed from (4.42) and (4.48) that:

$$\|\hat{\theta}(t) - \theta\|_{\mathcal{L}_\infty} \leq \frac{\|\Phi_e(t) - \Phi(t)\|_{\mathcal{L}_\infty}}{\sqrt{f_1^2(t) + f_2^2(t)}} \leq \frac{\gamma_2(t)}{\sqrt{f_1^2(t) + f_2^2(t)}} \triangleq \gamma_3(t) < \infty. \quad (4.50)$$

Following the same calculations above as in the IBO case and using the fact that $\hat{L}(t) \leq L_{\max}$ (see Remark 14), the following Proposition holds.

Proposition 10 *In the application of the fast estimator, if the estimate $\hat{R}_d(t)$ of R_d is chosen as:*

$$\hat{R}_{d_{\text{FE}}}(t) = \hat{L}(t) + d + L_{\max}^2 \gamma_3(t), \quad (4.51)$$

where $\gamma_2(t), \gamma_3(t)$ are the bounded terms defined in (4.48), (4.50), respectively, then $\hat{R}_{d_{\text{FE}}}(t) \geq R_d$ for all $t \geq 0$.

Proof.

$$\begin{aligned} \hat{R}_{d_{\text{FE}}}(t) - R_d &= \hat{L}(t) + d + L_{\max}^2 \gamma_3(t) - (L + d) \\ &= \hat{L}(t) - L + L_{\max}^2 \gamma_3(t) \\ &\geq -\|\hat{L}(t) - L\|_{\mathcal{L}_\infty} + L_{\max}^2 \gamma_3(t) \\ &\geq 0. \end{aligned}$$

□

Remark 17 From (4.48), (4.50), and (4.51), it can be seen that at $t = 0$, the estimate $\hat{R}_d(t)$ is again more conservative than setting $\hat{R}_d(t) = L_{\max} + d$. However, the bandwidth of the low-pass filter $C(s)$ can be increased so that the first term in $\gamma_2(t)$ decreases exponentially as $t \rightarrow \infty$. The additional terms that appear in the application of the fast estimator can be made arbitrarily small by increasing c and the adaptation gain Γ_c so that the estimate $\hat{R}_d(t)$ is close to the estimated size of the obstacle $\hat{L}(t)$ and is less than $L_{\max} + d$.

In light of Remarks 16 and 17, choosing $\hat{R}_d(t)$ in this way allows for a less conservative estimate of R_d since $\gamma_1(t)$ and $\gamma_3(t)$ contain exponentially decaying terms that can vanish arbitrarily fast with appropriate choice of ς, c as t increases. However, the value of c is easily chosen by the control designer as compared to the value of ς , which is only known to exist.

From (4.48), it can be seen that the transient term $\|\Phi_e(0) - \Phi_0\|_{\infty} e^{-ct}$ decays to zero at the rate of the bandwidth of the low-pass filter. Increasing c implies that the transient time decreases. Furthermore, the final estimation error is bounded by the term

$$\frac{\gamma_c}{\sqrt{\Gamma_c}} + \|1 - C(s)\|_{\mathcal{L}_1} \|\Phi(t)\|_{\mathcal{L}_{\infty}},$$

which can be made arbitrarily small by increasing c . In general, increasing c implies that γ_c increases, and the adaptation gain Γ_c must also be increased to keep the term $\gamma_c/\sqrt{\Gamma_c}$ small. Increasing the adaptation gain Γ_c requires faster computation and requires a smaller integration step. If the Hurwitz matrix A_m has the special property of being a diagonal matrix, the following property holds:

Proposition 11 *If A_m is a diagonal matrix of the form*

$$A_m = \text{diag} \left(A_{m_1}, A_{m_2} \right),$$

where A_{m_1}, A_{m_2} are negative constants, then γ_c is independent of c .

Proof. By definition of γ_c in (4.49), $\|C(s)(s\mathbb{I}_2 - A_m)^{-1}\|_{\mathcal{L}_1}$ is the only component of γ_c that

varies with changing c . This can be computed as

$$\|C(s)(s\mathbb{I}_2 - A_m)^{-1}\|_{\mathcal{L}_1} = \left\| \frac{c}{s+c} \begin{bmatrix} \frac{1}{s-A_{m_1}} & 0 \\ 0 & \frac{1}{s-A_{m_2}} \end{bmatrix} \right\|_{\mathcal{L}_1} \quad (4.52)$$

which is a 2-input, 2-output system of stable proper transfer functions. The \mathcal{L}_1 gain is given by

$$\left\| \frac{c}{s+c} \begin{bmatrix} \frac{1}{s-A_{m_1}} & 0 \\ 0 & \frac{1}{s-A_{m_2}} \end{bmatrix} \right\|_{\mathcal{L}_1} = \max_{i=1,2} \left\| \frac{c}{(s+c)(s-A_{m_i})} \right\|_{\mathcal{L}_1} \quad (4.53)$$

$$\begin{aligned} &= \left\| \frac{c}{(s+c)(s+\min(|A_{m_1}|, |A_{m_2}|))} \right\|_{\mathcal{L}_1} \quad (4.54) \\ &= \int_0^\infty \left| \mathcal{L}^{-1} \left\{ \frac{c}{(s+c)(s+\min(|A_{m_1}|, |A_{m_2}|))} \right\} \right| dt, \end{aligned}$$

where $\mathcal{L}^{-1}\{\cdot\}$ denotes the inverse Laplace transform. Using partial fraction decomposition and taking the integral, (4.53) becomes

$$\begin{aligned} &\|C(s)(s\mathbb{I}_2 - A_m)^{-1}\|_{\mathcal{L}_1} \\ &= \left| \left(\frac{e^{-ct}}{c - \min(|A_{m_1}|, |A_{m_2}|)} - \frac{ce^{-\min(|A_{m_1}|, |A_{m_2}|)t}}{\min(|A_{m_1}|, |A_{m_2}|)(c - \min(|A_{m_1}|, |A_{m_2}|))} \right) \right|_0^\infty \\ &= \left| 0 - \frac{1}{c - \min(|A_{m_1}|, |A_{m_2}|)} + \frac{c}{\min(|A_{m_1}|, |A_{m_2}|)(c - \min(|A_{m_1}|, |A_{m_2}|))} \right| \\ &= \frac{1}{\min(|A_{m_1}|, |A_{m_2}|)}. \end{aligned}$$

□

As a consequence of Proposition 11, increasing the bandwidth c does not lead to an increase of γ_c , so there is no need to compensate for the change in γ_c by increasing Γ_c . The performance of the estimator can be improved arbitrarily without tuning the adaptation gain Γ_c . Figures 4.3 - 4.6 illustrate tuning of the IBO and fast estimator, respectively.

Figures 4.3 - 4.4 show the estimation parameters $\hat{z}(t)$, $\hat{L}(t)$, $\hat{\theta}(t)$ for a single obstacle configuration located at $(P_x, P_y) = (5, 1)$ whose radius is $L = 0.5$ m. The parameter $\hat{\theta}(t)$ is chosen to lie within the bounds $\hat{\theta}(t) \in [0.4, 5]$ so that $\hat{L}(t) \in [0.2, 2.5]$. The path is generated by the

polynomial path generation method in Chapter 2 with the initial conditions

$$\begin{aligned}x(0) &= 0, \quad y(0) = 0, \quad x(t_f) = 40, \quad y(t_f) = 0, \\v(0) &= 1, \quad v(t_f) = 1, \quad a(0) = 0, \quad a(t_f) = 0, \\ \psi(0) &= 0^\circ, \quad \psi(t_f) = 15^\circ,\end{aligned}$$

where $t_f = 3$. The IBO parameters are chosen as

$$A_m = \text{diag}(-25, -25), Q = \text{diag}(5, 5), G = 30.$$

The fast estimator parameters are

$$\Gamma_c = 200,000, c = 1000,$$

and the initial estimate is $\hat{\theta}(0) = 0.4$. In Figure 4.3, both estimators converge to the true parameters given the same set of initial conditions. For the chosen estimator parameters, The IBO converges exponentially after 0.6 s, while the fast estimator converges after 0.4. Although the transient time of the fast estimator is less than that of the IBO for this case, the performance of the fast estimator is guaranteed only to be bounded. The estimation $\hat{\theta}(t)$ deviates within a small envelope around the true value as compared to the IBO estimate (Figure 4.4). The magnitude of this bound is determined by the value of the bandwidth of the low-pass filter c .

Figures 4.5 and 4.6 show the effect of G on the performance of the IBO. Increasing G appears to increase the exponential rate of convergence for this configuration as seen in Figures 4.5 and 4.6, where $\hat{L}_{IBO}(t)$ denotes the estimate $\hat{L}(t)$ of L generated by IBO. The amount of increase is undetermined by the theoretical results.

Figures 4.7, 4.8 show that performance of the fast estimator for three values of c . Since A_m is a diagonal Hurwitz matrix, Proposition 11 applies and the adaptation gain Γ_c is unchanged for all values of c . When $c = 20$, the error bound is large, which is shown in the blue dashed line in Figure 4.7. As c is increased to $c = 200$, the bound is reduced as seen by the green dash-dot line in Figure 4.7. Finally, when $c = 2000$ (red line), the bound is reduced

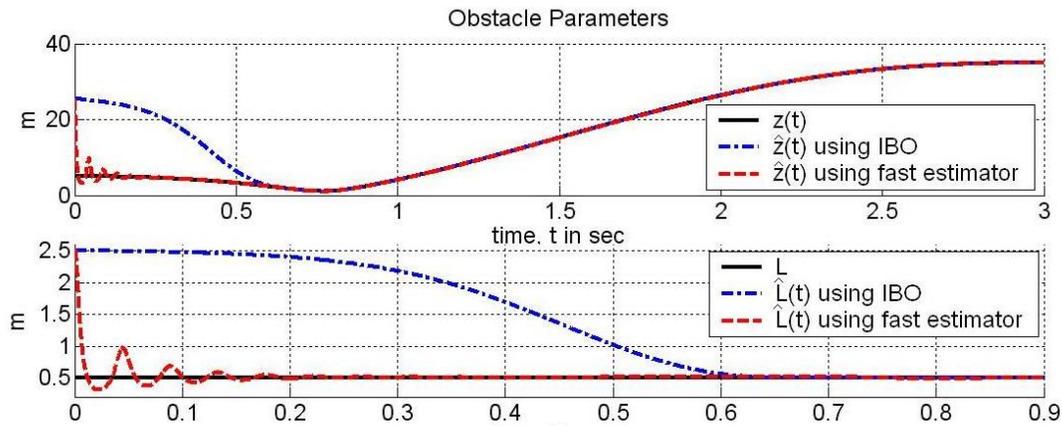


Figure 4.3: Estimation parameters using IBO and fast estimator.

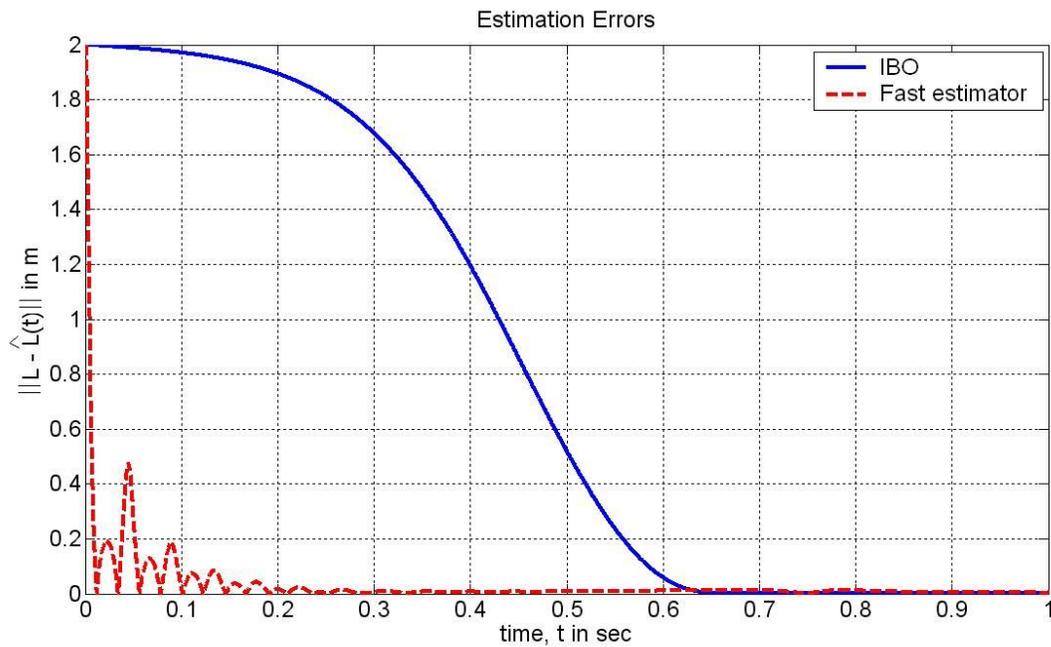
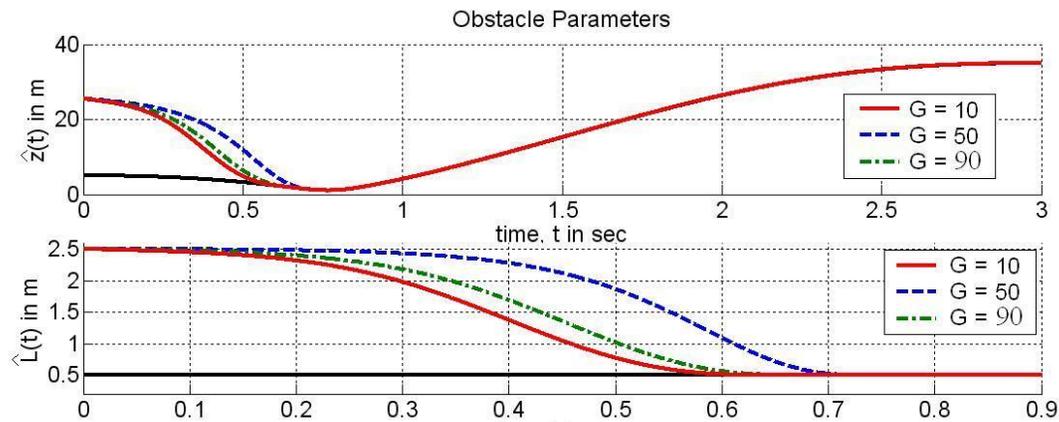
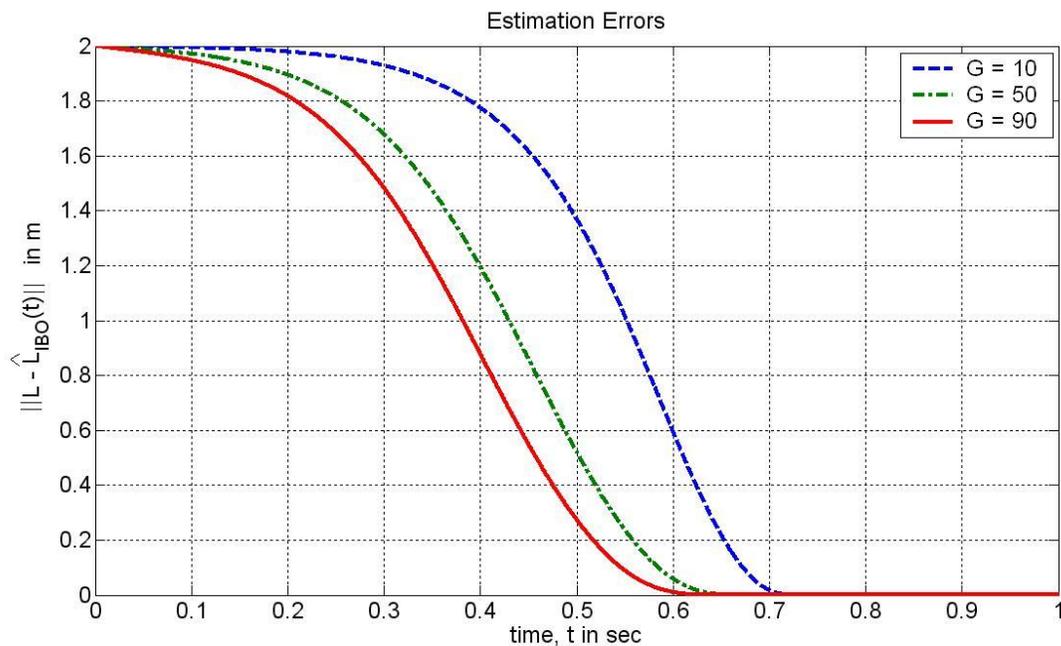


Figure 4.4: Estimation errors using IBO and fast estimator.

further. The error $L - \hat{L}_{FE}(t)$, where $\hat{L}_{FE}(t)$ denotes the estimate $\hat{L}(t)$ generated by the fast estimator, is shown in Figure 4.8. Increasing c decreases the error bound and the transience.

Figure 4.5: IBO performance for various values of G .Figure 4.6: Error of IBO for various values of G .

The advantage of the IBO over the fast estimator is the exponential property, which guarantees the estimation error continuously decrease. The fast estimator error is guaranteed only to be bounded and may not decrease beyond the bound (Figure 4.9). However, the

performance of the fast estimator can be tuned to exhibit comparable performance as the IBO. From Figure 4.8 and 4.6, the transient time for the IBO and fast estimator are similar when $G = 90$ and $c = 2000$. A zoomed in view of the estimation errors for $G = 90, c = 2000$ is shown in Figure 4.9. The fast estimator response error converges to less than 0.1 more quickly than the IBO for large values of G (0.2 s and 0.5 s, respectively).

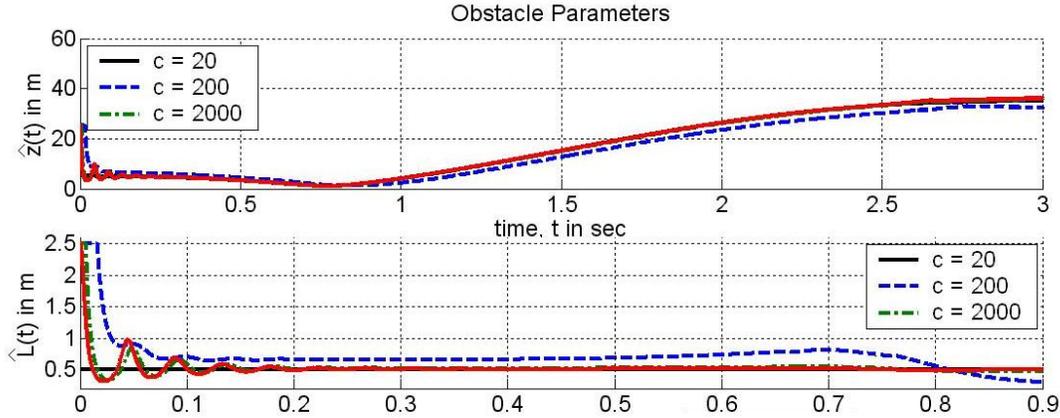


Figure 4.7: Fast estimator performance for various values of c .

