

ABSTRACT

Title of thesis: REACHING A TARGET WITHIN
A GPS-DENIED OR COSTLY AREA:
A TWO-STAGE OPTIMAL CONTROL APPROACH

Sheng Cheng
Master of Science, 2018

Thesis directed by: Professor Nuno C. Martins
Department of Electrical and Computer Engineering

In this thesis, a new class of problem is studied where a mobile agent is controlled to reach a target. Especially, the target is enclosed within a special area. The presence of this area requires a controller to have two stages: the outer stage steers the mobile agent to enter such area while the inner stage steers the mobile agent towards the target.

We consider two types of the special area: a time-costly area and a GPS-denied area. For the time-costly area, we formulate a two-stage optimal control problem where time is explicitly specified in the cost function. We solve the problem by solving its subproblems. The key subproblem is a nonconvex quadratic programming with two quadratic constraints (QC2QP). We study the QC2QP independently and prove the necessary and sufficient conditions for strong duality in a general QC2QP. Such conditions enable efficient solution methods for a QC2QP utilizing its dual and semidefinite relaxation. For the GPS-denied area, we formulate another two-stage optimal control problem where perturbation is considered. To deal with the pertur-

bation, we propose a robust controller using the variable horizon model predictive control. The performance of the two-stage controller for each type of the special area is demonstrated in simulations.

We construct and implement a two-stage controller that can steer a quadrotor to reach a target enclosed within a denied area. Such controller utilizes the formulation and solution methods in the theoretical study. We show experimental results where the controller can run in real-time using off-the-shelf fast optimization solvers. We also conduct a bat experiment to learn bat's strategy for target reaching inside a denied area.

REACHING A TARGET WITHIN
A GPS-DENIED OR COSTLY AREA:
A TWO-STAGE OPTIMAL CONTROL APPROACH

by

Sheng Cheng

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2018

Advisory Committee:
Professor Nuno C. Martins, Chair/Advisor
Professor Richard J. La
Professor Nikhil Chopra

© Copyright by
Sheng Cheng
2018

I dedicate this thesis to my parents.

Acknowledgments

I want to thank Professor Nuno Martins for being an enlightening, supportive, rigorous, and knowledgeable advisor during my MS studies. I would like to thank Professor Richard La and Nikhil Chopra for agreeing to serve on my thesis committee. I would also like to express my gratitude to Professor Cynthia F. Moss, Dr. Angeles Salles, and Dr. Jinhong Luo from Johns Hopkins University for their efforts on conducting the bat experiment. I want to thank Professor Wenbao Ai from Beijing University of Posts and Telecommunications for his foundation work in characterizing strong duality in the CDT subproblem. I am also thankful to the research opportunity funded by AFOSR Grant FA95501510367, the Air Force Center of Excellence: Nature Inspired Flight Technologies and Ideas (NIFTI).

I want to thank Professor Prakash Narayan, Michael Rotkowitz, Andre Tits, P.S. Krishnaprasad, and Patrick Fitzpatrick for the excellent courses they've guided me through. I am also grateful to the following university staff: Melanie Prange, Vivian Lu, Bill Churma, Emily Irwin, Maria Hoo, and Heather Stewart for their unfailing support and assistance.

I want to thank my girlfriend Jinuan for being my safe harbor and supporting me whenever life is happy or tough. I'm sincerely grateful to my parents, Dr. Cheng and Mrs. Li, for being my strongest backing and selfless love.

And finally, I want to thank my friends I made during all these years at Maryland: Yuntao Liu, Shang Li, Qiang Zhu, Feng Zhang, Peng Zan, Aneesh Raghavan, Ariyan Kabir, Pradeep Rajendran, Debdipta Goswami, Yunchuan Li, Zhenyu Lin,

Michael Lin, Reza Hadadi, Marcos Vasconcelos, Shinkyu Park, David Ward, Xiaoxu Meng, Xiaomin Lin, Yunlong Huang, Udit Halder, Vidya Raju, Jian Cheng, Yinjun Huang and Ahmed Arafa.

Table of Contents

Dedication	ii
Acknowledgements	iii
List of Tables	vii
List of Figures	viii
List of Abbreviations	x
1 Introduction	1
1.1 Motivation	2
1.2 Contributions	3
1.3 Notation	4
1.4 Outline	5
2 Bat Experiment	7
2.1 Creating a denied area	9
2.2 Experiment setup and procedure	10
2.3 Results and discussion	12
3 Strong Duality in General Quadratic Programming with Two Quadratic Constraints	14
3.1 Related literature	14
3.2 Main results	16
3.3 Relation between Theorem 3.1 and Theorem 3.2	26
4 Two-stage Optimal Control for Target Reaching Inside a Time-costly Area	28
4.1 Related literature	29
4.2 Problem formulation	29
4.3 Main results	34
4.3.1 The inner stage problem	34
4.3.2 Augmented outer stage problem	40

4.3.3	Solution of the original problem (P)	48
4.4	Numerical example	49
5	Two-stage Optimal Control for Target Reaching Inside a Denied Area	52
5.1	Related literature	52
5.2	Discrete-time two-stage optimal control problem	53
5.3	Problem formulation with perturbation	57
5.4	Robust controller using variable horizon model predictive control	66
5.5	Simulation result	76
6	Quadrotor Experiment	82
6.1	Introduction	82
6.2	AscTec Hummingbird	83
6.2.1	Dynamics and mathematical model	84
6.2.2	Simulink model	87
6.2.3	Quadrotor controller	88
6.2.3.1	Attitude control	90
6.2.3.2	Position control	92
6.2.3.3	Luenberger observer	93
6.2.4	Software and integration	96
6.2.5	Controller performance	96
6.3	Ar.Drone	100
6.3.1	Dynamics and mathematical model	100
6.3.2	Quadrotor controller	101
6.3.3	Software and integration	102
6.3.4	Controller performance	102
6.4	Denied area experiment	105
6.4.1	Problem formulation	105
6.4.2	Outer stage controller	106
6.4.3	Inner stage controller	107
6.4.4	System identification	109
6.4.5	Perturbation set determination	110
6.4.6	Optimization solver	110
6.4.7	Experiment results	111
7	Conclusion and Future Work	115
	Bibliography	118

List of Tables

2.1	Number of successful trials versus total trials	12
-----	---	----

List of Figures

1.1	Scenarios where a target resides in the special area	3
2.1	Spectrogram of echolocation pulses	8
2.2	Illustration of the denied area to the bat	9
2.3	Setup of the bat experiment	11
2.4	Successful target reaching	13
4.1	Illustration of the problem	29
4.2	Relation between $J_i^*(t_f)$ and the behavior of the optimal trajectory	40
4.3	Minimum cost and terminal time with bisection intervals	50
4.4	Optimal values of (P3)	50
4.5	Optimal trajectory	51
5.1	Illustration of failed controller subject to perturbation	58
5.2	Geometric illustration of (5.20) and (5.21)	62
5.3	Geometric illustration of (5.24)	64
5.4	Annular region to place the PSSS	71
5.5	Trajectory of a successful trial	78
5.6	Planned trajectories at all steps of a successful trial	79
5.7	Optimal switching time and optimal cost	80
5.8	Collection of 16 successful trials	80
5.9	Planned trajectories at all steps of an unsuccessful trial	81
6.1	Coordinate systems of AscTec Hummingbird	84
6.2	Nested control loops for AscTec Hummingbird	88
6.3	Simulink diagram of the nonlinear dynamics for the quadrotor	89
6.4	Illustration of packet loss	97
6.5	Performance of the AscTec Hummingbird controllers	98
6.6	Velocity estimations of AscTec Hummingbird	99
6.7	Coordinate systems of Ar.Drone	101
6.8	Performance of Ar.Drone hover controller	103
6.9	Performance of Ar.Drone trajectory controller	104
6.10	Size of the approximate propoagated set of perturbation	111

6.11 Trajectory of the Ar.Drone in an experiment	113
6.12 Attitude commands and measurement in the inner stage	114

List of Abbreviations

GPS	Global Positioning System
PSSS	Predicted set of switching states
PSTS	Predicted set of terminal states
QC2QP	Quadratic programming with two quadratic constraints
W.L.O.G.	Without loss of generality

Chapter 1: Introduction

The advancement of technology in recent years has boosted the development of mobile robots. Sensors with various modalities enable robots to perceive the environment more precisely. Numerous types of actuators have been built to arm the robot for different tasks. The recent breakthrough in artificial intelligence further extends robot's understanding of its surroundings and empower robots to interact closely with people. And, last but not least, powerful computing resources are able to process complicated algorithms that were inapplicable in the past due to limited computing capability.