Next, Figures 4.10 - 4.11 show the estimation parameters $\hat{z}(t), \hat{L}(t), \hat{\theta}(t)$ for a single obstacle configuration located at $(P_x, P_y) = (3, 3)$ whose radius is $L = 1$ m. The upper and lower bounds on $\hat{\theta}(t)$ and $\hat{L}(t)$ are the same as in the previous case. Initial conditions for the path are

$$\begin{aligned} x(0) &= 0, \quad y(0) = 0, \quad x(t_f) = 20, \quad y(t_f) = 10, \\ v(0) &= 1, \quad v(t_f) = 1, \quad a(0) = 0, \quad a(t_f) = 0, \\ \psi(0) &= 10^\circ, \quad \psi(t_f) = 0^\circ, \end{aligned}$$

where $t_f = 3$. The IBO parameters are chosen as

$$A_m = \text{diag}(-25, -25), \quad Q = \text{diag}(5, 5), \quad G = 10.$$

The fast estimator parameters are

$$\Gamma_c = 300,000, \quad c = 2000,$$

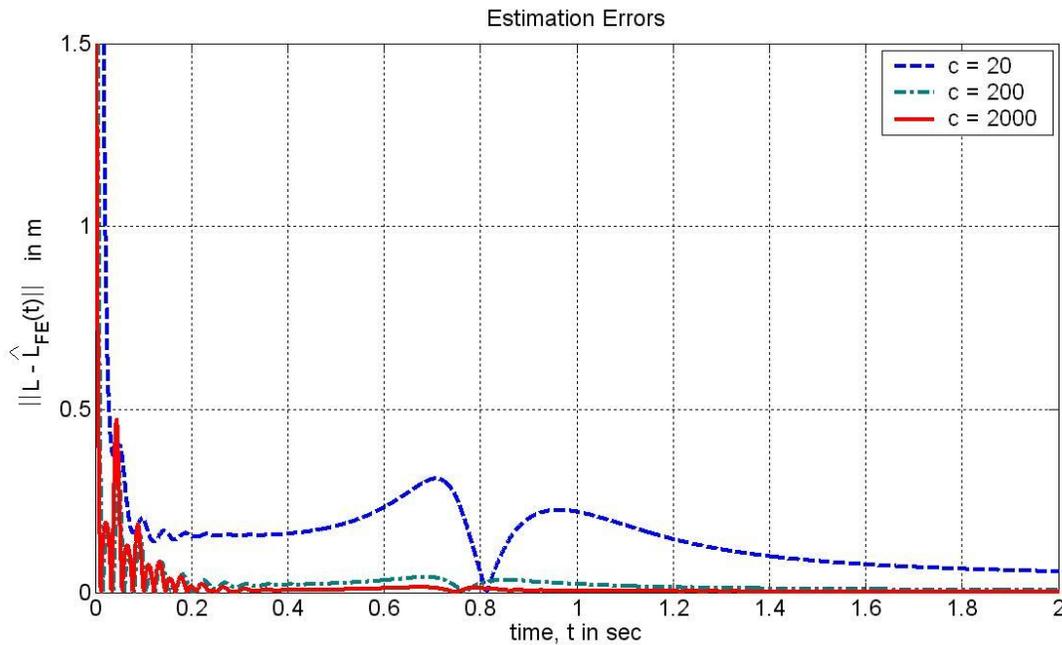


Figure 4.8: Error of fast estimator for various values of c .

and the initial estimate is $\hat{\theta}(0) = 0.4$. In Figure 4.10, both estimators again converge to the true parameters. The IBO converges exponentially after 0.7 s, while the fast estimator converges after 0.4. Figure 4.11 shows the estimator errors. The fast estimator converges to within 0.1% error quickly and remains within a small bound which is determined by the value of the bandwidth of the low-pass filter c .

Figures 4.12 and 4.13 show the effect of G on the performance of the IBO for $G = [1 \ 10 \ 50]$. In Figure 4.12, increasing G again increase the exponential rate of convergence for this configuration. The amount of increase or guidelines for choosing G is undetermined by the theoretical results.

Finally, Figures 4.14, 4.15 show the performance of the fast estimator for three values of c . Since A_m is a diagonal Hurwitz matrix, Proposition 11 applies and the adaptation gain Γ_c is unchanged for all values of c . The case when $c = 20$, is shown in the blue dashed line in Figure 4.7. As c is increased to $c = 200$, the bound is reduced as seen by the green

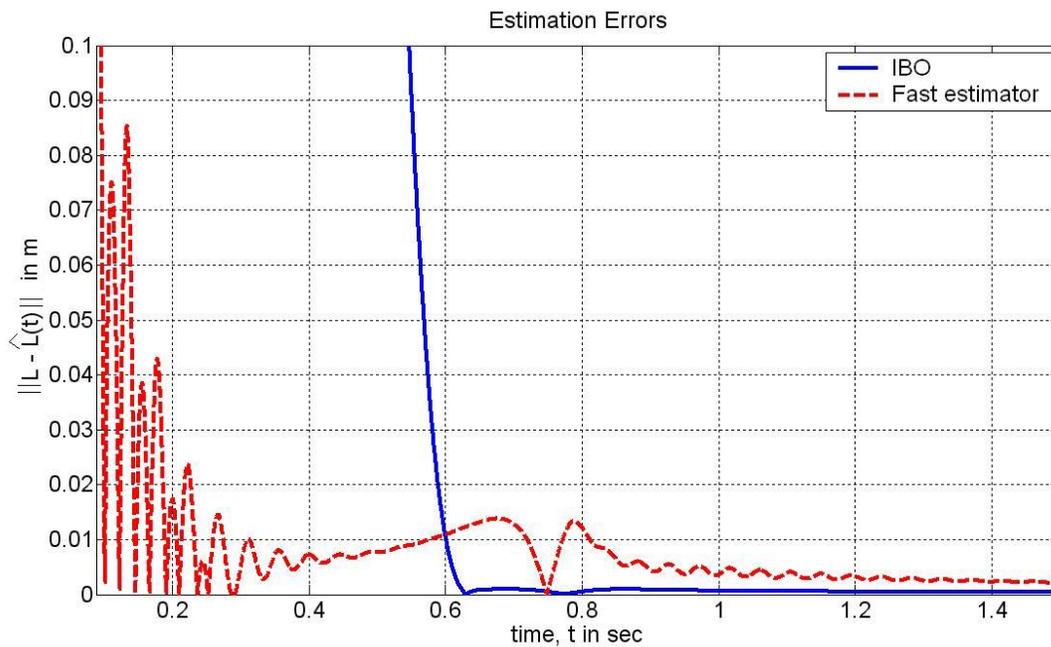


Figure 4.9: Zoomed errors of estimators for $G = 90$, $c = 2000$.

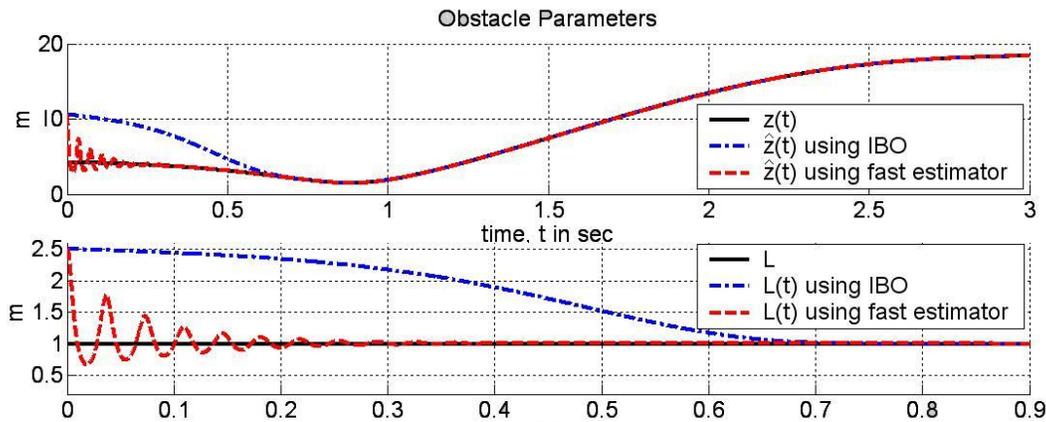


Figure 4.10: Estimation parameters using IBO and fast estimator, second obstacle configuration.

dash-dot line in Figure 4.7. When $c = 2000$ (red line), the bound is reduced further. The error $L - \hat{L}_{FE}(t)$ is shown in Figure 4.8. Increasing c decreases the error bound and the

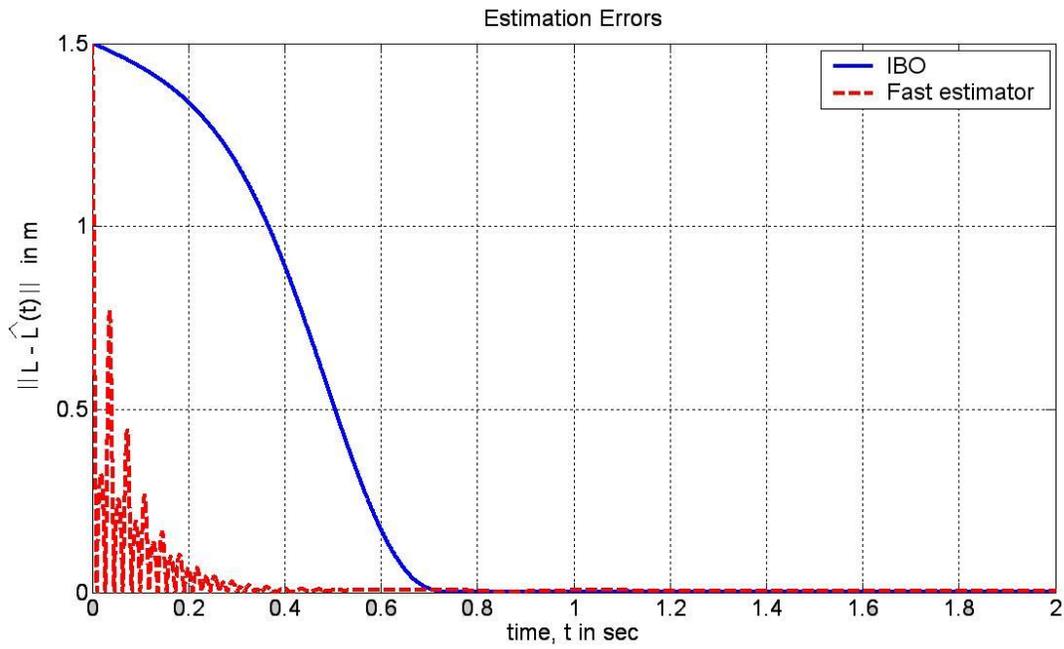


Figure 4.11: Estimation errors using IBO and the fast estimator, second obstacle configuration.

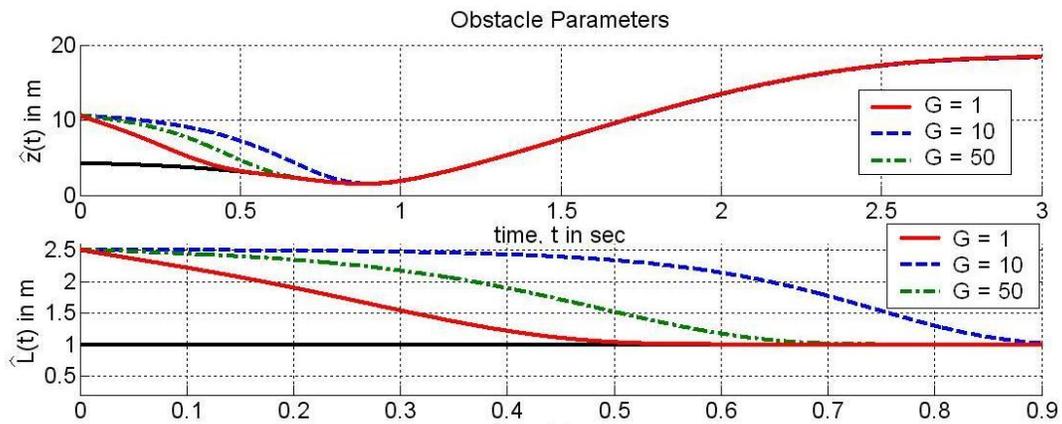


Figure 4.12: IBO performance for various values of G , second obstacle configuration.

transience.

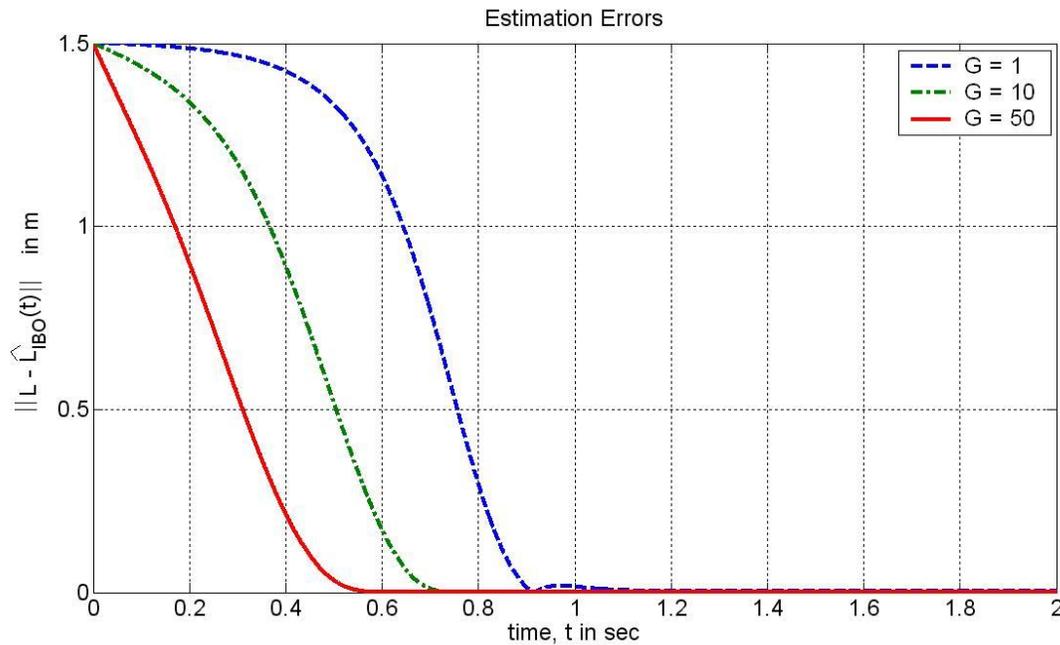


Figure 4.13: Error of IBO for various values of G , second obstacle configuration.

From Figure 4.15 and 4.13, the transient time for the IBO and fast estimator are similar when $G = 50$ and $c = 2000$. A zoomed in view of the estimation errors for $G = 50, c = 2000$ is shown in Figure 4.16. The fast estimator response error converges to less than 0.1 more quickly than the IBO for large values of G (0.2 s and 0.5 s, respectively).

Remark 18 *The fast estimation algorithm has been applied to estimate a ground target's unknown velocity and was tested using experimental data collected during flight tests of an aerial vehicle following a ground target with its onboard camera. Comparison with an earlier developed Linear Parametrically-Varying (LPV) filter in Ref. [36] showed significant improvement of the fast estimator over the LPV filter [62].*

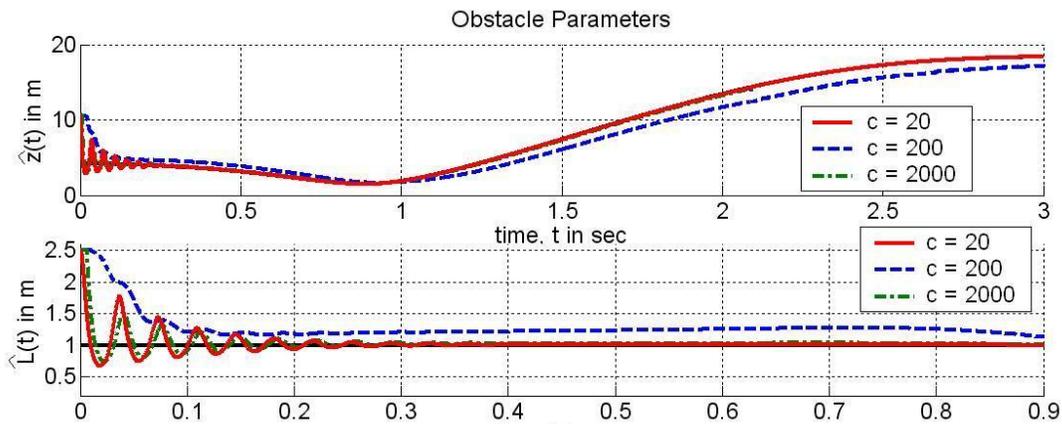


Figure 4.14: Fast estimator performance for various values of c , second obstacle configuration.

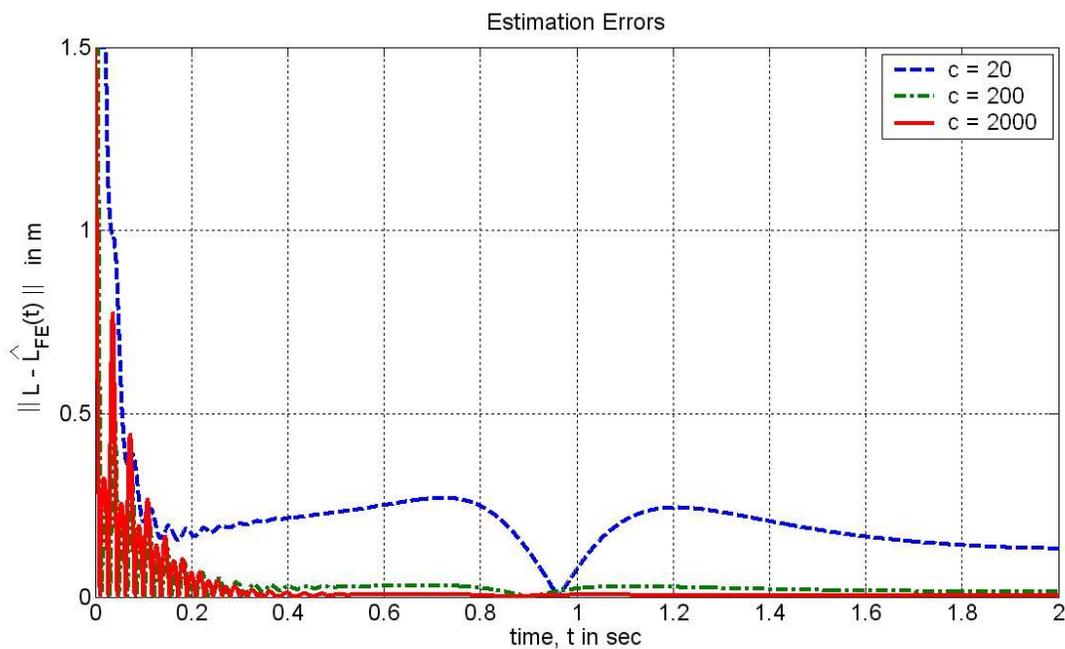


Figure 4.15: Error of the fast estimator for various values of c , second obstacle configuration.

4.5 Summary

Two estimators for approximating the unknown obstacle size are reviewed. The problem is formulated for vision-based obstacle avoidance for a mobile robot system and both es-

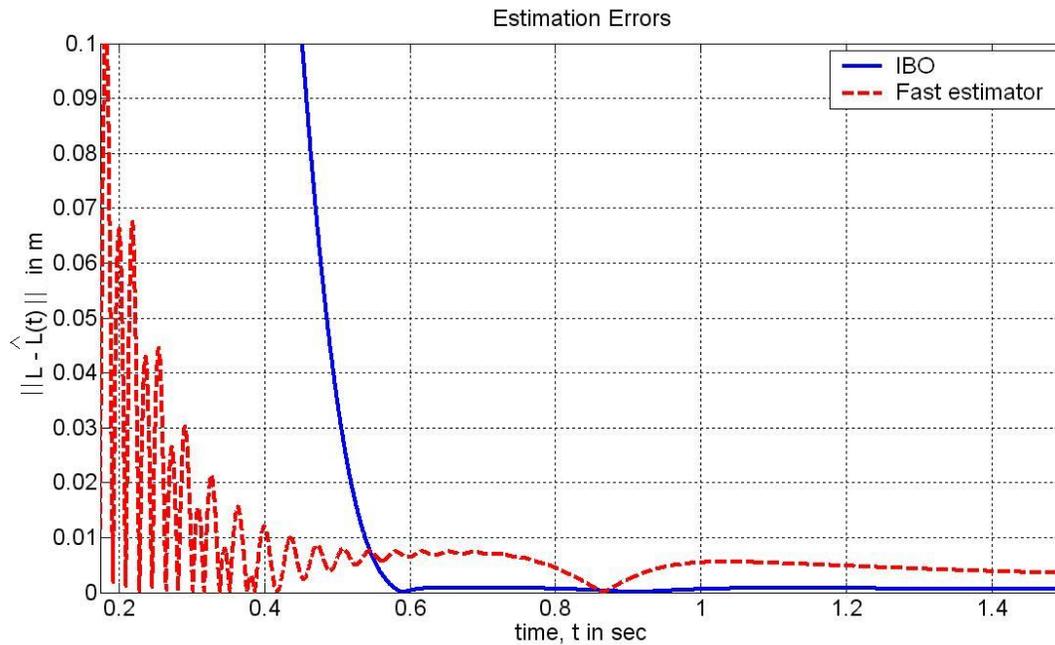


Figure 4.16: Zoomed errors of estimators for $G = 50$, $c = 2000$, second obstacle configuration.

timators were applied for this configuration. An analysis is made on the performance of the estimators, and their effect on the obstacle avoidance. An explicit relationship between the performance and the estimation parameters can be made on the fast estimator while no such relationship exists for the IBO. While the IBO is guaranteed to decrease the estimation error at an exponential rate, the fast estimator is guaranteed only to be bounded. The final precision error can be made arbitrarily small by proper choice of the estimation parameters. Simulations illustrate the theoretical findings.

Chapter 5

Multi-Vehicle Coordination for Simultaneous Arrival

Having solved the complete path following problem with obstacle avoidance for a single vehicle and an arbitrary speed profile, this chapter presents a solution for time-coordinated control of multiple vehicles. Applications for time-critical coordination include situations where all vehicles must rendezvous at their final destinations at the same time or at different times so as to meet a desired inter-vehicle arrival schedule. The latter problem of meeting an inter-vehicle arrival schedule is equivalent to the problem of simultaneous arrival with scaling. Hence, the problem of simultaneous arrival is considered without loss of generality. Let t_f denote the final arrival time and let $Q_i(t)$ denote the location of the i th vehicle's center of mass at time t . Let $s_i(t)$ denote the arc length of the path from the origin to $Q_i(t)$ at time t . Note that t_f is not fixed and varies depending on unforeseen events such as obstacles. The value of t_f will be dictated by the coordination solution below.

The vehicles exchange motion-related information based on a communication topology limited by bandwidth, range of communications, energy minimization, and possible interferences. In general, no vehicle will be able to communicate with the entire fleet due to these constraints. In order to capture the communication constraints for which control strategies

are designed, tools from Algebraic Graph Theory are used. This approach is common in the literature [22, 40, 26, 27, 28, 29, 30, 42, 43, 44] to describe the type of network available for communications and determine the performance that can be achieved with different coordination strategies. Section 5.1 will present some of the general ideas of Graph Theory and the important results relevant to this dissertation. In Section 5.2, a leader-follower solution will be presented for the case when communication between vehicles is *bi-directional*. That is, if vehicle i can communicate with vehicle j , then vehicle j can communicate with vehicle i . For this derivation, a leader is chosen whose desired speed profile is learned by the other vehicles. This desired speed profile is adjusted during the mission in order to accommodate “deviations from the plan.” That is, the leader’s speed profile can speed up or slow down according to the positions of the other vehicles on their respective paths. While this method is robust to delays and uncertainties by adjusting the leader’s desired speed profile, the mission success is dependent on the leader vehicle and will fail with loss or damage to the leader. Section 5.3 presents a solution for the case when communication between vehicles is *uni-directional* (i.e. vehicle i communicates with vehicle j does not imply vehicle j communicates with vehicle i). In the uni-directional framework, some vehicles only send information and others only receive information. For this framework, the leader is chosen as a “virtual” vehicle, which implies increased robustness to loss of vehicles. That is, the mission will not fail if any one vehicle is lost as long as the communication topology between the vehicles satisfies certain assumptions. The trade-off to this benefit is that all vehicles must know a reference speed profile *a priori*, which is fixed during the mission. The vehicles will coordinate to meet the time-critical mission objective using this reference velocity profile. In the context of Graph theory, *undirected* graphs are specially suited to model bi-directional communication networks. To model uni-directional communications networks, *directed* graphs must be used. The relevant concepts of Graph Theory are presented next.

5.1 Graph Theory: Review of Important Concepts

This section summarizes some key concepts and results of graph theory that are crucial in developing the coordination control laws of this dissertation. The definitions and theorems in this section borrow from [4], [7], and [33] for undirected graphs, and from [5], [33], [22], and [40] for directed graphs.

Undirected graphs

An *undirected graph* or simply a *graph* $\Gamma(\mathcal{V}, \mathcal{E})$ (abbrev. Γ) consists of a set of *vertices* $\mathcal{V}_i \in \mathcal{V}(\Gamma)$ and a set of *edges* $\mathcal{E}(\Gamma)$, where an edge $\{\mathcal{V}_i, \mathcal{V}_j\}$ is an unordered pair of distinct vertices \mathcal{V}_i and \mathcal{V}_j in $\mathcal{V}(\Gamma)$. A *simple graph* is a graph with no edges from one vertex to itself. The graphs considered in this dissertation are simple graphs, which will be henceforth referred to as graphs. To model vehicle communication, the vertices and the edges of a graph represent the vehicles themselves and the data links among them, respectively. If $\{\mathcal{V}_i, \mathcal{V}_j\} \in \mathcal{E}(\Gamma)$, then we say that \mathcal{V}_i and \mathcal{V}_j are adjacent or *neighbors*. The set of neighbors of vehicle i is denoted by \mathcal{N}_i . A *path* of length r from \mathcal{V}_i to \mathcal{V}_j in a graph is a sequence of $r + 1$ distinct vertices starting with \mathcal{V}_i and ending with \mathcal{V}_j , such that any two consecutive vertices are adjacent. The graph Γ is said to be *connected* if any two arbitrary vertices \mathcal{V}_i and \mathcal{V}_j can be joined by a path of finite length. An *orientation* of a graph Γ is the assignment of a direction to each edge of that graph. That is, for each edge $\{\mathcal{V}_i, \mathcal{V}_j\}$ in $\mathcal{E}(\Gamma)$, one of the $\mathcal{V}_i, \mathcal{V}_j$ is selected as the *head* of the edge and the other the *tail*, and view the edge oriented from its tail to its head. After this operation, the elements of $\mathcal{E}(\Gamma)$ are ordered pairs $(\mathcal{V}_i, \mathcal{V}_j)$, henceforth known as *arcs*. Figure 5.1 (a) shows the case of an undirected graph with three vertices and Figure 5.1 (b) shows a graph with an associated orientation. Formally, an orientation of Γ is defined as a function Σ from the arcs of Γ to $\{-1, 1\}$:

$$\Sigma(\{\mathcal{V}_i, \mathcal{V}_j\}) = +1 \quad \text{if edge } (\{\mathcal{V}_i, \mathcal{V}_j\}) \text{ is oriented from tail } \mathcal{V}_i \text{ to head } \mathcal{V}_j,$$

$$\Sigma(\{\mathcal{V}_i, \mathcal{V}_j\}) = -1 \quad \text{if edge } (\{\mathcal{V}_i, \mathcal{V}_j\}) \text{ is oriented from tail } \mathcal{V}_j \text{ to head } \mathcal{V}_i.$$

It follows that $\Sigma(\{\mathcal{V}_i, \mathcal{V}_j\}) = -\Sigma(\{\mathcal{V}_j, \mathcal{V}_i\})$. For example, in Figure 5.1, $\Sigma(V1, V2) = +1$ and

$\Sigma(V2, V1) = -1$. An *oriented graph* is a graph with a particular orientation denoted Γ^Σ . The *incidence matrix* \mathbb{M} of Γ^σ is the matrix with elements in the set $\{-1, 0, +1\}$ with rows and columns indexed by the vertices and the arcs of Γ^Σ , respectively. If Γ^Σ has M vertices and E arcs, \mathbb{M} is of order $M \times E$ and its kl -entries are

$$m_{kl} = \begin{cases} +1, & \text{if } \mathcal{V}_k \text{ is the head of arc } l, \\ -1, & \text{if } \mathcal{V}_k \text{ is the tail of arc } l, \\ 0, & \text{otherwise,} \end{cases}$$

where $l \in \{1, 2, \dots, E\}$. Each column of \mathbb{M} contains only two non-zero entries, $+1$ and -1 , representing the head and the tail of the specific arc. Given an undirected graph Γ , its *degree matrix* \mathbb{D} is defined as the diagonal matrix of order M with $|N_i|$ (the cardinality of the set of neighbors of vehicle i , N_i) in the ii -entry. The *adjacency matrix* \mathbb{A} is a square matrix with rows and columns indexed by the vertices, such that the ij -entry of \mathbb{A} is 1 if $\{\mathcal{V}_i, \mathcal{V}_j\} \in \mathcal{E}$ and zero otherwise.

Lemma 3 [7] *Let \mathbb{D} and \mathbb{A} be the degree matrix and adjacency matrix of an undirected graph Γ , respectively. If Σ is an arbitrary orientation of Γ and \mathbb{M} is the incidence matrix of Γ^Σ , then*

$$\mathbb{M}\mathbb{M}^\top = \mathbb{D} - \mathbb{A}.$$

As an example, consider the graph depicted in Figure 5.1 (a) with the associated orientation given in Figure 5.1 (b). For this example, $M = 3$ and $E = 2$. It can be verified that

$$\mathbb{M} = \begin{bmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbb{D} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \text{and } \mathbb{A} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Hence,

$$\mathbb{M}\mathbb{M}^\top = \mathbb{D} - \mathbb{A} = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

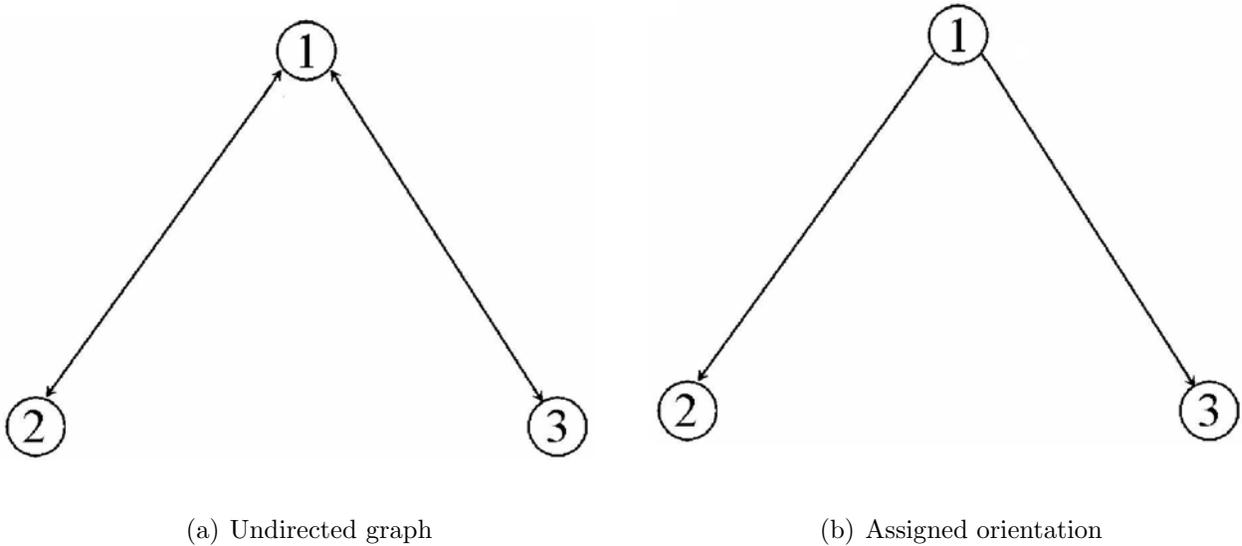


Figure 5.1: Undirected graph and an associated orientation.

Next, the concept of Laplacian of an undirected graph is introduced [7]. Let Γ be an undirected graph with M vertices and assign an arbitrary orientation Σ to it. Consider the corresponding incidence matrix \mathbb{M} . The *Laplacian* L of Γ is the symmetric, positive semi-definite square matrix $L = \mathbb{M}\mathbb{M}^\top$ of order $M \times M$. This definition of the Laplacian is equivalent to the more common form $L = \mathbb{D} - \mathbb{A}$ as a result of Lemma 3. By construction, L is independent of the particular orientation assigned to an undirected graph Γ . Furthermore, all eigenvalues of L are non-negative and $L1_M = 0_M$. If Γ is connected, $\text{Rank}(L) = M - 1$. Consequently, L has a single eigenvalue at zero with corresponding right eigenvector 1_M . Finally, the diagonal elements of the Laplacian matrix of a connected graph are positive and its off diagonals are non-positive. The properties of the Laplacian are summarized (see Ref. [30] and the references therein):

$$L \in \mathbb{R}^{M \times M}, \quad L \geq 0, \quad \text{rank}(L) = M - 1, \quad L1_m = 0_M, \quad L = L^\top. \quad (5.1)$$

The following result holds for a connected graph Γ with M vertices and E edges, the proof of which is given in [30] pp. 77.

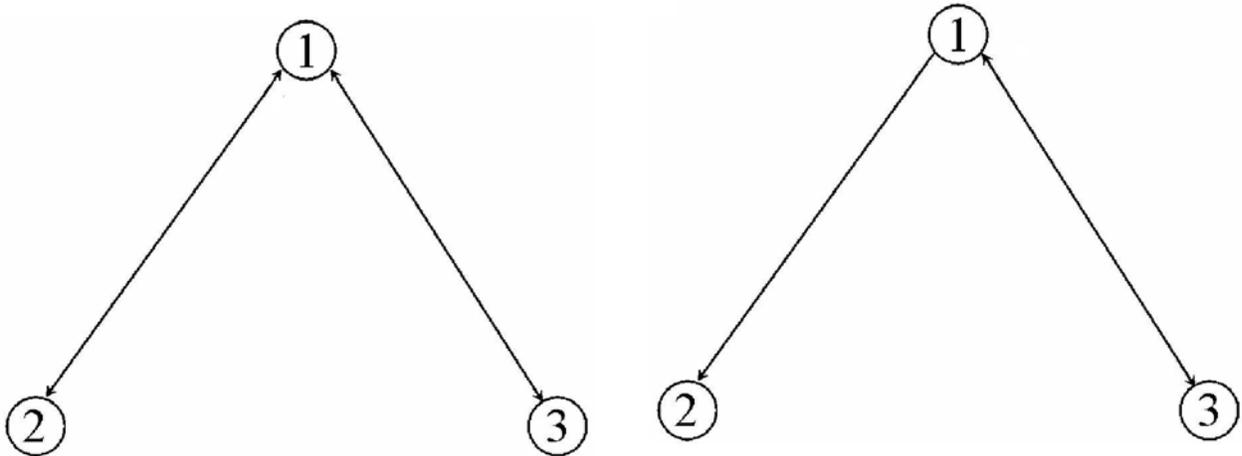
Lemma 4 [30] *If Γ is connected, there exists an orthonormal full rank incidence matrix $U \in \mathbb{R}^{M \times (M-1)}$ and a diagonal real matrix $L_d \in \mathbb{R}^{M \times M} > 0$ with the non-zero eigenvalues of L on its diagonal such that:*

$$\begin{aligned} \left[\frac{1_M}{\sqrt{M}} \quad U \right]^\top L \left[\frac{1_M}{\sqrt{M}} \quad U \right] &= \begin{bmatrix} 0 & 0 \\ 0 & L_d \end{bmatrix}, \\ U^\top 1_M &= 0_{M-1}, \quad U^\top U = \mathbb{I}_{M-1}, \quad U L_D U^\top = L. \end{aligned} \quad (5.2)$$

As an illustration, Figure 5.2 (a) shows an example of an undirected graph. For the graph depicted in Figure 5.2 (a), one can verify that the solutions

$$L = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} -\frac{2}{\sqrt{6}} & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \end{bmatrix}, \quad L_d = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}, \quad (5.3)$$

satisfy the properties in (5.1) and (5.2).



(a) Undirected graph

(b) Directed graph

Figure 5.2: Undirected and directed graphs: an example.

Directed Graphs

Theory for directed graphs are similar to the ideas of undirected graphs, with a few notable differences. A *directed graph* or *digraph* $\Gamma(\mathcal{V}, \mathcal{E})$ consists of a finite set \mathcal{V} of vertices $\mathcal{V}_i \in \mathcal{V}$

and a finite set \mathcal{E} of ordered pairs or *arcs* $(\mathcal{V}_i, \mathcal{V}_j) \in \mathcal{E}$. Given an arc $(\mathcal{V}_i, \mathcal{V}_j) \in \mathcal{E}$, its first and second elements are called the *tail* and *head* of the arc, respectively. For the purposes of this dissertation, the vertices of a graph represent vehicles and the arcs the uni-directional data links among them. The flow of information in an arc is defined to be directed from its head to its tail. The *in-degree* (*out-degree*) of a vertex \mathcal{V}_i is the number of arcs with \mathcal{V}_i as its head (tail). If $(\mathcal{V}_i, \mathcal{V}_j) \in \mathcal{E}$, then we say that \mathcal{V}_i is adjacent to \mathcal{V}_j . Recall the definition of an *oriented graph*, which is an undirected graph with an orientation assigned to each of its edges. A difference between a digraph and an oriented graph is that if \mathcal{V}_i and \mathcal{V}_j are vertices, a digraph allows both $(\mathcal{V}_i, \mathcal{V}_j)$ and $(\mathcal{V}_j, \mathcal{V}_i)$ as arcs, while only one is permitted in an oriented graph. In a digraph, the directions are fixed and imposed by the communication topology under consideration. In contrast for an oriented graph, the underlying graph is fixed, while the orientation may vary. A *path* of length r from \mathcal{V}_i to \mathcal{V}_j in a digraph is a sequence of $r + 1$ distinct vertices starting with \mathcal{V}_i and ending with \mathcal{V}_j such that consecutive vertices are adjacent. If there is a path in Γ from vertex \mathcal{V}_i to vertex \mathcal{V}_j , then \mathcal{V}_j is said to be *reachable* from \mathcal{V}_i . In this case, there is a path of consecutive communication links directed from vehicle j (transmitter) to vehicle i (receiver). Vertex \mathcal{V}_i is *globally reachable* if it is reachable from every other vertex in the digraph. The *adjacency matrix* of a digraph Γ is denoted \mathbb{A} and is a square matrix with rows and columns indexed by the vertices, such that the ij -entry of \mathbb{A} is 1 if $(\mathcal{V}_i, \mathcal{V}_j) \in \mathcal{E}$ and zero otherwise. The degree matrix \mathbb{D} of a digraph Γ is a diagonal matrix where the ii -entry equals the out-degree of vertex \mathcal{V}_i . The *Laplacian matrix* of a digraph is defined as

$$L = \mathbb{D} - \mathbb{A}.$$

The following Lemma plays a key role in the development that follows.

Lemma 5 [30] *The Laplacian matrix of a digraph with at least one globally reachable vertex has a simple eigenvalue at 0 with right eigenvector 1. All the other eigenvalues have positive real parts.*

From the above Lemma, the following properties can be derived for F_1, F_2, L_{11} , and $F =$

$[1_M \ F_1]$ [30]:

$$F_2 F_1 = \mathbb{I}_{M-1}, \quad F_2 1_M = 0_{M-1}, \quad \nu^\top F_1 = 0_{M-1}^\top, \\ F^{-1} = \begin{bmatrix} \nu^\top \\ F_2 \end{bmatrix}, \quad F^{-1} L F = \begin{bmatrix} 0 & 0_{M-1}^\top \\ 0_{M-1} & L_{11} \end{bmatrix}, \quad (5.4)$$

where ν^\top is the left eigenvector of L corresponding to the zero eigenvalue. To better illustrate the concepts of F, F_1, F_2, ν , and L_{11} , consider the example given in Figure 5.2 (b), where communication between vehicles 1 and 3 is bi-directional and communication between vehicles 1 and 2 is uni-directional.

The Laplacian is computed to be

$$L = \begin{bmatrix} 2 & -1 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \quad (5.5)$$

and the left eigenvector of L corresponding to the zero eigenvalue is

$$\nu^\top = [0 \ 1 \ 0].$$

Since $\nu^\top F_1 = 0_2^\top$, one choice of F_1 is

$$F_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}. \quad (5.6)$$

Similarly, one can choose F_2 as

$$F_2 = \begin{bmatrix} 1 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, \quad (5.7)$$

and verify that

$$L_{11} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix} \quad (5.8)$$

satisfies the properties in (5.4).