So far, controlling a mobile robot to reach a destination or a target is not a difficult task: a mobile platform for motion, a camera or a lidar for environment detection and localization, and a path planning algorithm for control will be enough, in general. Though the path towards the target might be curvy and zigzag, the robot can eventually reach there when the target resides in a homogeneous environment as its surroundings. What has not been considered is when the target resides in a special area. The special area may render the robot in critical conditions if staying there for too long. Hence, the time spent from entering such area to arriving at the target is limited. Or, the special area may forbid the robot to localize itself, which

essentially disables any feedback-based controllers to steer the robot as expected since perturbations are imperceivable to the controller.

In this thesis, we consider two types of the special area where the target resides in: a time-costly area and a denied area. The latter, within which no localization is available, is a generalization of the GPS-denied area. We formulate the problem as a two-stage optimal control problem and propose solution methods. We then apply the theoretical results to construct a quadrotor controller and test it in experiments. We also conduct a bat experiment to learn bat's strategy for reaching a target that resides in a denied area.

1.1 Motivation

Our motivation comes from real-life scenarios. One of the scenarios involves a helicopter rescue, as shown in Figure 1.1(A). When a helicopter needs to land for a rescue mission, all the other flying vehicles within certain distance to the landing site have to either leave the area or pause their current tasks so that the helicopter can land safely. We name such area by the time-costly area. The helicopter will incur a cost associated with the elapsed time between its entry to the perimeter of the landing site and its landing. Such cost comes from consumed fuels of the rescue helicopter and the delayed tasks of other flying vehicles.

Another scenario is the following. A terrorist is in a known location in some building, and we want to send a drone inside the building to eliminate the terrorist target. The drone must be able to plan its maneuvers towards the terrorist inside

the building, where GPS localization is denied, before its entry to the building. This scenario is depicted in Figure 1.1(B).

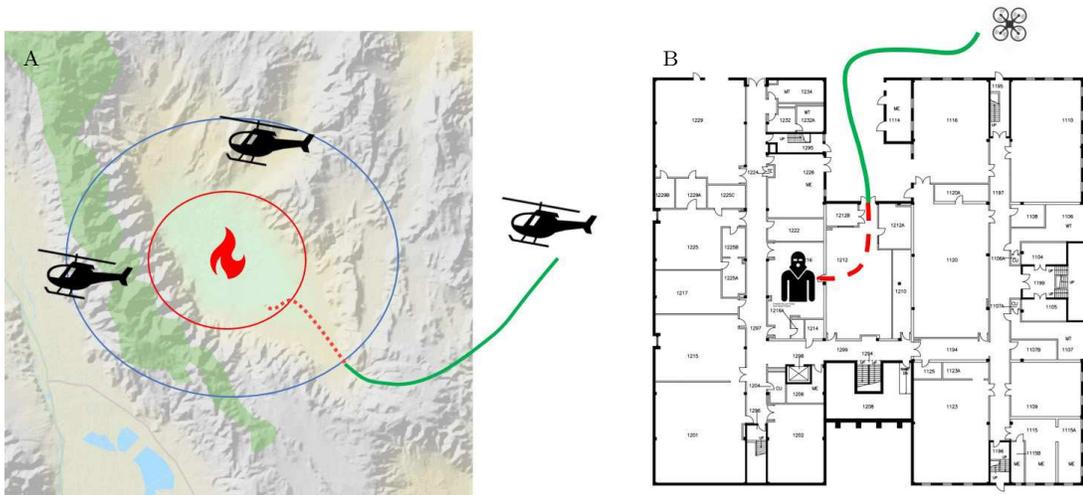


Figure 1.1: Scenarios where a target resides in the special area. (A) The target (marked by the red circle) is enclosed by the area which is time-costly (marked by the blue circle). (B) The target (terrorist) is inside the building where GPS localization is denied.

1.2 Contributions

The contributions of this work are summarized as follows.

1. A new class of problem has been formulated where a mobile agent is steered to reach a target that is enclosed within a special area. The presence of such area requires the controller to have two stages, as such area places specific requirements on the controller inside the area.
2. In the optimization problem with deterministic dynamics, we transform a key subproblem to a quadratic programming with two quadratic constraints

(QC2QP). We show the necessary and sufficient conditions for strong duality in a general QC2QP. Such result is an extension to earlier results on a special case of QC2QP, where one of the quadratic constraints is required to be strictly convex.

3. We propose a robust controller that deals with the perturbation in the system dynamics. Such controller handles stage switching and target reaching, subject to unknown perturbations, by solving a variable horizon model predictive control problem.
4. We conduct two experiments associated with the denied area. One studies the strategy taken by a bat to reach a target enclosed within a denied area. This provides data and reference for the future study on bio-inspired control and motion planning. Another applies the theoretical results reported in this thesis to build a controller that can steer a quadrotor to reach a target enclosed within a denied area. The simple formulation empowers the quadrotor controller to run in real-time using off-the-shelf fast solvers.

1.3 Notation

Throughout the paper, we adopt the following notations. The set of real numbers and the set of nonnegative real numbers are denoted by \mathbb{R} and \mathbb{R}_+ , respectively. The set of n -dimensional symmetric matrices is denoted by \mathcal{S}^n . We use $\text{int}(\mathcal{A})$ and $\partial\mathcal{A}$ to denote the interior and the boundary, respectively, of a set \mathcal{A} . We use \oplus to denote the Minkowski sum between two sets and \bigoplus to denote the Minkowski sum

of multiple sets. We use the dot notation to denote the matrix inner product, i.e., $A \cdot B \stackrel{\text{def}}{=} \text{Tr}(AB)$, for $A, B \in \mathcal{S}^n$. We use c_x , s_x , and t_x denote $\cos(x)$, $\sin(x)$, and $\tan(x)$, respectively. The null space of a linear mapping $L : \mathbb{V} \rightarrow \mathbb{W}$ between two vector spaces \mathbb{V} and \mathbb{W} is denoted by $\mathcal{N}(L)$. We use $0_{k \times j}$ to denote a matrix in $\mathbb{R}^{k \times j}$ with all entries being zero and I_k to denote a k -dimensional identity matrix. The matrix W^T is the transpose of W . We use notations $W \succ 0$ and $V \succeq 0$ to denote a symmetric positive definite matrix W and a symmetric positive semidefinite matrix V , respectively. The D -square of a vector $z \in \mathbb{R}^n$ is represented by $\|z\|_D^2$, i.e., $\|z\|_D^2 \stackrel{\text{def}}{=} z^T D z$, with $D \in \mathcal{S}^n$. A sequence of matrices $\{A_1, \dots, A_n\}$ is denoted by $\mathbf{A}_{1:n}$. The convex hull of vectors $\mathbf{v}_{1:m}$ is denoted by $\text{convh}\{v_1, \dots, v_m\}$. For $a, b \in \mathbb{R}$, we denote a approaches b from left by $a \rightarrow b^-$ and from right by $a \rightarrow b^+$. The optimal value of an optimization problem

$$\begin{aligned} & \underset{x \in \mathcal{C}}{\text{minimize}} && f(x) \\ & \text{subject to} && h(x) \leq 0, \end{aligned} \tag{P0}$$

is denoted by $\text{val}(\text{P0})$.

1.4 Outline

The thesis is structured in seven chapters, including this introduction. The rest of the thesis is organized as follows.

Chapter 2 presents the experiment conducted on a bat which is required to reach a target enclosed within a denied area. We introduce how an artificial denied area is created and the procedure of the experiment. We also show the result and

its relation to the later theoretical study.

Chapter 3 presents the necessary and sufficient conditions for strong duality for a general QC2QP. This result benefits us in solving nonconvex QC2QPs in Chapter 4 and Chapter 6, by utilizing strong duality.

Chapter 4 presents a framework of controlling a mobile agent to reach a target that is enclosed within a time-costly area. We define the time-costly area and formulate the problem as a two-stage optimal control problem. A solution method is provided by solving subproblems. We validate the optimal controller in a numerical example.

Chapter 5 presents an extension to the results in Chapter 4. We consider a more realistic case where perturbation appears in the dynamics and consider another type of the special area, the denied area. We reformulate the two-stage optimal control problem to deal with perturbations. And we propose a robust controller based on the new formulation using the variable horizon model predictive control and validate the controller in simulations.

Chapter 6 presents results of a quadrotor experiment where a quadrotor is controlled to reach a target enclosed within a denied area. We first introduce two quadrotor testbeds in the CPS and Cooperative Autonomy Laboratory and show the design as well as the performance of an attitude controller and a position controller. Then we show a formulation of the controller for target reaching inside a denied area and demonstrate its performance in experiments.

Chapter 7 concludes the thesis and outlines future research directions.

Chapter 2: Bat Experiment

The experiment in this chapter was conducted by Professor Cynthia F. Moss and Dr. Angeles Salles in the Comparative Neural Systems and Behavior Lab at Johns Hopkins University. We use the data obtained from the experiment to interpret the results.

Bats (of certain species, e.g., *Eptesicus fuscus*) are known to use echolocation to find and locate prey, even in darkness. A bat's hunting of a target insect is summarized into three phases [1]: search, approach, and capture. The three phases are discriminated by the distance between the bat and the insect as well as the pattern of the bat's echolocation pulses, especially, the rate of pulse repetitions. The search phase is when the bat is searching the environment for flying insects. In this phase, the interval between pulses is quite long, often lasting several hundred milliseconds [2]. The approach phase ensues when the bat first reacts to an insect, either by turning towards it, by increasing the pulse repetition rate, or both [1]. In this phase, the bat locates its prey more accurately. The rate of pulse repetitions increases progressively though often irregularly. The capture phase follows when the bat is fairly near the insect and emits a burst of pulses at a very high rate, the buzz. The spectrogram of echolocation pulses in these three phases is displayed in

Figure 2.1. The buzz typically has a lower frequency range, compared to the pulses in the first two stages. Previous research [3] has shown this adjustment increases bat's sonar field of view to avoid the target insect from escaping in the capture phase.

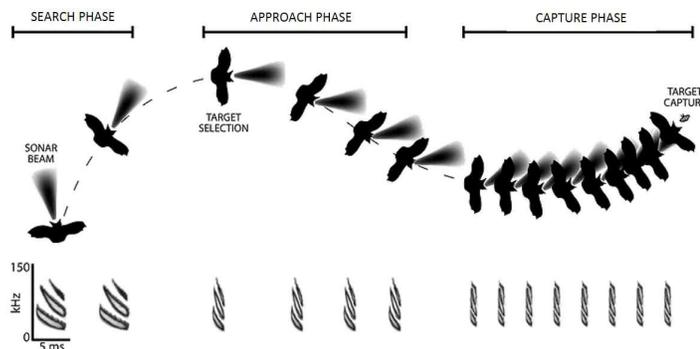


Figure 2.1: Spectrogram of a sequence of echolocation pulses produced by a European free-tailed bat in a target insect pursuit. Three phases, *search* \rightarrow *approach* \rightarrow *capture*, are marked according to the rate of pulse repetitions. Figure courtesy Dr. Melville J. Wohlgenuth.

The buzz brings limited information of the prey's location to the bat because the high frequency returning echoes overlap and interfere with each other. We define a denied area where the bat cannot locate the target insect by echolocation (nor by vision). Then a partially denied area forms near the target in the capture phase, since the information is limited. But it is not a big trouble for the bat because the area is relatively small. An interesting question is: how does the bat reach (and capture) the target insect enclosed within a denied area? What if the denied area is larger than the one in the capture phase? Will it learn to plan its future trajectory before reaching the denied area? Will it just fly straight towards the direction of

the insect in bat's memory? To answer these questions, we need to conduct a bat experiment using artificial denied area since such extreme scenario rarely exists in nature. We will enclose a target insect by an artificial denied area and record the bat's trajectory for reaching this target. The idea of the experiment is shown in Figure 2.2.

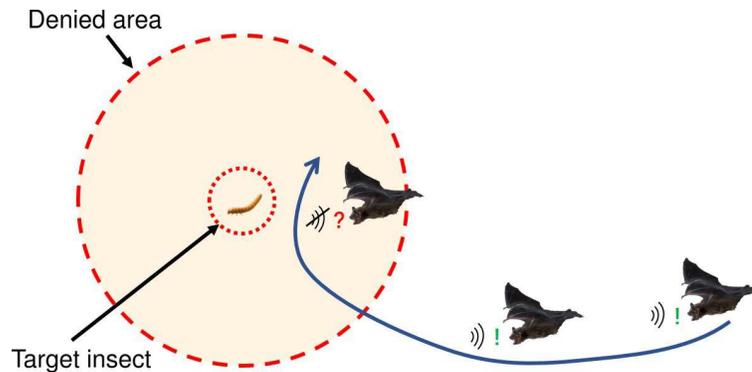


Figure 2.2: Illustration of the denied area to the bat. The target insect resides in the denied area. The bat can locate the insect only when it is outside the denied area. The exclamation mark and the question mark indicate the bat can and cannot locate the target insect, respectively.

2.1 Creating a denied area

We refer to the nature of how the denied area can be implemented. Tiger moths (*Lepidoptera: arctiidae*) can produce anti-bat sounds to respond hunting calls of bats for survival. [4] provides a list of sensory mechanisms by which the anti-bat sounds may function. This list includes startle, aversion, aposematism or warning, mimicry, phantom echo, distraction, interference, masking, and jamming. Startle, aversion, aposematism or warning, and mimicry are not suitable for creating the

denied area because they drive the bat away. Phantom echo is not suitable either because it misguides the bat somewhere else other than the insect. Interference degrades distance discrimination [5, 6], which is not enough for the denied area. Moreover, the interference signal needs to arrive within a time window in front of an echo [6] over which we have no control. Hence, interference is not suitable due to limited capability and practical difficulty. Jamming, though initially considered appropriate for creating the denied area, was not selected for the following two reasons. First, the mechanism of how jamming functions on bat’s echolocation has not been fully studied [4]. Second, the jamming signal, e.g., sinusoidal frequency modulated calls [7], is more difficult to reproduce than white noise. Therefore, masking is adopted as a strategy to create the denied area. This is achieved by broadcasting the white noise, which is triggered when the bat enters the denied area.

2.2 Experiment setup and procedure

The experiment is conducted in a room which is 6 meters wide, 6 meters long, and 2 meters tall. A mealworm (target insect) is tethered from the ceiling at the center of the room and is 1 meter above the ground. We use a customized electrostatic loudspeaker, powered by a wideband amplifier (model 7500; KrohnHite) to broadcast the white noise signal. The speaker is placed on the ground, right below the tethered mealworm. A motion capturing system containing 16 high-speed cameras (MX T40; Vicon Motion Systems) is used to record the trajectory of the

bat, at 200 frames per second. This setup is displayed in Figure 2.3.

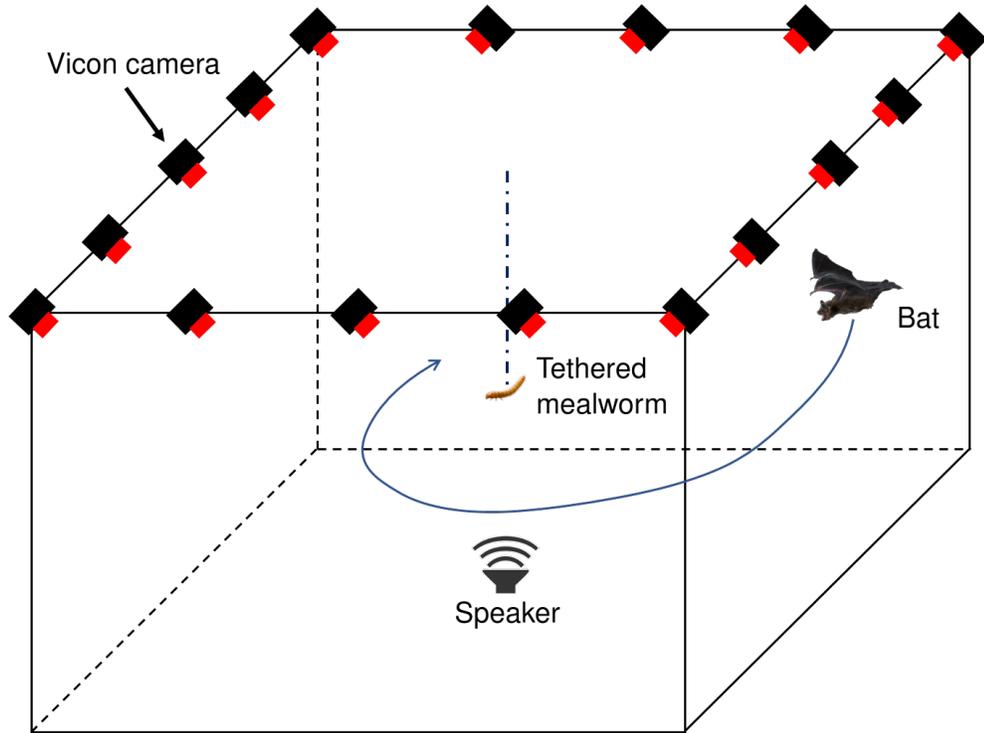


Figure 2.3: Setup of the bat experiment.

We use a single bat (*Eptesicus fuscus*), which has been trained to catch a tethered mealworm for 60 days before the experiment. In each experiment, the bat is released at random locations in the room. We choose the denied area to be a three-dimensional ball centered at the mealworm. The ball has radius 0.5 m, 0.75 m, and 1 m of which only one is selected for each experiment. A trigger will be activated when the bat is detected to be within the denied area by Vicon Motion Systems. Then this trigger decides randomly whether the speaker plays the white noise. If the speaker plays, then a denied area occurs. So the corresponding bat trajectories serve as the experiment group. The control group includes the bat trajectories when

the trigger decides to keep the speaker silent.