Remark 19 The decomposition (5.14) is a more general case than the decomposition associated with the undirected case in (5.2), corresponding to $F_2 = U^\top$, $F_1 = U$, and $L_{11} = L_d$. The matrix L_{11} is no longer diagonal as compared to L_d , and does not, in general, have the same eigenvalues as L .

5.2 Time-Critical Coordination With Bi-Directional Communication

In this section, vehicle communication is assumed to be bi-directional. A leader is elected whose desired speed profile is known only to itself. This speed profile is adjusted according to the location of the other vehicles along their paths. That is, if one or more vehicles are lagging behind, the leader will slow down its speed profile. Similarly, if the vehicles are traveling along their respective paths too quickly, the leader speed will be increased. Each vehicle learns the speed of the leader via communication amongst vehicles. The control law which will guarantee simultaneous arrival of all vehicles has a Proportional-Integral (PI) structure. Each vehicle learns the leader's speed profile online and adjusts their corresponding desired speed profiles, eliminating the need for all vehicles to know the leader's desired speed profile *a priori*.

Let s_{f_i} denote the total length of the i th spatial path as generated by the polynomial path generation method in Chapter 2. Define

$$\bar{s}_i(t) = \frac{s_i(t)}{s_{f_i}}.$$

If $\bar{s}_i(t_f) = 1$ for all $i = 1, 2, \dots, M$, where M is the total number of vehicles, then all vehicles arrive at their final destinations at the same time. Taking the time derivative of $\bar{s}_i(t)$ and substituting the virtual control law $\dot{s}_{PF}(t)$ in (2.89c) gives

$$\dot{\bar{s}}_i(t) = \frac{\dot{s}(t)}{s_{f_i}} = \frac{v_i(t) \cos \psi_{e_i}(t) + k_1 x_{e_i}(t)}{s_{f_i}}, \quad (5.9)$$

where k_1 is not indexed for notational simplicity. The coordination in time will be achieved by defining a desired speed profile $v_{d_i}(t)$ via dynamic inversion for each vehicle such that the coordination objective is achieved. Define

$$u_{c_i}(t) = \frac{v_{d_i}(t) \cos \psi_{e_i}(t) + k_1 x_{e_i}(t)}{s_{f_i}},$$

where the subscript c of u_{c_i} denotes coordination. Given $u_{c_i}(t)$, the desired velocity profile is computed as

$$v_{d_i}(t) = \frac{s_{f_i} u_{c_i}(t) - k_1 x_{e_i}(t)}{\cos \psi_{e_i}(t)}, \quad i = 1, 2, \dots, M.$$

It follows that

$$\dot{\bar{s}}(t) = u_c(t),$$

where $\dot{\bar{s}}(t) = [\dot{\bar{s}}_1(t) \ \dot{\bar{s}}_2(t) \ \dots \ \dot{\bar{s}}_M(t)]^\top$ and $u_c(t) = [u_{c_1}(t) \ u_{c_2}(t) \ \dots \ u_{c_M}(t)]^\top$. The coordination problem is reduced to choosing $u_c(t)$ such that $\bar{s}(t_f) = 1_M$, where 1_M is the $M \times 1$ vector with every element equal to 1. To account for communication between vehicles, Algebraic Graph Theory is used to model communication constraints (See Section 5.1). Let L denote the Laplacian of a connected undirected graph Γ that captures the underlying bidirectional communication network of the vehicle formation. In particular, the graph specifies for each vehicle the set of vehicles that it can exchange information with. Using the results from Lemma 4, there exist matrices $U \in \mathbb{R}^{M \times (M-1)}$ and $L_d \in \mathbb{R}^{M \times M}$ that satisfy the properties in (5.2). The coordination control law will exploit the properties of L , U , and L_d for the purposes of achieving the time-critical mission objective of simultaneous arrival of all vehicles.

Let

$$e_c(t) = U^\top \bar{s}(t)$$

denote the error vector. Using the property that $U^\top 1_M = \mathbf{0}_{M-1}$, it follows that $e_c(t) = \mathbf{0}_{M-1}$ if and only if $\bar{s}_1(t) = \bar{s}_2(t) = \dots = \bar{s}_M(t)$. This is well known in the literature on cooperative control as an Agreement Problem [22], [40]. By definition of $u_c(t)$, one can compute that

$$\dot{e}_c(t) = U^\top u_c(t). \quad (5.10)$$

Each vehicle communicates its coordination parameter $\bar{s}_i(t)$ with its neighbors according to the topology of the communications network, as expressed by the connected undirected graph Γ . Without loss of generality, elect vehicle 1 as the leader whose associated desired speed profile is $v_{d_1}(t)$. The coordination problem is further reduced to that of choosing a control law for $u_c(t)$ such that $e_c(t_f) = 0_{M-1}$, where the dynamics of $e_c(t)$ is given in (5.10). Lemma 6 defines such a control law.

Remark 20 *The problem formulation above leads to an inherently finite time horizon problem for the dynamics in (5.10). In this dissertation, the problem is solved indirectly using asymptotic analysis. Following a common practice, this can only be done approximately by judicious choice of the control gains and initial conditions. The solution to it is given below in the following Lemma, which is obtained in [43, 44] and [30].*

Lemma 6 *The control law for $u_c(t)$ chosen as*

$$u_c(t) = aL\bar{s}(t) + \begin{bmatrix} \frac{v_{d_1}(t)}{s_{f_1}} \\ \chi_I(t) \end{bmatrix} = C_1^\top \left(\frac{v_{d_1}(t)}{s_{f_1}} + aC_1L\bar{s}(t) \right) + C_2^\top (aC_2L\bar{s}(t) + \chi_I(t)) \quad (5.11)$$

$$\dot{\chi}_I(t) = bC_2L\bar{s}(t),$$

where a, b are negative scalars and

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 1 & 0_{M-1}^\top \\ 0_{M-1} & \mathbb{I}_{M-1} \end{bmatrix}, \quad (5.12)$$

solves the coordination problem for the dynamics in (5.10).

The proof of Lemma 6 requires the following two Lemmas, the proofs of which can be found in [43, 44].

Lemma 7 *Let A, B, C be positive definite matrices of the same dimensions with $A = A^\top$. Then the roots of $\det(\lambda^2A + \lambda B + C) = 0$ have negative real parts.*

Lemma 8 *The matrix $C_2U \in \mathbb{R}^{(M-1) \times (M-1)}$ has rank $M - 1$.*

The proof of Lemma 6 follows by defining

$$x_c(t) = \begin{bmatrix} \chi_I(t) - \frac{v_{d_1}}{s_{f_1}} \mathbf{1}_{M-1} \\ e_c(t) \end{bmatrix}$$

and showing that the closed-loop system defined in terms of $x_c(t)$ is exponentially stable using Lemmas 7 and 8. Details can be found in [43, 44].

Remark 21 *The control law in (5.11) has a PI structure, thus allowing each vehicle to learn the speed of the leader, rather than having it available a priori. In scalar form, the control law $u_c(t)$ can be written as*

$$\begin{aligned} u_{c_1}(t) &= \frac{v_{d_1}(t)}{s_{f_1}} + \sum_{j \in J_1} a(\bar{s}_1(t) - \bar{s}_j(t)), \\ u_{c_i}(t) &= \sum_{j \in J_i} a(\bar{s}_i(t) - \bar{s}_j(t)) + \chi_{I_i}(t), \\ \dot{\chi}_{I_i}(t) &= \sum_{j \in J_i} b(\bar{s}_i(t) - \bar{s}_j(t)), \end{aligned} \tag{5.13}$$

for $i = 2, \dots, n$, where J_i is the set of neighboring vehicles that vehicle i can exchange information with. Clearly, this implementation meets the communication constraints addressed in the coordination problem formulation. The gains a, b play the role of tuning knobs to adjust the speed of convergence of the coordination error $e_c(t)$ to 0. This is important in light of the comments made in Remark 20. Note that $u_{c_1}(t)$ depends on the difference between the coordination parameters of the leader and the other vehicles that the leader can communicate with, allowing for the leader to adjust its desired profile based on this information.

Figure 5.3 shows a simulation example with the communication topology depicted by the undirected graph Γ of Figure 5.2 (a) for three vehicles. The corresponding matrix parameters are given in (5.3) and the additional parameters are given by

$$\chi_I(0) = [8 \ 2]^\top, s_f = [30 \ 35 \ 25]^\top, a = -2, b = -5.$$

The leader's velocity is initiated at $v_1(0) = 1 \text{ m/s}$. The eigenvalues of the closed-loop system are

$$\lambda_1 = -5, \lambda_{2,3} = -1 \pm 2i, \lambda_4 = -1.$$

Figure 5.3 (top) shows the coordination parameter $\bar{s}(t)$, which converge to $\bar{s}_1(t) = \bar{s}_2(t) = \bar{s}_3(t)$ to satisfy the coordination objective. Figure 5.3 (bottom) shows the corresponding coordination error $e_c(t)$ which converges to zero for vehicles 2 and 3. Note that there is no coordination error for vehicle 1 since vehicle 1 is the leader whose speed profile must be learned by the other vehicles. Finally, Figure 5.4 shows the desired speed profile of each vehicle. The maximum allowable speed for the desired profiles are 5 m/s . The leader's speed profile is shown by the blue solid line.

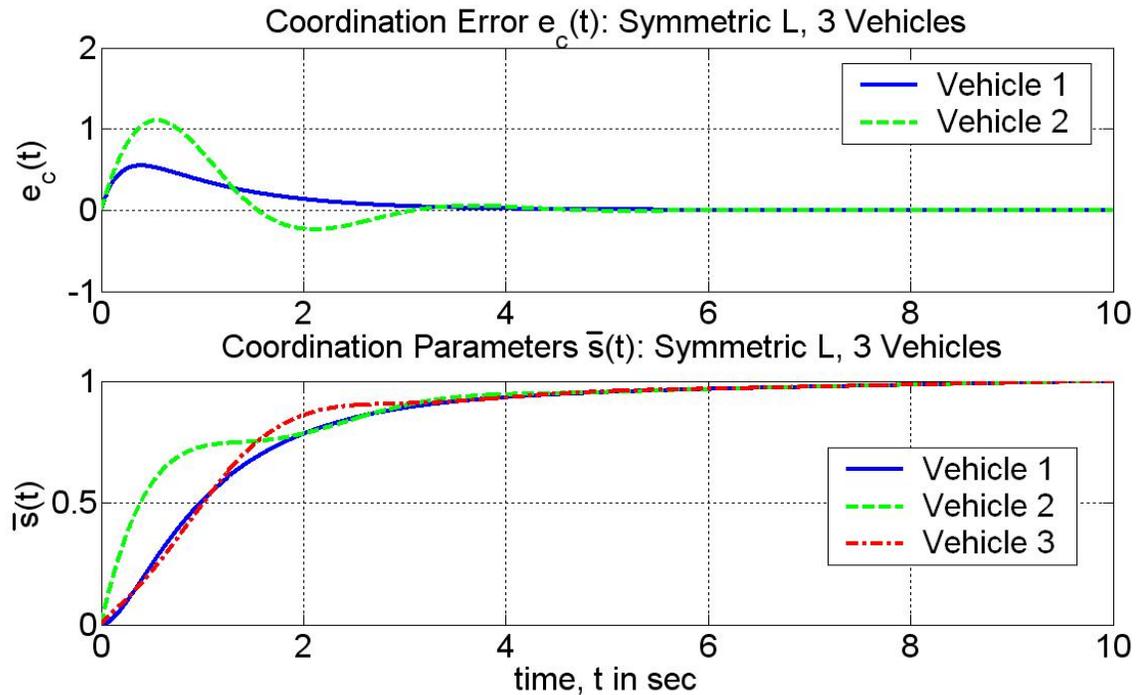


Figure 5.3: Symmetric Laplacian: an example

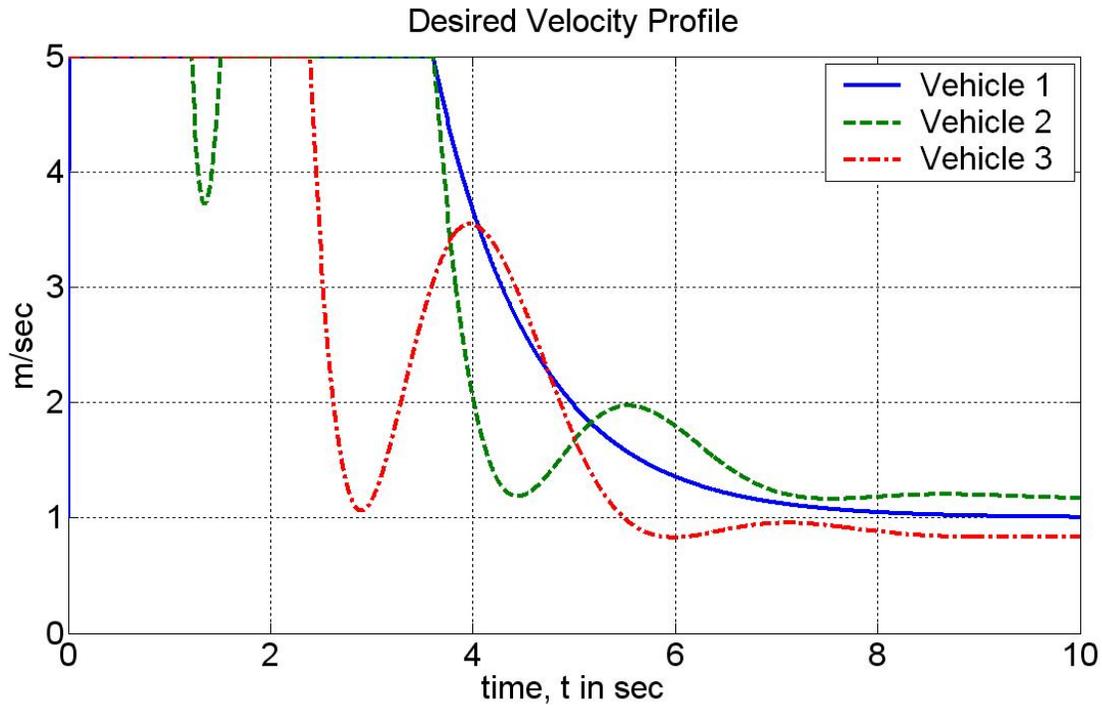


Figure 5.4: Desired speed profile of the leader with $v_{d_{\max}} = 5 \text{ m/s}$.

5.3 Time-Critical Coordination With Uni-Directional Communication

Section 5.2 considers the case when communication between vehicles is undirected. In the case when the underlying graph that captures the communication topology of the vehicles, Γ , becomes directed, the coordination controller (5.11), (5.12) is no longer valid. Indeed, L is no longer symmetric and cannot be decomposed into $L = UL_dU^\top$ as in (5.2). Recall from Section 5.1 that if Γ has at least one globally reachable vertex, the Laplacian can be decomposed as

$$L = F_1 L_{11} F_2, \quad (5.14)$$

where $F_1 \in \mathbb{R}^{M \times (M-1)}$, $F_2 \in \mathbb{R}^{(M-1) \times M}$, and $L_{11} \in \mathbb{R}^{(M-1) \times (M-1)}$ is a positive definite matrix. Let $F = [1_M \ F_1]$. The properties (5.4) hold for F_1, F_2, F, L_{11} , and L . The coordination error vector becomes

$$e_{c_D}(t) = F_2 \bar{s}(t) \quad (5.15)$$

and its dynamics is given by

$$\dot{e}_{c_D}(t) = F_2 u_{c_D}(t), \quad (5.16)$$

where $u_{c_D}(t)$ is the new control law corresponding to a directed graph Γ . If the control law (5.11) is chosen as $u_{c_D}(t)$, then the feedback system consisting of (5.16) and (5.11) is

$$\begin{aligned} \dot{\chi}_{I_D}(t) &= bC_2 L \bar{s}(t) \\ \dot{e}_{c_D}(t) &= aL_{11} e_{c_D}(t) + F_2 \left(C_1^\top \frac{v_{d_1}}{s_{f_1}} + C_2^\top \chi_{I_D}(t) \right). \end{aligned} \quad (5.17)$$

Define

$$x_{c_D}(t) = \begin{bmatrix} \chi_{I_D}(t) - \frac{v_{d_1}}{s_{f_1}} \mathbf{1}_{M-1} \\ e_{c_D}(t) \end{bmatrix}.$$

One can compute from (5.16) using the definitions of C_1, C_2 in (5.12) that the coordination system (5.17) reduces to

$$\dot{x}_{c_D}(t) = \begin{bmatrix} 0_{M-1} & bC_2 F_1 L_{11} \\ F_2 C_2^\top & aL_{11} \end{bmatrix} x_{c_D}(t). \quad (5.18)$$

The stability of the closed-loop system (5.18) is determined from the roots of

$$\det(\lambda^2 \mathbb{I}_{M-1} - \lambda aL_{11} - bF_2 C_2^\top C_2 F_1 L_{11}). \quad (5.19)$$

Defining $S = F_2 C_2^\top C_2 F_1$ implies that S is exactly represented by its singular value decomposition

$$S = S_1^\perp D S_2^\perp,$$

where $S_1^\perp = F_2, S_2^\perp = F_1$ are orthogonal matrices (5.4) and

$$D = C_2^\top C_2 = \begin{bmatrix} 0 & 0_{M-1}^\top \\ 0_{M-1} & \mathbb{I}_{M-1} \end{bmatrix}$$

is a diagonal matrix with non-negative diagonal elements. It is well-known that the rank of a matrix can be determined by the number of zero elements on the diagonal of D [50]. Hence, the $(M - 1) \times (M - 1)$ matrix $F_2 C_2^\top C_2 F_1$ has rank $M - 2$ and is not full rank. This further implies that $\lambda = 0$ is a solution for (5.19) since

$$\det(-b F_2 C_2^\top C_2 F_1 L_{11}) = -b \det(L_{11}) \det(F_2 C_2^\top C_2 F_1 L_{11}) = 0.$$

As a consequence, the system in (5.18) always has one zero eigenvalue and is marginally stable for any $a, b < 0$. The coordination objective is no longer achievable with the control law (5.11).

To illustrate the above circle of ideas, Figure 5.5 shows a simulation example with the communication topology depicted by the directed graph Γ of Figure 5.2 (b) for three vehicles. The corresponding matrix parameters are given in (5.5) - (5.8) and the additional parameters are given by

$$\chi_I(0) = [2 \ 1]^\top, s_f = [20 \ 15 \ 25]^\top, a = b = -1.$$

The desired leader velocity profile is again constant at $v_1 = 1$ m/s. The eigenvalues of the closed-loop system (5.18) are

$$\lambda_1 = -2.3247, \lambda_{2,3} = -0.3376 - 0.5623i, \lambda_4 = 0.$$

Figure 5.3 (top) shows the coordination parameter $\bar{s}(t)$ which converges for vehicle 3 but not vehicle 2. Intuitively, vehicle 2 cannot communicate with any vehicles in the fleet so it cannot coordinate its velocity with the other vehicles. Hence, the coordination objective is not satisfied with the control law in (5.11). Figure 5.3 (bottom) shows the corresponding coordination error $e_c(t)$, which does not converge to zero.

To correct for the new communication topology, consider a reference vehicle, denoted by vehicle \mathcal{L} . The reference vehicle's speed profile will be used in order to achieve time-critical coordination. Note that the reference vehicle is taken as a virtual vehicle only so that the mission objective of the fleet will not be compromised with the loss of any one vehicle in comparison to the case of bi-directional communication in Section 5.2. A new control law

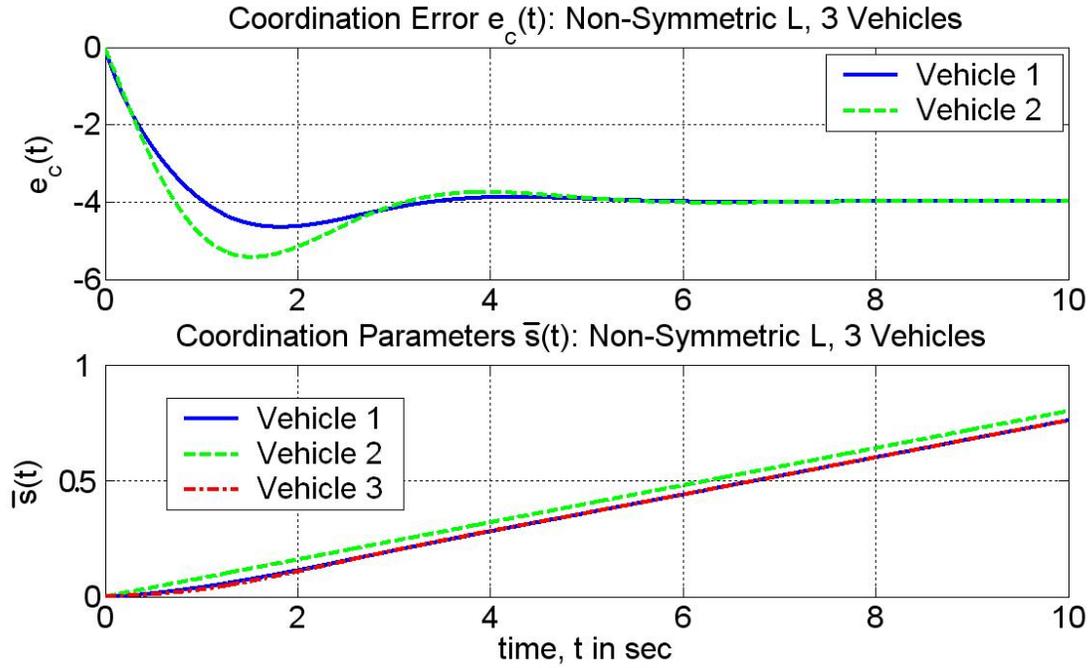


Figure 5.5: Non-symmetric Laplacian: an example

that achieves the coordination objective in the presence of uni-directional communication constraints is introduced.

Proposition 12 *The control law*

$$\begin{aligned}
 u_{c_D}(t) &= aL\bar{s}(t) + C_1^\top \frac{v_{\mathcal{L}}(t)}{s_{f_1}} + C_2^\top \chi_{I_D}(t), \\
 \dot{\chi}_{I_D}(t) &= bC_2L\bar{s}(t) + c \left(\chi_{I_D}(t) - \frac{v_{\mathcal{L}}}{s_{f_1}} \mathbf{1}_{M-1} \right),
 \end{aligned} \tag{5.20}$$

where $c < 0$, $v_{\mathcal{L}}$ is the reference speed profile of the virtual reference leader, and C_1, C_2 are defined in (5.12), solves the coordination objective for the system (5.16) and guarantees that the error, $e_{c_D}(t)$, for a directed graph Γ goes to zero exponentially if the graph Γ has at least one globally reachable vertex.

Proof. The closed-loop system of (5.16) with the controller in (5.20) becomes

$$\begin{aligned} \dot{x}_{c_D}(t) &= \begin{bmatrix} c\mathbb{I}_{M-1} & bC_2F_1L_{11} \\ F_2C_2^\top & aL_{11} \end{bmatrix} x_{c_D}(t) + \begin{bmatrix} 0_{M-1} \\ F_2C_1^\top \frac{v_{\mathcal{L}}}{s_{f1}} + F_2C_2^\top \frac{v_{\mathcal{L}}}{s_{f1}} \mathbf{1}_{M-1} \end{bmatrix}, \\ &= \begin{bmatrix} c\mathbb{I}_{M-1} & bC_2F_1L_{11} \\ F_2C_2^\top & aL_{11} \end{bmatrix} x_{c_D}(t), \end{aligned} \quad (5.21)$$

where

$$x_{c_D}(t) = \begin{bmatrix} \chi_{I_D}(t) - \frac{v_{\mathcal{L}}}{s_{f1}} \mathbf{1}_{M-1} \\ e_{c_D}(t) \end{bmatrix}$$

and the properties of C_1, C_2, F_2 are exploited as before. To show that the origin is an exponentially stable equilibrium of the system (5.21) for any $a, b, c < 0$, compute the eigenvalues of (5.21), which correspond to the roots of

$$\det(\lambda^2 \mathbb{I}_{M-1} - \lambda(aL_{11} + c\mathbb{I}_{M-1}) + (ac\mathbb{I}_{M-1} - bF_2C_2^\top C_2F_1)L_{11})$$

or equivalently

$$\det(\lambda^2 L_{11}^{-1} - \lambda(a\mathbb{I}_{M-1} + cL_{11}^{-1}) + (ac\mathbb{I}_{M-1} - bF_2C_2^\top C_2F_1)).$$

The result follows from an application of Lemma 7 since

$$ac\mathbb{I}_{M-1} - bF_2C_2^\top C_2F_1 > 0$$

for any $a, c < 0$ and L_{11} is positive definite.

To illustrate the theoretical findings, Figure 5.6 shows a simulation example with the communication topology depicted by the directed graph Γ of Figure 5.2 (b) for three vehicles using the controller (5.20). The reference velocity profile is constant at $v_{\mathcal{L}} = 1$ m/s. The parameters for the simulation are the same as in Figure 5.5. The eigenvalues of the closed-loop system (5.21) are

$$\lambda_1 = -4.241897, \quad \lambda_{2,3} = -1.379052 \pm 1.182636i, \quad \lambda_4 = -1.$$

Figure 5.6 (top) shows the coordination parameter $\bar{s}(t)$, which converge to $\bar{s}_1(t) = \bar{s}_2(t) = \bar{s}_3(t)$ to satisfy the coordination objective. Figure 5.6 (bottom) shows the corresponding coordination error $e_c(t)$ which converges to zero for vehicles 2 and 3. The coordination objective is satisfied in the presence of uni-directional communication constraints.

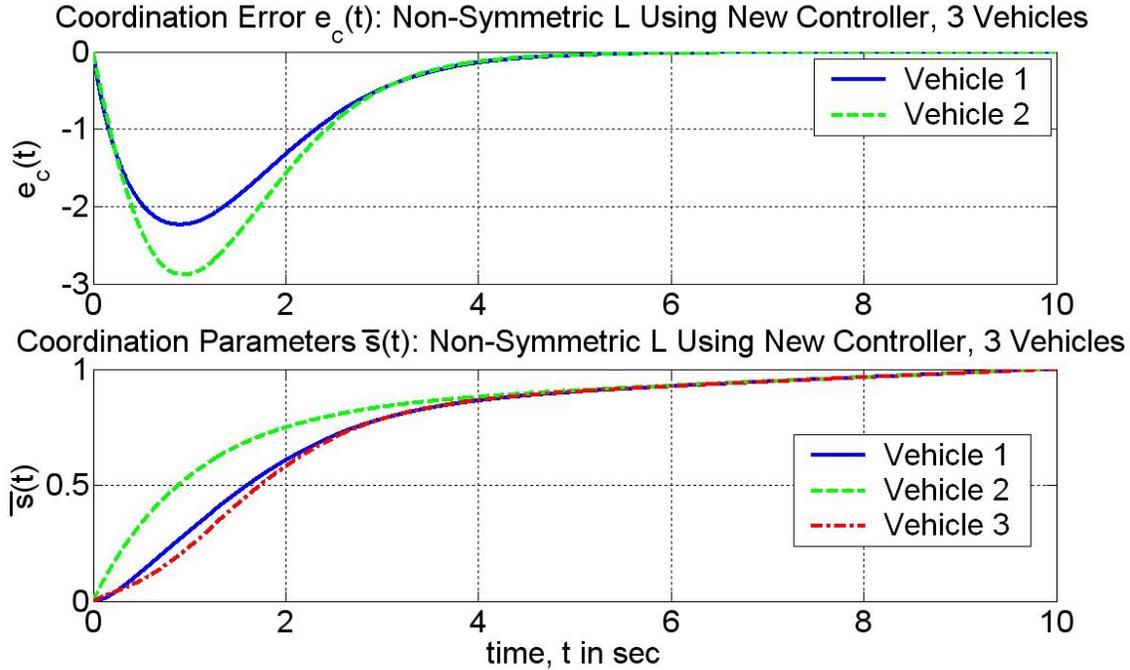


Figure 5.6: Non-symmetric Laplacian: recovery of coordination objective with new controller (5.20)

Remark 22 *The controller (5.20) requires that each vehicle knows the reference speed profile $v_{\mathcal{L}}$ a priori, a condition which was not present in the case when Γ is undirected. This is a consequence of the fact that communication channels have been further limited and is a design trade-off to be considered in implementation.*

The controller (5.20) retains a PI structure while introducing an additional error term $\chi_{I_D}(t) - \frac{v_{\mathcal{L}}}{s_{f_1}}$ into the dynamics of $\chi_{I_D}(t)$. This feedback term imposes the condition that the desired velocity profile $v_{\mathcal{L}}$ be known *a priori* by all vehicles. Alternatively, the author of

[30] implements a control law for $u_c(t)$ using a Proportional-Derivative (PD) design in the presence of directional communication constraints. For comparison, the controller of [30] is summarized:

Let

$$v_{e_i}(t) = v_i(t) - \frac{s_{f_i}}{v_{\mathcal{L}}}$$

denote the *coordination velocity error*. The coordination error dynamics can be computed to be

$$\dot{e}_{CPD}(t) = cF_2v_e(t) + F_2d(t), \quad (5.22)$$

where $v_e(t) = [v_{e_1}(t) \ v_{e_2}(t) \ \dots \ v_{e_M}(t)]^\top$ and the components of $d(t)$ are given by

$$d_i(t) = \frac{1}{s_{f_i}} \left((\cos \psi_{e_i}(t) - 1)v_i(t) + k_1x_{e_i}(t) \right),$$

for some gain $k_1 > 0$.

Proposition 13 [30] *The control law*

$$u_{cPD}(t) = av_e(t) + \frac{b}{c}L\bar{s}(t), \quad (5.23)$$

where $a, b < 0$ are scalars solves the coordination objective for the dynamics (5.22) if the graph Γ has at least one globally reachable vertex and

$$\frac{a^2}{b} > \max_{\mu \in \sigma(L) - \{0\}} \frac{\text{Im}(\mu)^2}{\text{Re}(\mu)}, \quad (5.24)$$

where $\sigma(\cdot)$ denotes the spectrum of a matrix and $\text{Im}(\cdot)$, $\text{Re}(\cdot)$ are the imaginary and real components of a complex number, respectively. Furthermore, the control law (5.23) can be written in scalar form as

$$u_{cPD_i}(t) = av_{e_i}(t) + bs_{f_i} \sum_{j \in J_i} (\bar{s}_i(t) - \bar{s}_j(t)).$$

The proof is based on Lyapunov-type analysis, details can be found in [30]. The controller in [30] satisfies the coordination objective as can be seen in Figure 5.7 with the same simulation parameters as Figures 5.5 - 5.6. Note that the controller (5.23) in [30] also requires that the reference speed profile $v_{\mathcal{L}}$ be known by all vehicles *a priori*. While the controller (5.23) satisfies the coordination control objective, it can suffer from steady-state error due to its PD structure as can be seen in Figure 5.7. The controller (5.20) has the benefit of an integral controller which drives the error $e_c(t)$ to zero at an exponential rate.

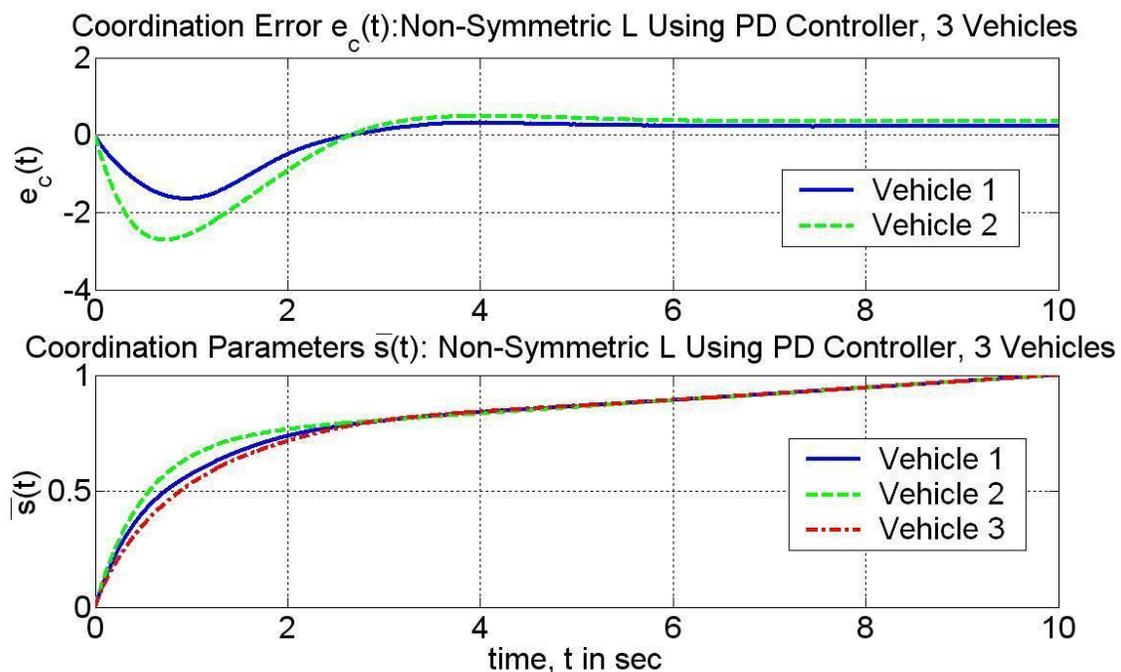


Figure 5.7: Non-symmetric Laplacian: PD controller (5.23)

One can compare the performance between the first PI controller of (5.11) in Section 5.2 and the new PI controller (5.20). The advantage of the second controller (5.20) is that it *ensures the coordination objective is met in the presence of bi-directional and uni-directional communication topologies* in comparison to (5.11). Furthermore, the method does not depend on any single vehicle as compared to the method in 5.11. The benefit of the first controller (5.11) is that *only the leader vehicle is required to know the desired speed profile beforehand,*

allowing for more flexibility due to unforeseen events and uncertainties. To compare the performance of both controllers, consider the symmetric Laplacian case from Section 5.2:

$$L = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

The closed-loop systems using the controllers (5.11) and (5.20) are compared and their eigenvalues are shown in Figure 5.8. The closed-loop eigenvalues using the first PI controller (5.11) are fixed at

$$\lambda_1 = -5, \lambda_{3,4} = -1 \pm 2i, \lambda_4 = -1$$

while the eigenvalues using the second PI controller (5.20) are the roots of

$$\left(\lambda + (6 - c)\lambda + (5 - 6c)\right) \left(\lambda + (2 - c)\lambda + (5 - 2c)\right), \quad c < 0.$$

Applying Lemma 7 implies that the corresponding solutions are stable eigenvalues for any $c < 0$. The eigenvalues for both controllers are shown in Figure 5.8 (top). The root locus plot for the system is shown in Figure 5.8 (bottom). Figures 5.8 shows that the controller (5.20) affords additional design flexibility and allows for the designer to choose the eigenvalue positions based on the choice of c . The performance of the controller (5.20) can be improved by judicious choice of c .

Remark 23 *The controllers (5.11), (5.20), and (5.23) are outer-loop controllers that assume perfect knowledge of the dynamics of $v(t)$. In practice, uncertainties and disturbances are present which can decrease the performance of the coordination controllers. In order to account for these uncertainties, the authors in [43, 44] have wrapped an adaptive output feedback controller known as the \mathcal{L}_1 adaptive controller around an existing autopilot to achieve inner-outer loop structure with guaranteed performance. The \mathcal{L}_1 controller adapts quickly to uncertainties via a large adaptation gain, and filters high-frequency signals using a low-pass filter. The bandwidth of the low-pass filter is chosen so that the \mathcal{L}_1 gain of the system's transfer function satisfies a stability requirement using Small-Gain Theorem. The extension*

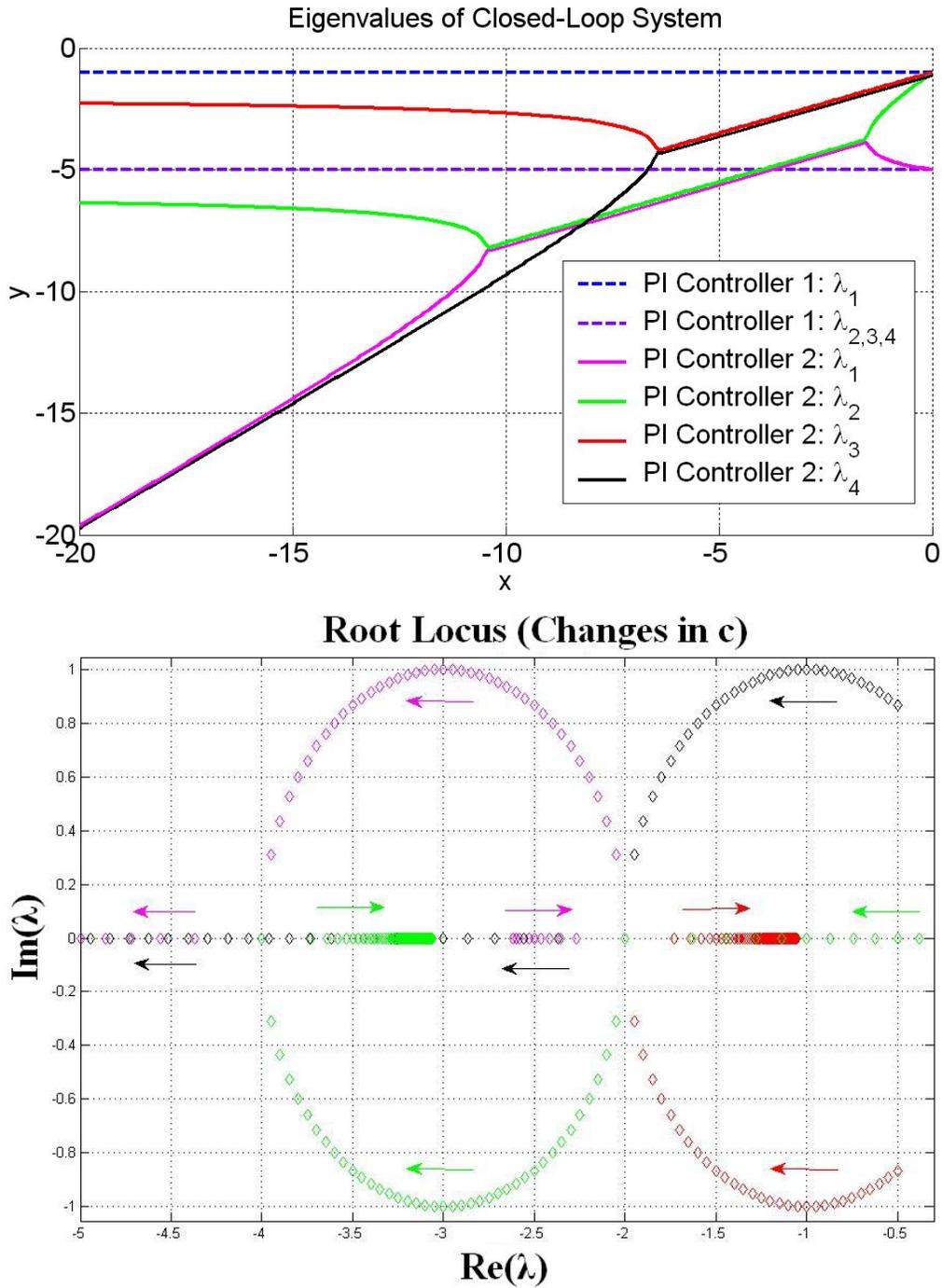


Figure 5.8: Eigenvalues corresponding to PI controllers (5.11) and (5.20) and root locus plot

of the \mathcal{L}_1 adaptive controller augmented with path following and time-critical coordination for a fleet of UAVs is presented in [43, 44].