During the bat’s flight, the investigator can choose to record the flying trajectory for the past 2 seconds when she/he observes the bat catches or attempts a capture of the mealworm.

2.3 Results and discussion

We record in total 47 trials. The trajectories are processed offline using MATLAB (R2016b; MathWorks) where the data are smoothed using a moving average filter with a ten-sample window. We define a successful trial in which the bat’s distance to the mealworm is less than 0.1 meter, which is the body length of the bat in the experiment. A summary of trials under different radii of the denied area is displayed in Table 2.1. Due to the limited number of trials, we cannot draw any statistical conclusion on the influence of the denied area on bat’s target reaching.

Table 2.1: Number of successful trials versus total trials under different radii of the denied area

denied area radius	successful/total speaker plays white noise	successful/total speaker remains silent
0.5 m	1/5	0/7
0.75 m	1/7	1/7
1 m	3/10	2/11

However, we can still observe curved trajectories inside the denied area. Such curve may suggest that the bat is actively steering itself, instead of flying straight, towards the target. Figure 2.4 shows the trajectories of the bat when denied area has radius of 1 meter and 0.5 meter. Similar trajectories appear in the simulation

results in Chapter 4 and Chapter 5.

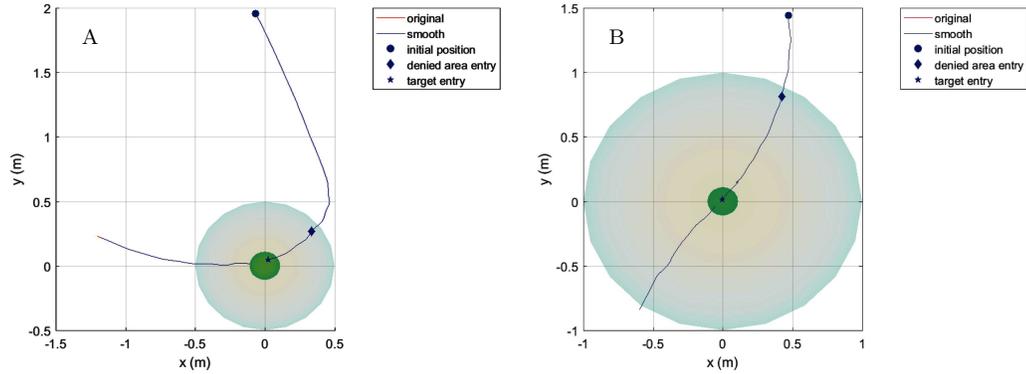


Figure 2.4: Successful target reaching under different size of the denied area. The mealworm resides in the target area, which is the green ball centered at the origin, with radius 0.1 m. The denied area is the larger ball, also centered at the origin. The diamond mark and star mark are on the boundary of the denied area and on the boundary of the target area, respectively. We only display the projection of the three-dimensional trajectories on the horizontal plane. (A) The radius of the denied area is 0.5 meter. (B) The radius of the denied area is 1 meter.

Chapter 3: Strong Duality in General Quadratic Programming with Two Quadratic Constraints

In this Chapter, we consider the following real-valued nonconvex quadratic programming with two quadratic constraints (QC2QP), which is key to the results in Chapter 4 and Chapter 6.

$$\begin{aligned} & \underset{z \in \mathbb{R}^n}{\text{minimize}} && q_0(z) = z^T Q_0 z + 2b_0^T z \\ & \text{subject to} && q_1(z) = z^T Q_1 z + 2b_1^T z + c_1 \leq 0, \\ & && q_2(z) = z^T Q_2 z + 2b_2^T z + c_2 \leq 0, \end{aligned} \tag{QP0}$$

where $Q_i \in \mathcal{S}^n$, $b_i \in \mathbb{R}^n$, $i = 0, 1, 2$, and $c_1, c_2 \in \mathbb{R}$.

3.1 Related literature

Two directions have been researched to solve (QP0) for a global solution. The first direction refers to either the semidefinite relaxation, or the Lagrange dual, or both. As the semidefinite relaxation and Lagrange dual are convex and dual of each other, (QP0) can be solved if either the relaxation is tight or the duality gap is zero. The second direction seeks conditions that characterize the global solution using the special problem structure of QC2QP.

Under the first direction, [8] shows strong duality holds if and only if the Hes-

sian of the Lagrangian is positive semidefinite at the global optimal solution. In general, this Hessian can have at most one negative eigenvalue at the global optimal solution [9]. A special case of QC2QP is the extended trust region problem, also known as the Celis-Dennis-Tapia (CDT) subproblem. The CDT subproblem minimizes a nonconvex quadratic objective function over the intersection of two ellipsoids [10]. The necessary and sufficient conditions for strong duality in the CDT subproblem has been provided in [8]. The result shows strong duality holds and the primal optimal solution can be recovered through the semidefinite relaxation if and only if the optimal solutions of the semidefinite relaxation and the dual violate certain conditions, called *Property I*. [11] further studies the CDT subproblems that has a positive duality gap. The result shows that an additional second-order cone (SOC) constraint can decrease the duality gap when certain conditions hold. Specifically, a sufficient condition is provided under which the duality gap can be completely eliminated with two additional SOC constraints for a class of CDT subproblems.

Another method of relaxation is to solve the QC2QP in the complex domain. [12] shows the necessary and sufficient conditions for strong duality, using the classical extended S-Lemma of Fradkov and Yakubovich. By this result and convexity of a quadratic mapping, the authors of [12] show a sufficient condition for strong duality in the real-valued QC2QP. Another sufficient condition for strong duality in the complex-valued problem is provided in [13]. The result is derived using a matrix rank-one decomposition for complex Hermitian matrices.

Under the second direction, necessary conditions for global optimality in QC2QP are proved in [9]. Specifically, the number of negative eigenvalues of the Hessian

of the Lagrangian is characterized at the global optimal solution. For the CDT subproblem, [14] proves necessary and sufficient conditions for global and local optimality using copositivity. Numerical experiments are conducted with randomly generated instances of feasible problems. The result displays a distribution of positive semidefiniteness and copositivity conditions on the Hessian of the Lagrangian at computed minimizers, where positive semidefinite condition holds for a majority of all randomly generated instances.

3.2 Main results

We consider a more general case than the CDT subproblem considered in [8]. We assume Slater's condition hold for (QP0) and its dual. This assumption is weaker than the assumption in [8], which requires at least one of Q_1 and Q_2 to be positive definite. We will add an extra condition to *Property I* and name the combined conditions *Property I⁺*. We prove that strong duality holds if and only if the solutions of the semidefinite relaxation and the dual violate *Property I⁺*.

We adopt the following notations to rewrite (QP0) in a homogeneous quadratic form.

$$M(q_0) = \begin{bmatrix} 0 & b_0^T \\ b_0 & Q_0 \end{bmatrix}, M(q_i) = \begin{bmatrix} c_i & b_i^T \\ b_i & Q_i \end{bmatrix}, i = 1, 2. \quad (3.1)$$

Then (QP0) is equivalently written as

$$\begin{aligned}
& \underset{z \in \mathbb{R}^n, t \in \mathbb{R}}{\text{minimize}} && M(q_0) \cdot \begin{bmatrix} t \\ z \end{bmatrix} \begin{bmatrix} t \\ z \end{bmatrix}^T = z^T Q_0 z + 2tb_0^T z \\
& \text{subject to} && M(q_i) \cdot \begin{bmatrix} t \\ z \end{bmatrix} \begin{bmatrix} t \\ z \end{bmatrix}^T = z^T Q_i z + 2tb_i^T z + t^2 c_i \leq 0, i = 1, 2, \\
& && t^2 = 1.
\end{aligned} \tag{QP}$$

The semidefinite relaxation of (QP) is the following:

$$\begin{aligned}
& \underset{X \in \mathcal{S}^{n+1}}{\text{minimize}} && M(q_0) \cdot X \\
& \text{subject to} && M(q_i) \cdot X \leq 0, i = 1, 2, \\
& && I_{00} \cdot X = 1, \\
& && X \succeq 0,
\end{aligned} \tag{SP}$$

where $I_{00} = \begin{bmatrix} 1 & 0_{1 \times n} \\ 0_{n \times 1} & 0_{n \times n} \end{bmatrix}$.

The dual problem of (SP) is the following:

$$\begin{aligned}
& \underset{y_i \in \mathbb{R}, i=0,1,2}{\text{maximize}} && y_0 \\
& \text{subject to} && Z = M(q_0) - y_0 I_{00} + y_1 M(q_1) + y_2 M(q_2) \succeq 0, \\
& && y_i \geq 0, i = 1, 2.
\end{aligned} \tag{SD}$$

Note that (SD) is also the dual of (QP).