5.4 An Illustrative Example

To encompass the components of path generation, path following, obstacle avoidance, vision-based estimation, and time-critical coordination, a simulation example is presented for three vehicles. The ideas present in this dissertation present path generation, path following, and obstacle avoidance with vision decoupled from vehicle coordination. Due to this decoupling, the original optimality claims in path generation and obstacle avoidance are not guaranteed in the presence of coordination. Coordination modifies each vehicle's speed profiles, which changes the optimality properties of the path deformation. The relationship between time-critical coordination on the other components (i.e. path generation, path following, obstacle avoidance) remains an open problem for future research. The following simulation example serves as an illustration of the different ideas in this dissertation without claims of optimality.

First, the path generation parameters for a desired configuration are as follows.

Parameter	Robot 1	Robot 2	Robot 3	Parameter	Robot 1	Robot 2	Robot 3
x_0 (m)	60	10	90	v_f (m/s)	1	2	1
y_0 (m)	0	10	-20	a_0 (m/s ²)	0	0.1	0
x_f (m)	100	40	140	a_f (m/s ²)	0	0.2	0
y_f (m)	0	25	0	ψ_{B_0} (deg)	15	10	-10
v_0 (m/s)	1.5	1	2.5	ψ_{B_f} (deg)	0	-10	5

The velocity and acceleration limits for each vehicle are $v_{\max} = 3$ m/s, $v_{\min} = 0.1$ m/s, $a_{\max} = 0.5$ m/s². The final time is chosen to lie within the bounds $t_f \in [20, 200]$ seconds and the optimization weights in (2.29) for each vehicle are equivalent at $w_1, w_2, w_3 = 100$.

The obstacle configuration are as follows. There are three obstacles, located at

$$P_x = [75 \ 25 \ 125]^\top, \quad P_y = [-1 \ 15 \ -13]^\top$$

with radii $L = [5 \ 3 \ 2]^\top$ m, respectively. The bound on L is $L_{\max} = 10$ m and the *a priori* distance $d = 1$.

Figure 5.9 shows the admissible paths generated by the polynomial functions for $\eta(s) = 1$. Obstacles are shown by the dotted red lines. Figure 5.10 shows the associated velocity and acceleration profiles, which are bound by the limits $v_{\max}, v_{\min}, a_{\max}$.

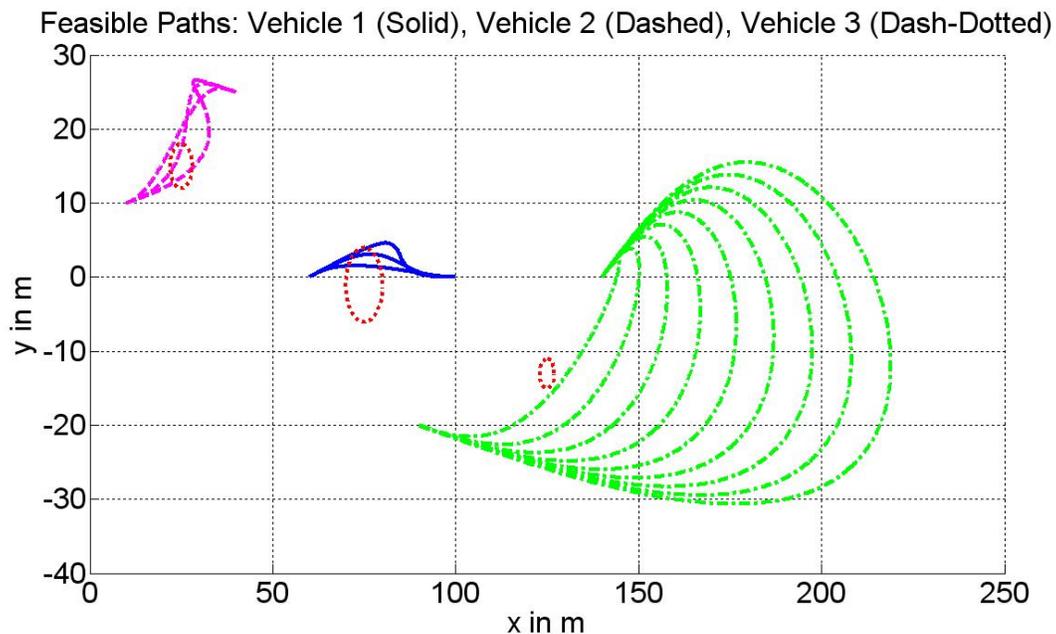


Figure 5.9: Feasible paths

Out of these feasible paths, the optimal path must be chosen with respect to the cost function

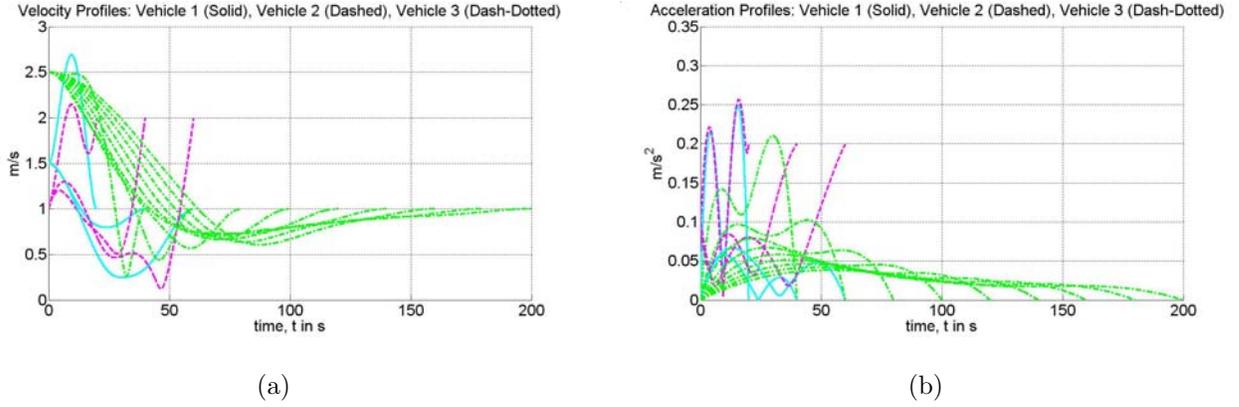


Figure 5.10: Feasible velocity and acceleration profiles

defined in (2.29), where $J_{e,i}$ is given as

$$J_{e,i} = \int_0^{t_f} v_i(t)^3 dt,$$

where $v_i(t)$ is the speed of vehicle i , and

$$J_{o,i} = \int_0^{t_f} \frac{1}{\|p_i(t) - (P_{x,i}, P_{y,i})\|} dt,$$

where $P_{x,i}, P_{y,i}$ are the i th component of the obstacle locations P_x, P_y , respectively. The total cost function is

$$J = \sum_{i=1}^3 J_{e,i} + J_{o,i}.$$

Optimizing over the set of feasible paths leads to a final time $t_f = 40$ and the desired path is shown in Figure 5.11 with associated speed and acceleration profiles in Figure 5.12. The circled stars in Figure 5.11 denote the starting position of each vehicle for clarity. Note that the desired paths pass through obstacles. Preliminary obstacle locations may be incorrect and may change as more accurate information from onboard cameras is processed. Hence, the path generation is less aggressive in diverting the vehicle paths away from obstacles.

The vehicle separation distance Ξ is determined by the parameters L_{\max}, κ_{\max} , and d . If $\kappa_{\max} < 1$, where κ_{\max} is the maximum deformation per τ -step, then one can compute from the values of κ_{\max}, L_{\max} , and d that the desired separation used in the path generation Ξ is

$$\Xi = 2(L_{\max} + d) + 2\max(\kappa_{\max}, 1) + \epsilon_d = 30,$$

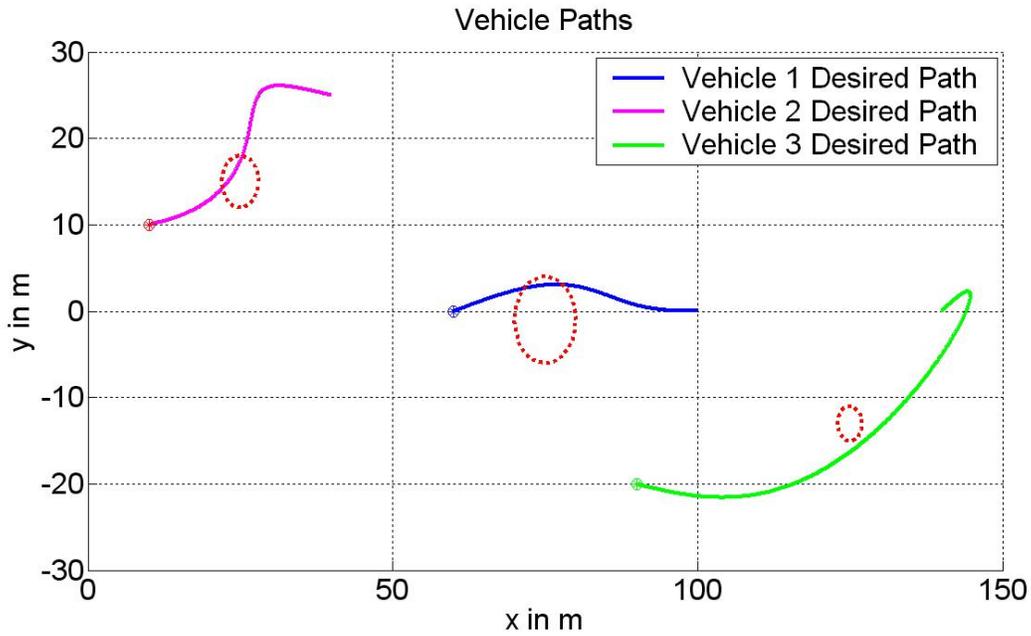


Figure 5.11: Desired path

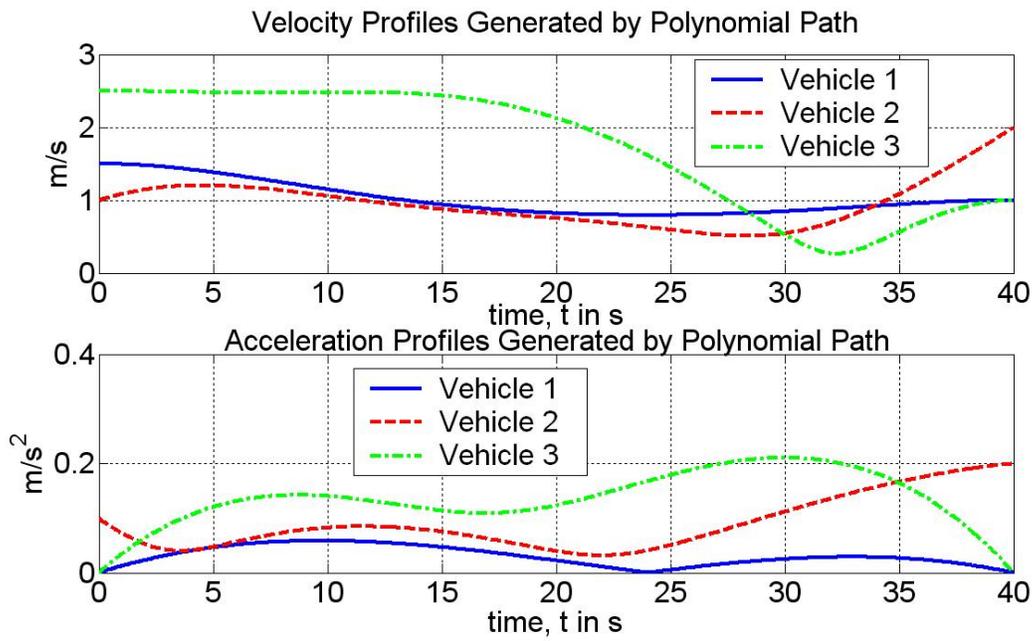


Figure 5.12: Desired velocity and acceleration profiles

where ϵ_d is chosen as $\epsilon_d = 6$. The distance between vehicles is shown in Figure 5.13. The minimum separation distance $\Xi = 30$ m is marked by the dotted black line.

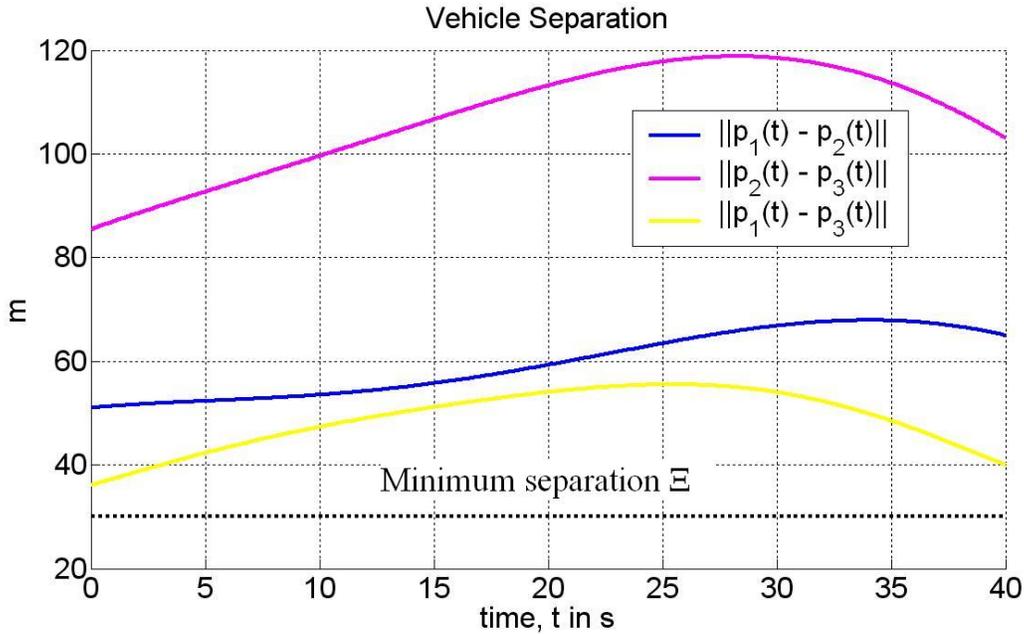


Figure 5.13: Vehicle separation distances from desired paths

To follow the desired paths, the path following controller (2.89) is used with the following gains:

$$k_1 = 3; k_2 = 1; k_3 = 3; k_4 = 3; k_5 = 3; \epsilon_0 = 0.75.$$

The desired speed profile is determined from the coordination control law described later. First, the path following is shown in the absence of coordination. The desired speed profile is chosen to be constant at $v_d(t) = v_d = 1$ m/s.

Path following errors are initiated as:

$$\begin{aligned} x_{e,1}(0) &= 2 \text{ m} & y_{e,1}(0) &= -1.5 \text{ m} & \psi_{e,1}(0) &= 0^\circ \\ x_{e,2}(0) &= 1 \text{ m} & y_{e,2}(0) &= -2 \text{ m} & \psi_{e,2}(0) &= 5^\circ \\ x_{e,3}(0) &= 1.5 \text{ m} & y_{e,3}(0) &= 1 \text{ m} & \psi_{e,3}(0) &= -5^\circ. \end{aligned}$$

The initial rates of rotation are $r_1(0) = 0.5$ rad/s, $r_2(0) = 1$ rad/s, $r_3(0) = 0.75$ rad/s. Using

the desired paths generated by the polynomial path generation method, the vehicles track the paths in the absence of obstacles (Figure 5.14). The black dotted circles represent the obstacles and boundaries. The outer black dotted circle represents the sensor radius $R_s = 10$ m, which is larger than the danger radius $R_{d,i} = L_{d,i} + d$ for $i = \{1, 2, 3\}$. The innermost circle represents the obstacle radii L_i . For the obstacle sizes chosen,

$$R_d = [R_{d,1} \ R_{d,2} \ R_{d,3}]^\top = [6 \ 4 \ 3]^\top.$$

The corresponding path following errors are shown in Figure 5.15. No obstacle avoidance is taken at this time. Note that the vehicles follow their respective paths, which leads to collision with obstacles.

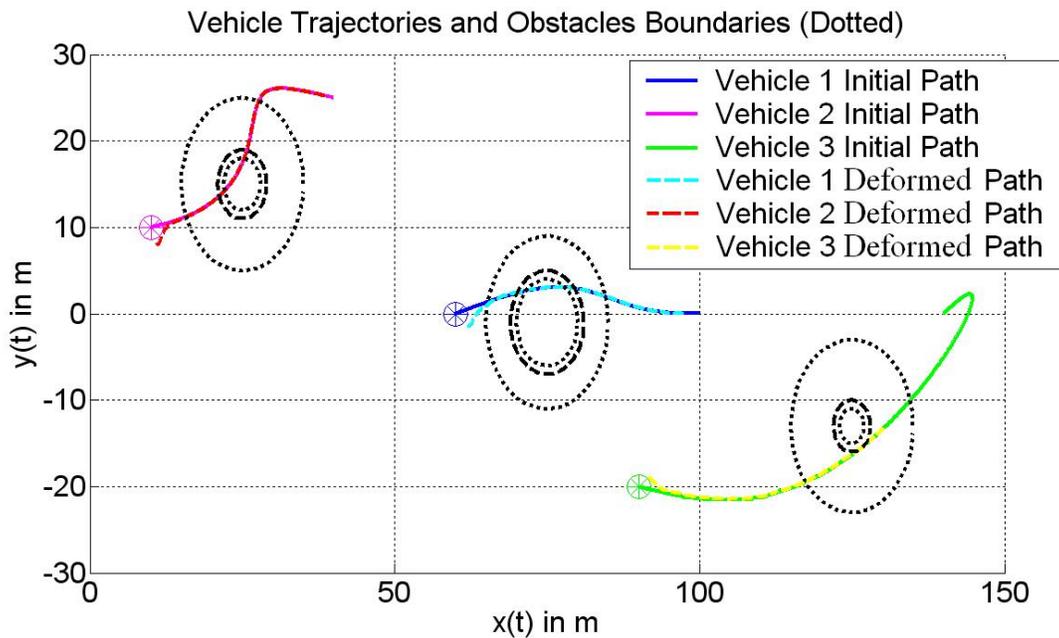


Figure 5.14: Vehicle path following, no obstacle avoidance

To include obstacle avoidance, the path deformation parameters are chosen as $d_0 = 0.001$ and $\kappa_{\max} = 0.75 < 1$. This choice of κ_{\max} justifies the choice of $\Xi = 30$ m since $\kappa_{\max} < 1$. The deformation of the original paths around obstacles is shown in Figure 5.16 (a), where obstacles are shown by the red dotted lines. The obstacle avoidance scheme is successful

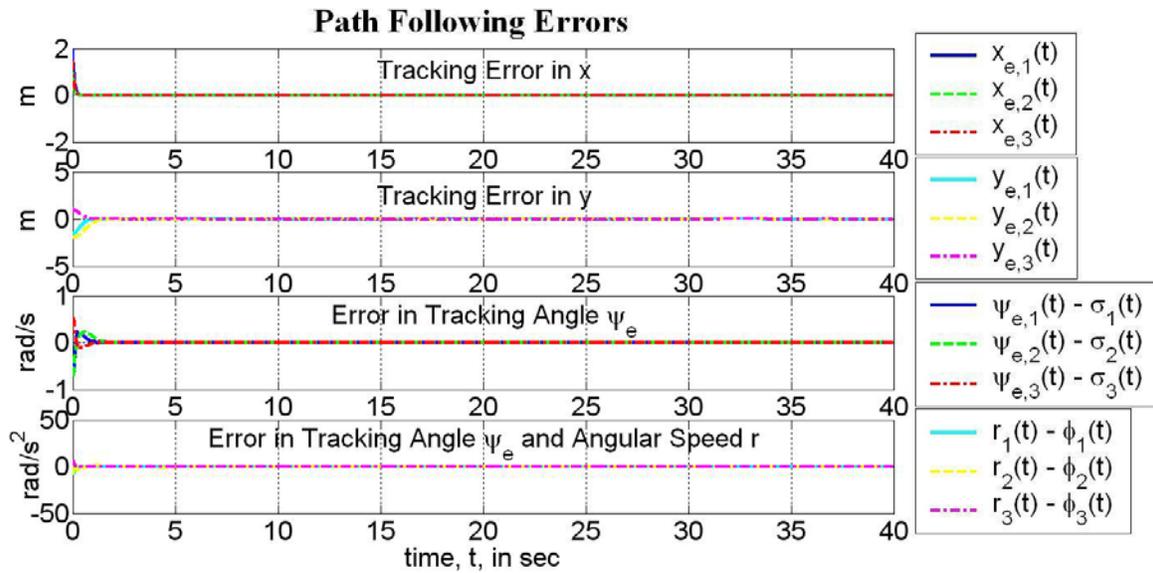
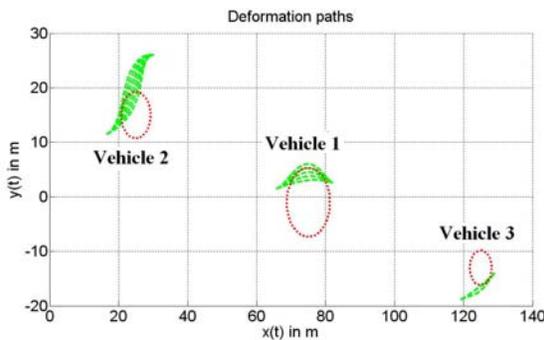
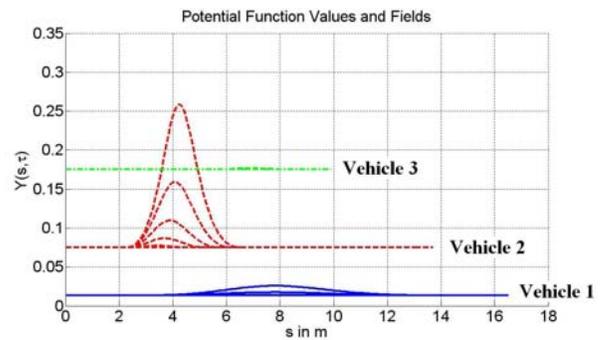


Figure 5.15: Path following errors

in diverting the path from obstacles and reducing the potential function defined in (3.23) chosen for obstacle avoidance (Figure 5.16 (b)). Figure 5.17 shows the vehicle separation distances before and after deformation. The separation before deformation as defined by the desired polynomial paths are shown in Figure 5.17 by the solid lines and the vehicle separation distances of the vehicle paths after deformation are shown by the dashed lines. All vehicles remain a minimum distance of $\Xi = 30$ m apart after obstacle avoidance.