Throughout this chapter, we assume that (QP0), and hence (QP), satisfies the Slater's condition, i.e., $\exists z \in \mathbb{R}^n$ such that

$$z^T Q_i z + 2b_i^T z + c_i < 0, i = 1, 2. \tag{3.2}$$

And assume (SD) satisfies the Slater's condition, i.e., $\exists y_0 \in \mathbb{R}, y_1, y_2 > 0$ such that

$$M(q_0) - y_0 I_{00} + y_1 M(q_1) + y_2 M(q_2) \succ 0, \quad (3.3)$$

which is, by Schur complement, if and only if $\exists y_0 \in \mathbb{R}, y_1, y_2 > 0$ such that

$$Q_0 + y_1 Q_1 + y_2 Q_2 \succ 0, \quad (3.4a)$$

$$-y_0 + y_1 c_1 + y_2 c_2 > b^T(y_1, y_2)(Q_0 + y_1 Q_1 + y_2 Q_2)^{-1} b(y_1, y_2), \quad (3.4b)$$

where $b(y_1, y_2) \stackrel{\text{def}}{=} b_0 + y_1 b_1 + y_2 b_2$.

The existence condition stated in (3.4) is not easy to check, though exceptions exist. By Proposition 2.1 of [15], (SD) satisfies Slater's condition if either the objective function is strictly convex, or at least one of the constraints is elliptical, i.e., if either $Q_0 \succ 0$, or $Q_i \succ 0$ and $b_i^T Q_i^{-1} b_i - c_i > 0$ for at least one of $i = 1, 2$.

Slater's condition holds for (SP) when it holds for (QP). Then, (SP) and (SD) both have attainable optimal solutions. We denote the optimal solutions of (QP), (SP), and (SD), respectively, by x^* , \hat{X} , and $(\hat{Z}, \hat{y}_0, \hat{y}_1, \hat{y}_2)$. Note that a primal-dual feasible pair X and (Z, y_0, y_1, y_2) are optimal if and only if they satisfy the complementary conditions:

$$XZ = 0_{(n+1) \times (n+1)}, y_i M(q_i) \bullet X = 0, i = 1, 2. \quad (3.5)$$

Property I, which we shall state in Definition 3.1, is the key to the necessary and sufficient conditions for a duality gap between (QP) and (SD) when $Q_1 \succ 0$.

Definition 3.1 (Definition 4.1 of [8]). *For \hat{X} and $(\hat{Z}, \hat{y}_0, \hat{y}_1, \hat{y}_2)$, a given pair of optimal solutions for (SP) and (SD), respectively, we say that this pair has Property I if:*

1. $\hat{y}_1\hat{y}_2 \neq 0$;
2. $\text{rank}(\hat{Z}) = n - 1$;
3. $\text{rank}(\hat{X}) = 2$ and there is a rank-one decomposition of \hat{X} , $\hat{X} = \hat{x}_1\hat{x}_1^T + \hat{x}_2\hat{x}_2^T$ such that

$$M(q_1) \cdot \hat{x}_i\hat{x}_i^T = 0, i = 1, 2, \quad (3.6)$$

$$(M(q_2) \cdot \hat{x}_1\hat{x}_1^T)(M(q_2) \cdot \hat{x}_2\hat{x}_2^T) < 0. \quad (3.7)$$

Theorem 3.1 (Theorem 4.2 of [8]). *Consider (QP) where Slater's condition is satisfied and $Q_1 \succ 0$. Suppose that \hat{X} and $(\hat{Z}, \hat{y}_0, \hat{y}_1, \hat{y}_2)$ is a pair of optimal solutions for the semidefinite relaxation (SP) and dual problem (SD), respectively. Then, $\text{val}(\text{SP}) < \text{val}(\text{QP})$ holds if and only if the pair \hat{X} and $(\hat{Z}, \hat{y}_0, \hat{y}_1, \hat{y}_2)$ has Property I.*

We add an extra condition to *Property I* and name the combined conditions *Property I⁺* as the following.

Definition 3.2. *For \hat{X} and $(\hat{Z}, \hat{y}_0, \hat{y}_1, \hat{y}_2)$, a given pair of optimal solutions for (SP) and (SD), respectively, we say that this pair has Property I⁺ if:*

1. *Property I holds;*
2. $M(q_1) \cdot \hat{x}_1\hat{x}_2^T \neq 0$.

As we shall see in the following Theorem, *Property I⁺* is the key to the necessary and sufficient conditions for a duality gap between (QP) and (SD) when the positive definiteness of Q_1 is not assumed.

Theorem 3.2. Consider (QP) where the Slater's condition holds for the primal (QP) and its dual (SD). Suppose that \hat{X} and $(\hat{Z}, \hat{y}_0, \hat{y}_1, \hat{y}_2)$ is a pair of optimal solutions for the semidefinite relaxation (SP) and dual problem (SD), respectively. Then, $\text{val}(\text{SP}) < \text{val}(\text{QP})$ holds if and only if the pair \hat{X} and $(\hat{Z}, \hat{y}_0, \hat{y}_1, \hat{y}_2)$ has Property I⁺.

In order to prove Theorem 3.2, we shall use the following result.

Lemma 3.1 (Theorem 2.4 of [16]). Let $X = VV^T$ be a solution of (SP), where $V \in \mathbb{R}^{n \times r}$ and $r = \text{rank}(X)$. Define the linear mapping $\mathcal{A}_V : \mathcal{S}^r \rightarrow \mathbb{R}^3$ as

$$\mathcal{A}_V(\Delta) \stackrel{\text{def}}{=} \begin{bmatrix} (V^T M(q_1) V) \cdot \Delta \\ (V^T M(q_2) V) \cdot \Delta \\ (V^T I_{00} V) \cdot \Delta \end{bmatrix}. \quad (3.8)$$

Then X is the unique solution of (SP) if and only if

1. X has the maximum rank among all solutions;
2. $\mathcal{N}(\mathcal{A}_V) = \{0_{r \times r}\}$.

Proof of Theorem 3.2. (\Leftarrow). We first show \hat{X} is the unique solution to (SP) using Lemma 3.1. Then a positive duality gap is a trivial consequence.

Let \tilde{X} denote an optimal solution of (SP). Then, by Sylvester's Inequality and the complementary condition (3.5), we have

$$\text{rank}(\tilde{X}) + \text{rank}(\hat{Z}) - (n + 1) \leq \text{rank}(\tilde{X}\hat{Z}) = 0 \Rightarrow \text{rank}(\tilde{X}) \leq 2, \quad (3.9)$$

i.e., the maximum rank of the optimal solution of (SP) is 2. Since $\text{rank}(\hat{X}) = 2$ already, in order to show that \hat{X} is the unique solution of (SP), by Lemma 3.1, we

only need to show $\mathcal{N}(\mathcal{A}_V) = \{0_{2 \times 2}\}$, where $V \in \mathbb{R}^{(n+1) \times 2}$ is defined as $\hat{X} = VV^T$.

This is equivalent to show that the solution $\Delta \in \mathcal{S}^2$ of the following equation,

$$\begin{bmatrix} (V^T M(q_1) V) \cdot \Delta \\ (V^T M(q_2) V) \cdot \Delta \\ (V^T I_{00} V) \cdot \Delta \end{bmatrix} = 0_{3 \times 1}, \quad (3.10)$$

is $\Delta = 0_{2 \times 2}$ only. Consider the following partitions

$$V = \begin{bmatrix} \hat{x}_1 & \hat{x}_2 \end{bmatrix}, \hat{x}_1 = \begin{bmatrix} t_1 \\ z_1 \end{bmatrix}, \hat{x}_2 = \begin{bmatrix} t_2 \\ z_2 \end{bmatrix}, \Delta = \begin{bmatrix} \Delta_1 & \Delta_2 \\ \Delta_2 & \Delta_3 \end{bmatrix}, \quad (3.11)$$

where $z_1, z_2 \in \mathbb{R}^n$ and $t_1, t_2, \Delta_1, \Delta_2, \Delta_3 \in \mathbb{R}$. Since $\hat{y}_1 \hat{y}_2 \neq 0$, then the inequality constraints in (SP) are all active at \hat{X} . Especially,

$$M(q_2) \cdot \hat{X} = M(q_2) \cdot \hat{x}_1 \hat{x}_1^T + M(q_2) \cdot \hat{x}_2 \hat{x}_2^T = 0. \quad (3.12)$$

W.L.O.G., let $\alpha > 0$ be such that

$$\alpha = M(q_2) \cdot \hat{x}_1 \hat{x}_1^T = -M(q_2) \cdot \hat{x}_2 \hat{x}_2^T. \quad (3.13)$$

Then (3.10) is a linear equation of Δ_1, Δ_2 and Δ_3 .

$$\begin{bmatrix} V^T M(q_1) V \cdot \Delta \\ V^T M(q_2) V \cdot \Delta \\ V^T I_{00} V \cdot \Delta \end{bmatrix} = 0 \Rightarrow \overbrace{\begin{bmatrix} 0 & 2M(q_1) \cdot \hat{x}_1 \hat{x}_2^T & 0 \\ \alpha & 2M(q_2) \cdot \hat{x}_1 \hat{x}_2^T & -\alpha \\ t_1^2 & 2t_1 t_2 & t_2^2 \end{bmatrix}}^{\stackrel{\text{def}}{=} \Gamma} \begin{bmatrix} \Delta_1 \\ \Delta_2 \\ \Delta_3 \end{bmatrix} = 0_{3 \times 1}. \quad (3.14)$$

Notice that $I_{00} \cdot \hat{X} = 0$ implies $t_1^2 + t_2^2 = 1$, i.e., t_1 and t_2 can not be 0 simultaneously.

W.L.O.G., assume $t_1 \neq 0$. Then the row vectors in Γ are linearly independent

because $M(q_1) \cdot \hat{x}_1 \hat{x}_2^T \neq 0$, and hence, Γ has full rank. So the only solution to (3.10)

is $\Delta = 0_{2 \times 2}$. This completes our proof that \hat{X} is the unique solution to (SP).

Let z^* denote the optimal solution of (QP0). Then, W.L.O.G., $\begin{bmatrix} 1 \\ z^* \end{bmatrix}$ is an optimal solution to (QP) and $\begin{bmatrix} 1 \\ z^* \end{bmatrix} \begin{bmatrix} 1 \\ z^* \end{bmatrix}^T$ is feasible to (SP). And the corresponding optimal value of (QP) is

$$M(q_0) \cdot \begin{bmatrix} 1 \\ z^* \end{bmatrix} \begin{bmatrix} 1 \\ z^* \end{bmatrix}^T = (z^*)^T Q_0 z^* + 2b_0^T z^* = \text{val}(\text{QP}). \quad (3.15)$$

Since \hat{X} is the unique solution to (SP), we conclude that $\text{val}(\text{QP}) > \text{val}(\text{SP})$.

(\Rightarrow) We prove by contraposition. We will enumerate five exhaustive (but not mutually exclusive) possibilities, denoted by *Case i*, with $i = 1, 2, 3, 4, 5$.

Case 1. $\hat{y}_1 \hat{y}_2 = 0$.

The proof that the semidefinite relaxation is tight in this case can be found in [15].

Case 2. $\hat{y}_1 \hat{y}_2 \neq 0$ and $\text{rank}(\hat{X}) \neq 2$.

The condition $\hat{y}_1 \hat{y}_2 \neq 0$ implies that, by the complementary condition (3.5),

$$M(q_1) \cdot \hat{X} = M(q_2) \cdot \hat{X} = 0. \quad (3.16)$$

Let $r \stackrel{\text{def}}{=} \text{rank}(\hat{X})$. Obviously, $r > 0$ because $I_{00} \cdot \hat{X} = 1$. If $r = 1$, then the rank-one decomposition of $\hat{X} = \hat{x} \hat{x}^T$ provides the optimal solution to (QP). And we do not need to consider the case $r \geq 3$ because (SD) satisfies Slater's condition and hence (SP) is solvable, and hence by Theorem 2.1 of [16], $r \leq 2$.

Case 3. $\hat{y}_1 \hat{y}_2 \neq 0$, $\text{rank}(\hat{X}) = 2$, and $M(q_1) \cdot \hat{x}_i \hat{x}_i^T = M(q_2) \cdot \hat{x}_i \hat{x}_i^T = 0$, $i = 1, 2$.

Clearly, $\hat{x}_1 \hat{x}_1^T$ and $\hat{x}_2 \hat{x}_2^T$ are both optimal solutions to (SP). We adopt the

partition of \hat{x}_1 and \hat{x}_2 in (3.11). Then $I_{00} \cdot \hat{X} = 1$ implies that $t_1^2 + t_2^2 = 1$, i.e., at least one of t_1 and t_2 is nonzero. W.L.O.G., assume $t_1 \neq 0$. Then \hat{x}_1/t_1 is a homogenized solution to (QP) and z_1/t_1 is a solution to (QP0).

Case 4. $\hat{y}_1\hat{y}_2 \neq 0$, $\text{rank}(\hat{X}) = 2$, $(M(q_2) \cdot \hat{x}_1\hat{x}_1^T)(M(q_2) \cdot \hat{x}_2\hat{x}_2^T) < 0$, and $\text{rank}(\hat{Z}) \neq n - 1$.

Since $\text{rank}(\hat{Z}) + \text{rank}(\hat{X}) \leq n + 1$, $\text{rank}(\hat{X}) = 2$, and $\text{rank}(\hat{Z}) \neq n - 1$, it follows that $\text{rank}(\hat{Z}) < n - 1$. Now $\hat{X} + \hat{Z}$ is singular and both \hat{X} and \hat{Z} are positive semidefinite. So there must be a nontrivial $y \in \mathbb{R}^{n+1}$ in the intersection of the null space of \hat{X} and \hat{Z} . Let

$$X \stackrel{\text{def}}{=} \hat{X} + yy^T = \hat{x}_1\hat{x}_1^T + \hat{x}_2\hat{x}_2^T + yy^T. \quad (3.17)$$

Obviously, $\text{rank}(X) = 3$ and $\hat{Z}X = 0_{(n+1) \times (n+1)}$ because $\hat{Z}\hat{X} = 0_{(n+1) \times (n+1)}$. By Lemma 3.3 of [8], we know there exist an $x \in \mathbb{R}^{n+1}$ such that X is rank-one decomposable¹ at x and that

$$M(q_1) \cdot xx^T = M(q_2) \cdot xx^T = 0. \quad (3.18)$$

Since x is in the range space of X , it must be in the null space of \hat{Z} . Then $\hat{Z} \cdot xx^T = 0$ and the complementary condition (3.5) implies that xx^T/t^2 is an optimal solution to (SP), where t is the first element of x . Hence x/t is an optimal solution to (QP). Note that $t \neq 0$ for the following reason. Consider the following partition

¹As a convention, a matrix X is rank-one decomposable at x_1 if there exists other $r - 1$ vectors x_2, \dots, x_r such that $X = x_1x_1^T + x_2x_2^T + \dots + x_rx_r^T$, where $r \stackrel{\text{def}}{=} \text{rank}(X)$ [8].

of x ,

$$x = \begin{bmatrix} t \\ z \end{bmatrix}, \quad (3.19)$$

where $z \in \mathbb{R}^n$. By contradiction, assume $t = 0$. Then $\hat{Z} \cdot xx^T = 0$ implies

$$z^T(Q_0 + \hat{y}_1 Q_1 + \hat{y}_2 Q_2)z = 0. \quad (3.20)$$

On the other hand, by (3.18), we have

$$z^T Q_1 z = z^T Q_2 z = 0. \quad (3.21)$$

Combining (3.20) and (3.21), we know $z^T Q_0 z = 0$. The only z that satisfies $z^T Q_i z = 0$, for $i = 0, 1, 2$, is $z = 0_{n \times 1}$ for the following reason. By contradiction, assume $z \neq 0_{n \times 1}$. Then, for arbitrary $\bar{y}_1, \bar{y}_2 > 0$, we have

$$z^T(Q_0 + \bar{y}_1 Q_1 + \bar{y}_2 Q_2)z = 0. \quad (3.22)$$

This equality contradicts Slater's condition (3.4a) of (SD) and hence $z = 0_{n \times 1}$.

Then, $x = 0_{(n+1) \times 1}$. This is a contradiction because $x \neq 0_{(n+1) \times 1}$. So we have proved $t \neq 0$.

Case 5. $\hat{y}_1 \hat{y}_2 \neq 0$, $\text{rank}(\hat{Z}) = n - 1$, $\text{rank}(\hat{X}) = 2$, $M(q_1) \cdot \hat{x}_i \hat{x}_i^T = 0, i = 1, 2$, $(M(q_2) \cdot \hat{x}_1 \hat{x}_1^T)(M(q_2) \cdot \hat{x}_2 \hat{x}_2^T) < 0$, and $M(q_1) \cdot \hat{x}_1 \hat{x}_2^T = 0$.