(a) Deformed paths



(b) Potential function values

Figure 5.16: Path deformation algorithm

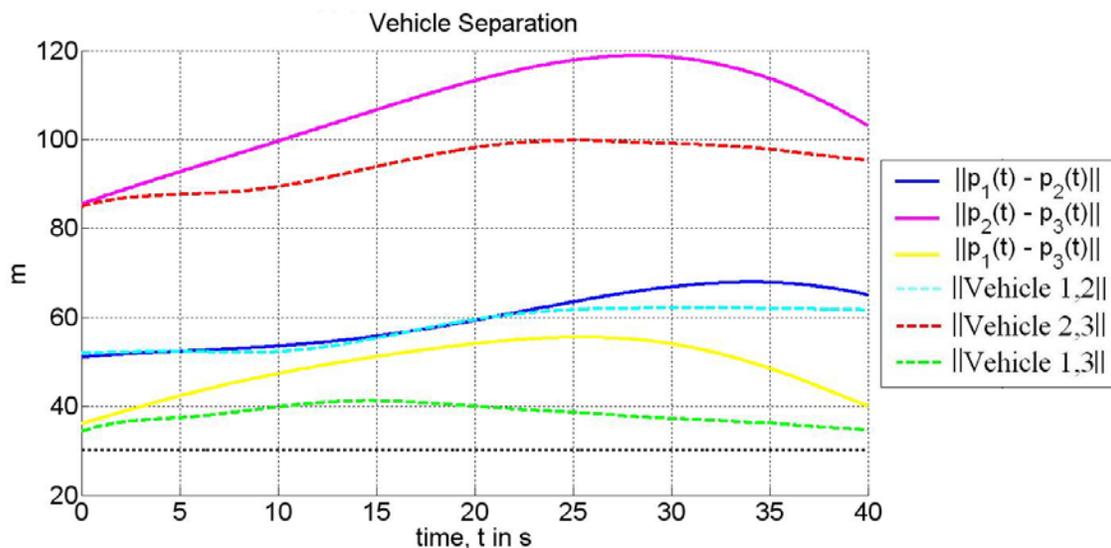


Figure 5.17: Vehicle separation distances

The above simulations assumed that the obstacle size L was known to the vehicles. Next, the case when L is unknown is considered. To add vision-based estimation to determine the unknown parameter L , the vehicles are assumed to each have a sensing radius of $R_s = 10$ m. The estimation parameters are chosen as $\Gamma = 1000$, $c = 5$, $A_m = \text{diag}(-5, -5)$, and $Q = \mathbb{I}_2$. Figure 5.18 shows the estimated values $\hat{L}_i(t)$ of the obstacle radii L_i for $i = \{1, 2, 3\}$. All estimates are seen to converge to their true values $L = [5 \ 3 \ 2]^\top$ m. Figure 5.19 shows the bearing and subtended angles $\beta(t), \alpha(t)$ and their estimates $\hat{\beta}(t), \hat{\alpha}(t)$, respectively, for each vehicle. All estimates converge to their true values upon application of the fast estimator. Figure 5.20 shows the resulting trajectories with obstacle avoidance and estimation. The original undeformed paths are shown by the dashed lines, obstacles and their respective boundaries are shown by the black dotted line, and the deformed paths using the estimated parameters \hat{L} from the fast estimator are shown by the solid lines. The deformed paths divert the vehicles away from their respective obstacles as compared to the original paths and then rejoin them when the obstacle has been avoided. Note that Vehicle 3 in particular does not complete its path at the end of the simulation. The time-critical coordination will

adjust the vehicle’s speed profile to ensure that all vehicles arrive at their final destinations (i.e. the ends of their respective paths) simultaneously.

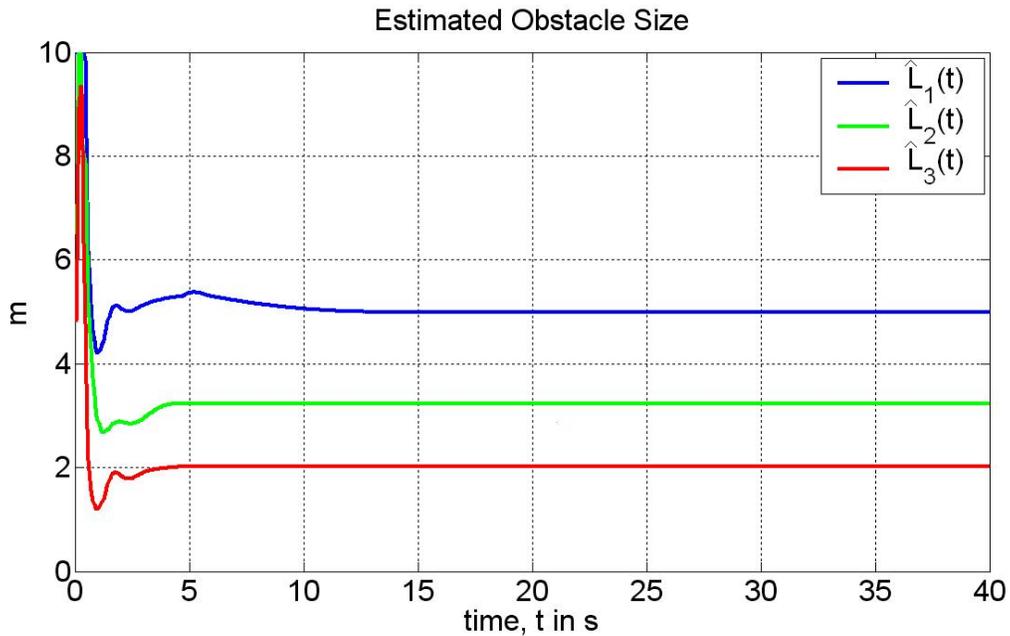
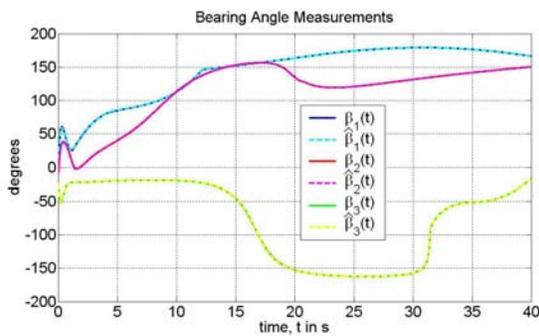
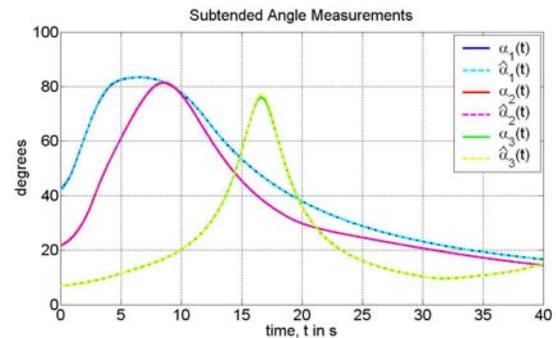


Figure 5.18: Estimates of obstacle radii L_1, L_2, L_3



(a) Bearing angle measurements



(b) Subtended angle measurements

Figure 5.19: $\beta(t), \alpha(t)$ and their estimates $\hat{\beta}(t), \hat{\alpha}(t)$ for each vehicle

Finally, to coordinate the three vehicles so that they arrive at their final destinations simultaneously, consider the case when the underlying directed communication topology is

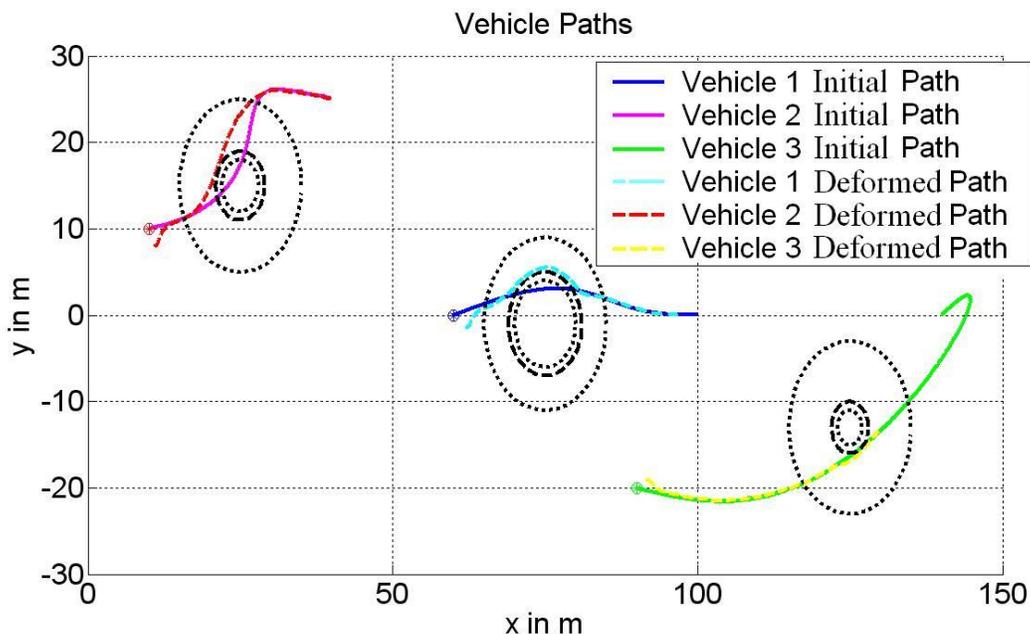


Figure 5.20: Vehicle trajectories with obstacle avoidance and estimation

represented by the Laplacian

$$L = \begin{bmatrix} 2 & -1 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

It can be verified that

$$F_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}, \quad F_2 = \begin{bmatrix} 1 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, \quad L_{11} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$$

satisfy the properties in (5.4) for the Laplacian given. The coordination errors are shown in Figure 5.21 (a). The controller (5.20) guarantees that the errors converge to zero exponentially. The coordination parameters $\bar{s}(t)$ are shown in Figure 5.21 (b). The parameters converge after 5 seconds, long before the final arrival time of $t_f = 40$ seconds. The coordination algorithm changes the speed profiles of the vehicles. Hence, the progression of the vehicles along their respective path is also changed. To view this change, Figure 5.22 shows

the vehicle trajectories with time-critical coordination. Note that all vehicles arrive at their final destinations as compared to Figure 5.20, where Vehicle 3 has not completed its path at the end of the simulation.

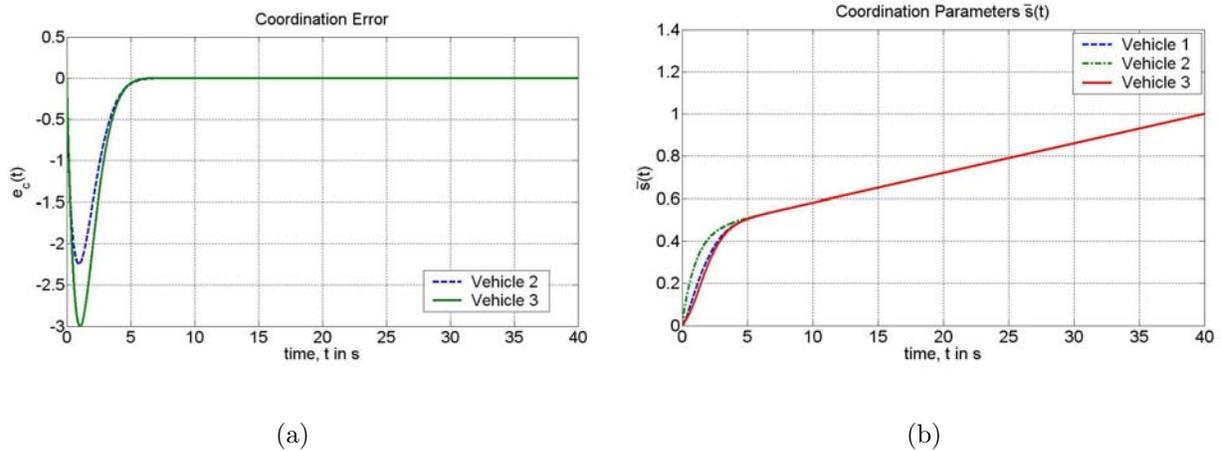


Figure 5.21: Coordination parameters

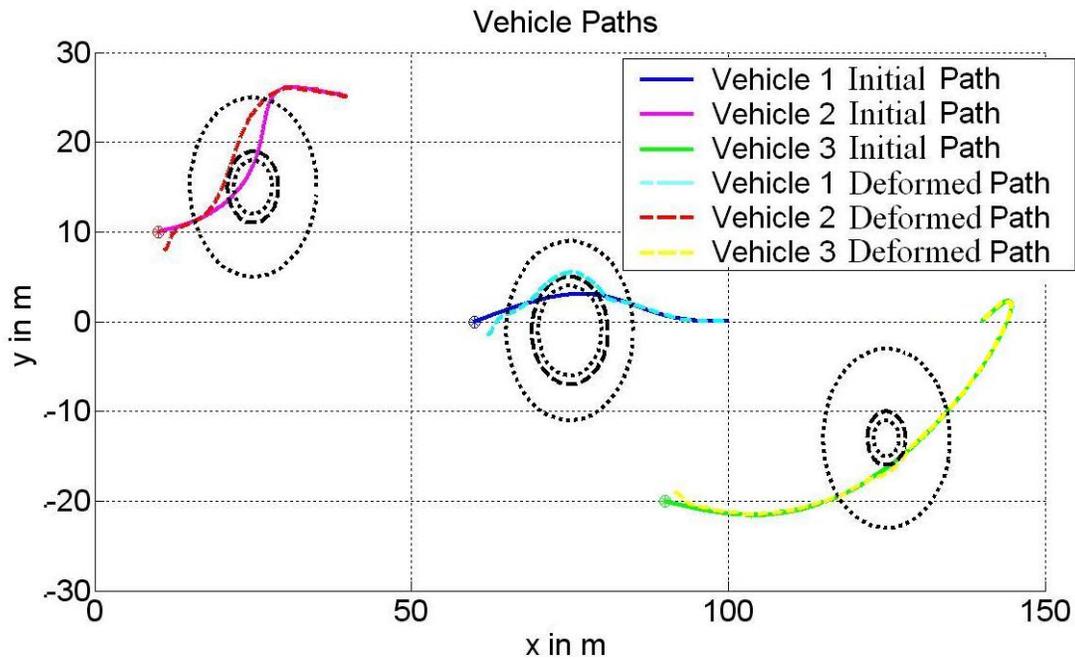


Figure 5.22: Vehicle trajectories with obstacle avoidance, estimation, and coordination

5.5 Summary

Time-critical coordination is formulated for two scenarios: bi-directional communication and uni-directional communication between vehicles. An existing algorithm for the former case is given [43, 44], which requires that only a leader vehicle know the desired speed profile *a priori*. A controller is introduced which guarantees coordination for both types of communication and uses a PI-type structure to drive the coordination error to zero. The trade-off of this controller is that it requires that each vehicle know a reference speed profile beforehand. An advantage of this controller is that it does not require a leader whose safety is essential to the success of the mission. Furthermore, the method is advantageous over a PD controller in [30], which can be prone to steady-state errors. In the direction of future research, an algorithm can be considered where the remaining vehicles “agree” on a new leader in the event that the leader vehicle is lost to complete the mission. This will relax the assumption on the need of *a priori* knowledge of the reference velocity for all vehicles and give more flexibility to the entire mission.

Chapter 6

Conclusion

6.1 Summary

This dissertation provided a real-time path generation method for multiple vehicles using a single polynomial function. The method ensures collision-free maneuvers and incorporates the coordination objective in the absence of delays. The position and velocity of each vehicle are decoupled by defining a dummy variable, s , whose relationship with time is defined differently for each vehicle. This relationship is explored for various types of functions. Obstacle avoidance is implemented on a preliminary level since accurate information about obstacles may not be available. A path following controller ensures that the vehicles track their generated paths. As each vehicle travels along its path, it can encounter at most one obstacle, which is not an obstacle for any other vehicles. In the presence of this obstacle, a real-time obstacle avoidance algorithm is triggered which avoids obstacles by iteratively deforming the original path. During the deformation, the kinematic inputs are constrained to remain within vehicle limits by implementing the projection operator. The deformation process is shown to be a bounded process and does not lead to diverging paths. The inputs to the obstacle avoidance algorithm are obstacle size and position, which are estimated via two estimators: identifier-based observer, and fast estimator. The performance of the fast estimator can be

arbitrarily reduced with judicious choice of the estimation parameters as compared to the IBO. Multiple vehicle coordination for the rendezvous problem is achieved by introducing an outer-loop which adjusts each vehicle's speed profiles in response to information exchanged over a communication network. First, bi-directional communication between vehicles is considered within a leader-follower architecture. Then a virtual leader approach is considered for uni-directional communication between vehicles. The uni-directional loses some mission flexibility by imposing the constraint that each vehicle knows a reference speed profile *a priori*, but allows for a wider range of communication topologies.

6.2 Recommendations For Future Work

Multiple obstacles and non-symmetric obstacles are not considered. Conjecture 1, which is not proven, would provide a theoretical guarantee that the obstacle avoidance scheme diverts the path from obstacles. In order to consider multiple obstacles, a more sophisticated de-confliction law must be developed, and bounds on obstacle separation must be derived. The obstacle avoidance path deformation must be expanded to account for multiple deformations. This requires significant extension of the method provided in this work. Furthermore, visual measurements are assumed to be available as long as an obstacle is within the sensing range of a vehicle. Future work directed at including missing measurements from occlusion of obstacles and/or poor lighting can be addressed. Time-dependent communication topologies are not considered, which occurs more frequently in realistic applications. An extension of the method in [43, 44] can be made that allows vehicles to “agree” on a new leader in the event that the leader vehicle is lost to complete the mission. This will relax the assumption on the need of *a priori* knowledge of the reference velocity for all vehicles and give more flexibility to the entire mission.