We show that we can obtain another rank-one decomposition of $\hat{X} = \check{x}_1 \check{x}_1^T + \check{x}_2 \check{x}_2^T$ such that $M(q_1) \cdot \check{x}_i \check{x}_i^T = M(q_2) \cdot \check{x}_i \check{x}_i^T = 0, i = 1, 2$. This is achievable because any rank-one decomposition of \hat{X} must be a linear combination of \hat{x}_1 and \hat{x}_2 . And $M(q_1) \cdot \hat{x}_1 \hat{x}_2^T = 0$ together with $M(q_1) \cdot \hat{x}_1 \hat{x}_1^T = M(q_1) \cdot \hat{x}_2 \hat{x}_2^T = 0$ indicate that an arbitrary linear combination \tilde{x} of \hat{x}_1 and \hat{x}_2 , i.e.,

$$\tilde{x} = \alpha_1 \hat{x}_1 + \alpha_2 \hat{x}_2, \quad (3.23)$$

for $\alpha_1, \alpha_2 \in \mathbb{R}$, satisfies $M(q_1) \cdot \tilde{x}\tilde{x}^T = 0$. Hence, we only need to obtain \tilde{x}_1 and \tilde{x}_2 as linear combinations of \hat{x}_1 and \hat{x}_2 , where we only require $M(q_2) \cdot \tilde{x}_i\tilde{x}_i^T = 0$ for $i = 1, 2$. We adopt the following method from the proof of Lemma 2.2 of [15] to obtain \tilde{x}_1 and \tilde{x}_2 .

Consider a quadratic equation of $\beta \in \mathbb{R}$,

$$0 = (\beta\hat{x}_1 + \hat{x}_2)^T M(q_2)(\beta\hat{x}_1 + \hat{x}_2) = \beta^2 \hat{x}_1^T M(q_2)\hat{x}_1 + 2\beta \hat{x}_1^T M(q_2)\hat{x}_2 + \hat{x}_2^T M(q_2)\hat{x}_2. \quad (3.24)$$

This equation must have two distinctive real roots with opposite signs since

$$(\hat{x}_1^T M(q_2)\hat{x}_1)(\hat{x}_2^T M(q_2)\hat{x}_2) < 0. \quad (3.25)$$

Let $\bar{\beta}$ be one of the roots and

$$\tilde{x}_1 = \frac{\bar{\beta}}{\sqrt{\bar{\beta}^2 + 1}}\hat{x}_1 + \frac{1}{\sqrt{\bar{\beta}^2 + 1}}\hat{x}_2, \quad (3.26)$$

$$\tilde{x}_2 = -\frac{1}{\sqrt{\bar{\beta}^2 + 1}}\hat{x}_1 + \frac{\bar{\beta}}{\sqrt{\bar{\beta}^2 + 1}}\hat{x}_2. \quad (3.27)$$

Then we have a new rank-one decomposition $\hat{X} = \tilde{x}_1\tilde{x}_1^T + \tilde{x}_2\tilde{x}_2^T$ and $M(q_1) \cdot \tilde{x}_i\tilde{x}_i^T = M(q_2) \cdot \tilde{x}_i\tilde{x}_i^T = 0$ for $i = 1, 2$. The rest of this case continues in *Case 3*. \square

Remark 3.1. *Theorem 2.6 of [11] also proves the uniqueness of the solution of (SP) in the CDT subproblem when Property I holds. The method uses a property of boundary points of a SOC while we use a result on the uniqueness of a solution of a semidefinite programming.*

3.3 Relation between Theorem 3.1 and Theorem 3.2

Theorem 3.2 is an extension to Theorem 3.1 in the following aspects. First, Theorem 3.2 characterizes the necessary and sufficient conditions for a duality gap under a weaker assumption than Theorem 3.1. Second, the necessary and sufficient conditions in Theorem 3.2 involve an extra condition in *Property I*⁺ compared to *Property I*, which is required by Theorem 3.1.

On the other hand, when $Q_1 \succ 0$, Theorem 3.2 coincides with Theorem 3.1 because the extra condition in *Property I*⁺, $M(q_1) \cdot \hat{x}_1 \hat{x}_2^T \neq 0$, is redundant in *Property I*. We show the redundancy in the following proposition.

Proposition 3.1. *Consider (QP) where Slater's condition holds and $Q_1 \succ 0$. Let \hat{X} and $(\hat{Z}, \hat{y}_0, \hat{y}_1, \hat{y}_2)$ denote a pair of optimal solutions for (SP) and (SD), respectively. Suppose $\text{rank}(\hat{X}) = 2$, $\hat{y}_1 \hat{y}_2 \neq 0$, and there exists a rank-one decomposition $\hat{X} = \hat{x}_1 \hat{x}_1^T + \hat{x}_2 \hat{x}_2^T$ such that $M(q_1) \cdot \hat{x}_i \hat{x}_i^T = 0$ for $i = 1, 2$. Then $M(q_1) \cdot \hat{x}_1 \hat{x}_2^T \neq 0$.*

Proof. We apply a change of coordinates to make

$$M(q_1) = \begin{bmatrix} -1 & 0_{1 \times n} \\ 0_{n \times 1} & I_n \end{bmatrix}. \quad (3.28)$$

The linear transformation in the change of coordinates is displayed at the end of this proof.

By contradiction, assume $M(q_1) \cdot \hat{x}_1 \hat{x}_2^T = 0$. Then, adopting the partition of

\hat{x}_1 and \hat{x}_2 in (3.11), we have

$$\begin{cases} M(q_1) \cdot \hat{x}_1 \hat{x}_1^T = 0 \\ M(q_1) \cdot \hat{x}_2 \hat{x}_2^T = 0 \\ M(q_1) \cdot \hat{x}_1 \hat{x}_2^T = 0 \end{cases} \Rightarrow \begin{cases} z_1^T z_1 = t_1^2 \\ z_2^T z_2 = t_2^2 \\ z_1^T z_2 = t_1 t_2 \end{cases} \Rightarrow (z_1^T z_1)(z_2^T z_2) = (z_1^T z_2)^2. \quad (3.29)$$

By Cauchy-Schwarz inequality, the last equality implies z_1 and z_2 are linearly dependent, and so are \hat{x}_1 and \hat{x}_2 , which contradicts the fact that $\text{rank}(\hat{X}) = 2$ and $\hat{X} = \hat{x}_1 \hat{x}_1^T + \hat{x}_2 \hat{x}_2^T$.

Adopting the partition of x in (3.19), we apply the following linear transformation in the change of coordinates,

$$\begin{bmatrix} \tilde{t} \\ \tilde{z} \end{bmatrix} = \begin{bmatrix} \sqrt{b_1^T Q_1^{-1} b_1 - c_1^2} & 0_{n \times 1} \\ Q_1^{-1/2} b_1 & Q_1^{1/2} \end{bmatrix} \begin{bmatrix} t \\ z \end{bmatrix}, \quad (3.30)$$

where $\tilde{t}, t \in \mathbb{R}$ and $\tilde{z}, z \in \mathbb{R}^n$. The vector $\begin{bmatrix} \tilde{t} \\ \tilde{z}^T \end{bmatrix}^T$ is a partition of the new variable \tilde{x} after the change of coordinates. Then,

$$\begin{aligned} \tilde{q}_1(\tilde{x}) &= \tilde{x}^T M(q_1) \tilde{x} \\ &= \begin{bmatrix} \tilde{t} \\ \tilde{z} \end{bmatrix}^T \begin{bmatrix} -1 & 0_{1 \times n} \\ 0_{n \times 1} & I_n \end{bmatrix} \begin{bmatrix} \tilde{t} \\ \tilde{z} \end{bmatrix} \\ &= \begin{bmatrix} t \\ z \end{bmatrix}^T \begin{bmatrix} \sqrt{b_1^T Q_1^{-1} b_1 - c_1^2} & b_1^T Q_1^{-1/2} \\ 0_{1 \times n} & Q_1^{1/2} \end{bmatrix} \begin{bmatrix} -1 & 0_{1 \times n} \\ 0_{n \times 1} & I_n \end{bmatrix} \begin{bmatrix} \sqrt{b_1^T Q_1^{-1} b_1 - c_1^2} & 0_{n \times 1} \\ Q_1^{-1/2} b_1 & Q_1^{1/2} \end{bmatrix} \begin{bmatrix} t \\ z \end{bmatrix} \\ &= z^T Q_1 z + 2t b_1^T Q_1 z + t^2 c_1 = q_1(x). \end{aligned} \quad (3.31)$$

□

Chapter 4: Two-stage Optimal Control for Target Reaching Inside a Time-costly Area

In this chapter, we consider a problem whose goal is to steer a mobile agent to reach a target. Especially, the target is enclosed within a special area that is time-costly, while the mobile agent starts outside such area, as displayed in Figure 4.1. We define the time-costly area such that the mobile agent begins to incur a cost when it enters such area for the first time. And this cost increases as the elapsed time, between its first entry to such area and its arrival on the target, goes on.