Bibliography

- [1] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, Dover Publishing Company, 1965.
- [2] V. Akulov, V. Yu, and A. Schisljonok. Foundation of a New Direct Method for Trajectories Optimization. *Air Force Engineering Academy Press*, pp. 1 - 11, Moscow, Russia, 1991.
- [3] J. Asensio, J. Montiel, and L. Montano. Goal Directed Reactive Robot Navigation with Relocation Using Laser and Vision. *In Proceedings of IEEE International Conference on Robotics and Automation*, 2905 - 2910, Detroit, MI, 1999.
- [4] R. Balakrishnan and K. Ranganathan. *A Textbook of Graph Theory*, Springer Publishing Company, 2000.
- [5] J. Bang-Jensen and G. Gutin. *Digraph theory, Algorithms and Applications*, Springer Publishing Company, 2002.
- [6] R. Beard, J. Lawton, and F. Hadaegh. A Coordination Architecture for Spacecraft Formation Control. *IEEE Transactions on Control Systems Technology*, 9: 777 - 790, 2001.
- [7] N. Biggs. *Algebraic graph theory*, Cambridge University Press, 1996.
- [8] M. Blagodarniy. Direct Method for N-Point Boundary Value Problems. *Air Force Engineering Academy Press*, pp. 3 - 17, Moscow, Russia, 1989.

- [9] A. Bowling and O. Khatib. Design of Non-Redundant Manipulators for Optimal Dynamic Performance. *In Proceedings of International Conference on Advanced Robots*, pp. 865 - 872, Monterey, CA, 1997.
- [10] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [11] F. Boyer and F. Lamiroux. Trajectory Optimization Applied to Kinodynamic Motion Planning for a Realistic Car Model. *IEEE International Conference on Robotics and Automation*, Orlando, FL, 2006.
- [12] C. Cao and N. Hovakimyan. Vision-Based Aerial Tracking Using Intelligent Excitation. *In Proceedings of American Control Conference*, pp. 5091 - 5096, Portland, OR, 2005.
- [13] H. Christophersen, R. Pickell, J. Neidhoefer, A. Koller, S. Kannan, and E. Johnson. A Compact Guidance, Navigation, and Control System for Unmanned Aerial Vehicles. *Journal of Aerospace Computing, Information, and Communication*, 3(5): 187 - 213, 2006.
- [14] J. Cortes and F. Bullo. Coordination and Geometric Optimization via Distributed Dynamical Systems. *SIAM Journal on Control and Optimization*, 44(5): 15431574, 2005.
- [15] P. Costa. Adaptive Model Architecture and Extended Kalman-Bucy Filters. *IEEE Transaction on Aerospace and Electronic Systems*, 30: 525 - 533, 1994.
- [16] A. Dippold, L. Ma, and N. Hovakimyan. Vision-Based Obstacle Avoidance of Wheeled Robots Using Fast Estimation. *In Proceedings of AIAA Guidance, Navigation, and Control Conference*, AIAA-2008-7449, Honolulu, HI, 2008.
- [17] V. Dobrokhodov, I. Kaminer, K. Jones, I. Kitsios, C. Cao, L. Ma, N. Hovakimyan, and C. Woolsey. Rapid Motion Estimation of a Target Moving with Time-Varying Velocity. *In Proceedings of AIAA Guidance, Navigation, and Control Conference*, AIAA-2007-6746, Hilton Head, SC, 2007.

- [18] E. Einhorn, C. Schröter, H. Böhme, and H. Gross. A Hybrid Kalman Filter Based Algorithm for Real-time Visual Obstacle Detection. *Proceedings of the European Conference on Mobile Robots*, 156 - 161, Freiburg, Germany, 2007.
- [19] L. Elsgolts. *Calculus of Variations*. Pergamon Press, London, UK, 1961.
- [20] L. Euler. *Institutiones Calculi Integralis*, Academy of Sciences, St. Petersburg, Russia, 1770.
- [21] D. Farina and B. Ristic. Tracking a Ballistic Target: Comparison of Several Nonlinear Filters. *IEEE Transactions on Aerospace and Electronic Systems*, 38: 854 - 867, 2002.
- [22] J. Fax and R. Murray. Information Flow and Cooperative Control of Vehicle Formations. *IEEE Transactions on Automatic Control*, 49(9): 1465 - 1476, 2004.
- [23] D. Fox, W. Burgard, and S. Thrun. The Dynamic Window Approach to Collision Avoidance. *IEEE Robotics & Automation Magazine*, 4(1): 23 - 33, 1997.
- [24] B. Galerkin. Series Development for Some Cases of Elastic Equilibrium of the Rods and Plates. *Bulletin of Engineers and Technicians*, 19: 897 - 908, 1915.
- [25] G.Gerten. Protecting the Global Positioning System. *Aerospace and Electronic Systems Magazine*, 20(11): 3 - 8, 2005.
- [26] R. Ghabcheloo, A. Pascoal, C. Silvestre, and I. Kaminer. Coordinated Path Following Control of Multiple Wheeled Robots with Directed Communication Links. *In Proceedings of IEEE Conference on Decision and Control*, 7084 - 7089, Seville, Spain, 2005.
- [27] R. Ghabcheloo, A. Pascoal, C. Silvestre, and I. Kaminer. Coordinated Path Following Control of Multiple Wheeled Robots Using Linearization Techniques. *International Journal of Systems Science*, 37(6): 399 - 414, 2006.
- [28] R. Ghabcheloo, A. Pascoal, C. Silvestre, and I. Kaminer. Non-linear Co-ordinated Path Following Control of Multiple Wheeled Robots with Bidirectional Communication Con-

- straints. *International Journal of Adaptive Control and Signal Processing*, 21: 133 - 157, 2006.
- [29] R. Ghabcheloo, A. Aguiar, A. Pascoal, and I. Kaminer. Time-Coordinated Path Following: Simultaneous Arrival and Conflict Resolution. *Submitted to American Control Conference*, 2009.
- [30] R. Ghabcheloo. Coordinated Path Following of Multiple Autonomous Vehicles. *Ph.D Dissertation at Technical University of Lisbon*, Portugal, 2007.
- [31] B. Ghosh, E. Loucks, and M. Jankovic. An Introduction to Perspective Observability and Recursive Identification Problems in Machine Vision. *In Proceedings of IEEE Conference on Decision and Control*, pp. 3329 - 3234, 1994.
- [32] B. Ghosh, M. Jankovic, and E. Loucks. On the Problem of Observing Motion and Shape. *In Proceedings of IEEE Conference on Automation, Robotics, and Computer Vision*, pp. 653 - 657, 1994.
- [33] C. Godsil and G. Royle. *Algebraic Graph Theory*, Springer-Verlag New York, Inc., 2001.
- [34] D. Greenwood. *Principles of Dynamics*. Prentice Hall, 1988.
- [35] W. Guan, F. Hsiao, and C. Ho. Development of Low-Cost Differential Global Positioning System for Remotely Piloted Vehicles. *Journal of Aircraft*, 36(4): 617 - 625, 1999.
- [36] J. Hespanha, O. Yakimenko, I. Kaminer, and A. Pascoal. Linear Parametrically Varying Systems with Brief Instabilities: An Application to Integrated Vision/IMU Navigation *IEEE Transactions on Aerospace and Electronics*, 40(4): 889 - 902, 2004.
- [37] M. Hillion and F. Lamiroux. Taking Into Account Velocity and Acceleration Bounds in Nonholonomic Trajectory Deformation. *IEEE International Conference on Robotics and Automation*, Roma, Italy, 2007.
- [38] R. Horn and C. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

- [39] T. Hu, A. Kahng, and G. Robins. Optimal robust path planning in general environments. *IEEE Transactions on Robotics and Automation*, 9(6): 775 - 784, 1993.
- [40] A. Jadbabaie, J. Lin, and A. Morse. Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules. *IEEE Transactions on Automatic Control*, 48(6): 988 - 1001, 2003.
- [41] M. Jankovic and B. Ghosh. Visually Guided Ranging From Observations of Points, Lines, and Curves Via an Identifier Based Nonlinear Observer. *Systems & Control Letters*, 25: 63 - 73, 1995.
- [42] I. Kaminer, O. Yakimenko, A. Pascoal, and R. Ghabcheloo. Path Generation, Path Following, and Coordinated Control for Time Critical Missions of Multiple UAVs. *In Proceedings of American Control Conference*, pp. 4906 - 4913, Minneapolis, MN, 2006.
- [43] I. Kaminer, O. Yakimenko, V. Dobrokhodov, A. Pascoal, N. Hovakimyan, A. Young, C. Cao, and V. Patel. Coordinated Path Following for Time-Critical Missions of Multiple UAVs via \mathcal{L}_1 Adaptive Output Feedback Controllers. *In Proceedings of AIAA Guidance, Navigation, and Control Conference*, AIAA-2007-6409, Hilton Head, SC, 2007.
- [44] I. Kaminer, N. Hovakimyan, V. Patel, C. Cao, A. Young, A. Pascoal, and V. Dobrokhodov. Time-Critical Coordinated Path Following for Multiple UAVs via \mathcal{L}_1 Adaptive Output Feedback Controllers. *In Proceedings of European Control Conference*, Kos, Greece, 2007.
- [45] L. Kantorovich and V. Krylov. *Approximate Methods of Higher Analysis*. Interscience Publishers, Inc., New York, NY, 1958.
- [46] H. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.
- [47] O. Khatib. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *International Journal of Robotics Research*, 5(1): 90 - 98, 1986.

- [48] O. Khatib and J. Burdick. Motion and Force Control of Robot Manipulators. *In Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1381 - 1386, 1986.
- [49] D. Kirk. *Optimal Control Theory, An Introduction*. Prentice Hall, 1970
- [50] F. Kleibergen and R. Paap. Generalized Reduced Rank Tests Using the Singular Value Decomposition. *Journal of Econometrics*, 133(1): 97 - 126.
- [51] N. Krylov. Les Méthodes de Solution Approchée des Problèmes de la Physique Mathématique. *Mémorial des Sciences Mathématiques*, fascicule 49, Paris, France, 1931.
- [52] F. Lamiroux, D. Bonnafous, and O. Lefebvre. Reactive Path Deformation for Nonholonomic Mobile Robots. *IEEE Transactions on Robotics*, 20(6): 967 - 977, December, 2004.
- [53] J. Langelaan and S. Rock. Navigation of Small UAVs Operating in Forests. *In Proceedings of AIAA Guidance, Navigation, and Control Conference*, AIAA-2004-5140 , Providence, RI, 2004.
- [54] J. Langelaan. State Estimation for Autonomous Flight in Cluttered Environments. *Journal of Guidance, Control, and Dynamics*, 30(5): 1414 - 1426, 2007.
- [55] L. Lapierre, R. Zapata, and P. Lepinay. Simultaneous Path Following and Obstacle Avoidance Control of a Unicycle-type Robot. *In Proceedings of IEEE International Conference on Robotics and Automation*, pp. 2617 - 2622, Roma, Italy, 2007.
- [56] J. Laumond. Robot Motion Planning and Control. *Lecture Notes in Control and Information Sciences*, Springer, New York, NY, 1998.
- [57] O. Lefebvre, F. Lamiroux, and D. Bonnafous. Fast Computation of Robot-Obstacle Interactions in Nonholonomic Trajectory Deformation. *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

- [58] L. Ledsinger. Solutions to Decomposed Branching Trajectories with Powered Flyback Using Multidisciplinary Design Optimization. *PhD dissertation*, Georgia Institute of Technology, School of Aerospace Engineering, July 2000.
- [59] Z. Lin, B. Francis, and M. Maggiore. State Agreement for Coupled Nonlinear Systems with Time-Varying Interaction. *SIAM Journal of Control and Optimization*, 46: 288 - 307, 2007.
- [60] L. Ljung. Asymptotic Behavior of the Extended Kalman Filter as a Parameter Estimator for Linear Systems. *IEEE Transactions on Automatic Control*, 24(1): 36 - 50, 1979.
- [61] L. Ma, C. Cao, N. Hovakimyan, C. Woolsey, and W. Dixon. Fast Estimation for Range Identification in the Presence of Unknown Motion Parameters. *In Proceedings of International Conference on Informatics in Control, Automation and Robotics*, Angers, France, 2007.
- [62] L. Ma, C. Cao, N. Hovakimyan, V. Dobrokhodov, and I. Kammer. Adaptive Vision-Based Guidance Law with Guaranteed Performance Bounds for Tracking a Ground Target with Time-Varying Velocity. *In Proceedings of AIAA Guidance, Navigation, and Control Conference*, AIAA-2008-7445, Honolulu, HI, 2008.
- [63] M. Mesbahi and F. Hadaegh. Formation Flying Control of Multiple Spacecraft via Graphs, Matrix Inequalities, and Switching. *Journal of Guidance, Control and Dynamics*, 24(2): 369 - 377, 2001.
- [64] A. Micaelli and C. Samson. Trajectory Tracking for Unicycle-Type and Two-Steering-Wheels Mobile Robots. *Technical Report No. 2097*, INRIA, Sophia-Antipolis, France, 1993.
- [65] L. Moreau. Stability of Multiagent Systems with Time - Dependent Communication Links. *IEEE Transactions on Automatic Control*, 50(2): 169 - 182, 2005.

- [66] A. Morgan and K. Narendra. On the Stability of Nonautonomous Differential Equations $\dot{x} = [A + B(t)]x$ With Skew Symmetric Matrix $B(t)$. *SIAM Journal of Control*, 15: 163 - 176, 1977.
- [67] K. Narendra and A. Annaswamy. *Stable Adaptive Systems*, Prentice Hall, Englewood Cliffs, NJ, 1989.
- [68] A. Neljubov. Mathematical Methods of Air Vehicle with Thrust Vector Turn Possibility Battle, Takeoff, and Landing Maneuvers Calculation. *Air Force Engineering Academy Press*, Moscow, Russia, 1986.
- [69] A. Neljubov. Mathematical Methods of Calculus of Combat, Takeoff/Climb, and Landing Approach Maneuvers for the Aircraft with 2-D Thrust Vectoring, Flight Characteristics and Combat Maneuvering of Manned Vehicles. *Air Force Engineering Academy Press*, pp. 1 - 17, Moscow, Russia, 1986.
- [70] B. d'Andrea-Novell, G. Bastin, and G. Campion. Dynamic Feedback Linearization of Nonholonomic Wheeled Mobile Robots. *In Proceedings of IEEE International Conference on Robots and Automation*, pp. 2527 - 2532, Nice, France, 1992.
- [71] P. Ogren, M. Egerstedt, and X. Hu. A Control Lyapunov Function Approach to Multiagent Coordination. *IEEE Transactions on Robotics and Automation* 18: 847 - 851, 2002.
- [72] P. Ogren and N. Leonard. A Convergent Dynamic Window Approach to Obstacle Avoidance. *IEEE Transactions on Robotics and Automation*, 21(2): 188 - 195, 2005.
- [73] S. Oh and E. Johnson. Relative Motion Estimation for Vision-Based Formation Flight Using Unscented Kalman Filter. *In Proceedings of AIAA Guidance, Navigation, and Control Conference*, AIAA-2007-6866, Hilton Head, SC, 2007.

- [74] A. Pascoal, C. Silvestre, and P. Oliveira. Vehicle and Mission Control of Single and Multiple Autonomous Marine Robots. *IEE Control Engineering Series*, pp. 353 - 386, 2006.
- [75] F. Pereira and J. Borges de Sousa. Coordinated Control of Networked Vehicles: An Autonomous Underwater System. *Automation and Remote Control*, 65(7):1037 - 1045, 2004.
- [76] Y. Petillot, I. Ruiz, and D. Lane. Underwater Vehicle Obstacle Avoidance and Path Planning Using a Multi-Beam Forward Looking Sonar. *IEEE Journal of Oceanic Engineering*, 26(2): 240 - 251, 2001.
- [77] L. Piegl. Interactive Data Interpolation by Rational Bezier Curves. *IEEE Computer Graphics and Applications*, 7(4):45 - 58, 1987.
- [78] J. Pomet and L. Praly. Adaptive Nonlinear Regulation: Estimation From the Lyapunov Equation. *IEEE Transactions on Automatic Control*, 37(6): 729 - 740, 1992.
- [79] S. Quinlan and O. Khatib. Elastic bands: Connecting Path Planning and Control. *In Proceedings of IEEE International Conference on Robotics and Automation*, pp. 802 - 807, 1993.
- [80] J. Redding, J. Amin, and J. Boskovic. A Real-Time Obstacle Detection and Reactive Path Planning System for Autonomous Small-Scale Helicopters. *In Proceedings of AIAA Guidance, Navigation, and Control Conference*, AIAA-2007-6413, Hilton Head, SC, 2007.
- [81] C. Ranieri and C. Ocampo. Optimization of Roundtrip, Time-Constrained, Finite Burn Trajectories Via an Indirect Method. *In Proceedings of AIAA Astrodynamics Specialists Conference*, AAS03-572, Big Sky, MO, 2003.
- [82] C. Reynolds. Flocks, Herds and Schools: a Distributed Behavioral Model. *Computer Graphics*, 21(4): 71 - 87, 1987.

- [83] E. Rimon and D. Koditschek. Exact Robot Navigation Using Artificial Potential Functions. *IEEE Transactions on Robotics and Automation*, 8: 501 - 518, 1992.
- [84] W. Ritz. Über eine Neue Methode zur Lösung Gewisser Variationsprobleme der Mathematischen Physik. *Journal für Reine und Angewandte Mathematik*, 135: 1 - 61, 1908.
- [85] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill Science/Engineering/Math, 1976.
- [86] J. Sklar. Interference Mitigation Approaches for the Global Positioning System. *Lincoln Laboratory Journal*, 14(2): 167 - 180.
- [87] D. Soetanto, L. Lapierre, and A. Pascoal. Adaptive Non-Singular Path Following Control of Dynamic Wheeled Robots. *In Proceedings of International Conference on Informatics in Control, Automation, and Robotics*, Coimbra, Portugal, 2003.
- [88] B. Sridhar and R. Suorsa. Vision-Based Obstacle Detection for Rotocraft Flight. *Journal of Robotic Systems*, 9: 709 - 727, 1992.
- [89] D. Stilwell and B. Bishop. Platoons of Underwater Vehicles. *IEEE Control Systems Magazine*, pp. 45 - 52, 2000.
- [90] D. Stilwell, E. Boltt, and D. Roberson. Sufficient Conditions for Fast Switching Synchronizations in Time-Varying Network Topologies. *SIA Journal of Applied Dynamical Systems*, 6(1): 140 - 156, 2006.
- [91] R. Suorsa and B. Sridhar. A Parallel Implementation of a Multisensor Feature-Based Range-Estimation Method. *IEEE Transactions on Robotics and Automation*, 10(6): 755 - 768, 1994.
- [92] V. Taranenko. Experience of Ritz's, Puankare's, and Lyapunov's Methods Utilization for Flight Dynamics Tasks Solution. *Air Force Engineering Academy Press*, Moscow, Russia, 1968.

- [93] V. Taranenکو and V. Momdgi. Direct Variational Method in Boundary Tasks of Flight Dynamics. *Mashinostroenie*, Moscow, Russia, 1968.
- [94] M. Verhaegen. Round-Off Error Propagation in Four Generally-Applicable, Recursive, Least-Squares Estimation Schemes. *Automatica*, 25(3): 437 - 444, 1989.
- [95] Y. Watanabe, E. Johnson, and A. Calise. Vision-based Approach to Obstacle Avoidance. *In Proceedings of AIAA Guidance, Navigation, and Control Conference*, AIAA 2005-6092, San Francisco, CA, 2005.
- [96] Y. Watanabe, A. Calise, and E. Johnson. Vision-Based Obstacle Avoidance for UAVs. *In Proceedings of AIAA Guidance, Navigation, and Control Conference*, AIAA-2007-6829 , Hilton Head, SC, 2007.
- [97] O. Yakimenko. Direct Method for Rapid Prototyping of Near-Optimal Aircraft Trajectories. *AIAA Journal of Guidance, Control, and Dynamics*, 23(5): 865 - 875, 2000.
- [98] P. Zarchan and J. Alpert. Using Filter Banks to Improve Interceptor Performance Against Weaving Targets. *In Proceedings of AIAA Guidance, Navigation, and Control Conference*, AIAA 2006-6700, Keystone, CO, 2006.