The goal of this problem is to find a control and a terminal time such that the mobile agent can reach the target while incurring a minimum cost which will be defined later. The presence of the time-costly area enables decomposition of this problem into two stages. We refer to the stages before and after the mobile agent's first entry to the time-costly area as the outer stage and the inner stage, respectively. Hence, we can formulate the problem as a two-stage optimal control problem, where each stage becomes a subproblem. The key idea in solving the problem is to augment the minimum cost of the inner stage subproblem as a terminal performance index of the outer stage subproblem.

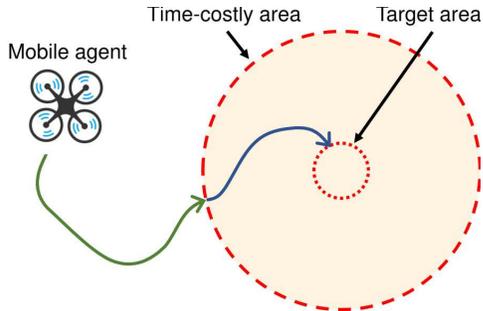


Figure 4.1: Illustration of the problem

4.1 Related literature

The authors of [17] first considers the vehicle maneuvering problem in which the terminal set was defined as the target region for the maneuver. Such set is not necessarily invariant, which is a typical setting named invariant set in the Model Predictive Control literature [18]. The two-stage optimal control problem is studied in [19]. By Pontryagin’s maximum principle, the author proves necessary conditions for optimality in a general two-stage problem with an adjustable intermediate time.

4.2 Problem formulation

To model the motion of the mobile agent in a two-dimensional¹ (2D) plane, we consider the following linear time-invariant dynamics

$$\dot{x}(t) = Ax(t) + Bu(t), x(0) = x_0, \quad (4.1)$$

¹The results in this thesis can be easily extended to the motion of a mobile agent in a three-dimensional (3D) space, by considering the state variable with a 3D position portion and a 3D velocity portion and corresponding changes in problem setting. We adopt the 2D motion here to make the figures and illustrations straightforward.

where $u(t) \in \mathbb{R}^2$ is the control input and $x(t) \in \mathbb{R}^4$ is the state variable.

Definition 4.1. *We define the subspace of the first two dimensions of the state space as the position subspace, and the last two dimensions as the velocity subspace.*

Then the projections of state variable $x(t)$ onto the position subspace and velocity subspace are denoted by $x_p(t) \in \mathbb{R}^2$ and $x_v(t) \in \mathbb{R}^2$, respectively, i.e.,

$$x(t) = \begin{bmatrix} x_p(t) \\ x_v(t) \end{bmatrix}. \quad (4.2)$$

Matrices $A \in \mathbb{R}^{4 \times 4}$ and $B \in \mathbb{R}^{4 \times 2}$ are known. The initial state x_0 is given. We define the set of admissible controls by

$$\mathcal{U}(t_1, t_2) \stackrel{\text{def}}{=} \{u : [t_1, t_2] \rightarrow \mathbb{R}^2, \text{ piecewisely continuous}\}. \quad (4.3)$$

Assumption 4.1. *(A, B) is a reachable pair.*

The time-costly set and the target set are denoted by \mathcal{D}_1 and \mathcal{D}_2 , where

$$\mathcal{D}_i \stackrel{\text{def}}{=} \{z \in \mathbb{R}^4 : \|z\|_D^2 \leq d_i^2\}, i = 1, 2, \quad (4.4)$$

$$D = \begin{bmatrix} D_p & 0_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} \end{bmatrix}, \quad (4.5)$$

with $D_p \in \mathcal{S}^2$, $D_p \succ 0$ and $d_1 > d_2$. Values of D_p , d_1 and d_2 are known.

Note that the projections of \mathcal{D}_1 and \mathcal{D}_2 onto the position subspace are the elliptical sets

$$\mathcal{D}_1^p \stackrel{\text{def}}{=} \{z_p \in \mathbb{R}^2 : \|z_p\|_{D_p}^2 \leq d_1^2\}, \quad (4.6)$$

$$\mathcal{D}_2^p \stackrel{\text{def}}{=} \{z_p \in \mathbb{R}^2 : \|z_p\|_{D_p}^2 \leq d_2^2\}, \quad (4.7)$$

which correspond to the time-costly area and the target area, respectively, in the illustration shown in Figure 4.1.

Definition 4.2. *We use the name switching time and switching state for the time and state of the mobile agent, respectively, when it enters the time-costly area for the first time.*

We use $u_o(t)$ and $u_i(t)$ to denote the control in the outer stage and in the inner stage, respectively, i.e.,

$$u(t) = \begin{cases} u_o(t), & \text{if } t \in [0, t_0), \\ u_i(t), & \text{if } t \in [t_0, t_f), \end{cases} \quad (4.8)$$

where t_0 denotes the switching time and t_f denotes the terminal time. Let $u_o \in \mathcal{U}(0, t_0)$ and $u_i \in \mathcal{U}(t_0, t_f)$.

The outer stage control $u_o(t)$ is supposed to control the mobile agent to reach the boundary of the time-costly area at t_0 , i.e.,

$$\|x(t_0)\|_D^2 = d_1^2. \quad (4.9)$$

The inner stage control $u_i(t)$ is supposed to control the mobile agent to reach the target area at t_f , i.e.,

$$\|x(t_f)\|_D^2 \leq d_2^2. \quad (4.10)$$

The cost function for the outer stage is quadratic as the following,

$$\frac{1}{2} \int_0^{t_0} \|u_o(t)\|_R^2 + \|x(t)\|_Q^2 dt, \quad (4.11)$$

where $R \succ 0, Q \succeq 0$ are of conformed dimensions and given. The cost function for

the inner stage is the sum of control effort and an elapsed time cost, i.e.,

$$\frac{1}{2} \int_{t_0}^{t_f} \|u_i(t)\|_R^2 dt + \phi(t_f - t_0), \quad (4.12)$$

where $\phi : \mathbb{R}_+ \rightarrow \mathbb{R}$ is a continuously differentiable and strictly increasing function which is known. We use $\phi(t_f - t_0)$ to stress the explicit time cost in the time-costly area. The elapsed time of the outer stage is the switching time t_0 which is fixed. And the elapsed time of the inner stage $t_f - t_0$ is allowed to vary within the range $(0, T]$, with T given.

Now the problem studied in this chapter is formulated as the following.

Problem: Find a control u and a terminal time t_f such that the system (4.1) is steered to the target with a minimum sum of the outer stage cost and the inner stage cost, i.e.,

$$\begin{aligned} & \underset{u \in \mathcal{U}(0, t_f), t_f \in \mathbb{R}}{\text{minimize}} && \frac{1}{2} \int_0^{t_0} \|u_o(t)\|_R^2 + \|x(t)\|_Q^2 dt + \frac{1}{2} \int_{t_0}^{t_f} \|u_i(t)\|_R^2 dt + \phi(t_f - t_0) \\ & \text{subject to} && \dot{x}(t) = Ax(t) + Bu(t), x(0) = x_0, \\ & && u(t) = \begin{cases} u_o(t), & \text{if } t \in [0, t_0), \\ u_i(t), & \text{if } t \in [t_0, t_f), \end{cases} \\ & && \|x(t_0)\|_D^2 = d_1^2, \\ & && \|x(t_f)\|_D^2 \leq d_2^2, \\ & && t_f \in (t_0, t_0 + T]. \end{aligned} \quad (\text{P})$$

To tackle this problem, we propose the following subproblems.

Problem 1 (Inner stage problem): Find an inner stage control u_i which steers the system (4.1) to the target area from t_0 to t_f , with a minimum control effort, i.e.,

$$\begin{aligned} & \underset{u_i \in \mathcal{U}(t_0, t_f)}{\text{minimize}} && \frac{1}{2} \int_{t_0}^{t_f} \|u_i(t)\|_R^2 dt \\ & \text{subject to} && \dot{x}(t) = Ax(t) + Bu_i(t), x(t_0) \text{ given,} \end{aligned} \tag{P1}$$

$$\|x(t_f)\|_D^2 \leq d_2^2.$$

Let $J_i^* : \mathbb{R}_+ \times \mathbb{R}^4 \rightarrow \mathbb{R}_+$ be such that $J_i^*(t_f, x(t_0)) = \text{val}(\text{P1})$, which is the minimum control effort of the inner stage with terminal time t_f and a given initial state $x(t_0)$.

Problem 2 (Augmented outer stage problem): Find an outer stage control u_o which steers the system (4.1) to the boundary of the time-costly area from 0 to t_0 . Such control must also minimize the outer stage cost plus a terminal performance index, which is augmented from the minimum cost of the inner stage, i.e.,

$$\begin{aligned} & \underset{u_o \in \mathcal{U}(0, t_0)}{\text{minimize}} && \frac{1}{2} \int_0^{t_0} \|u_o(t)\|_R^2 + \|x(t)\|_Q^2 dt + J_i^*(t_f, x(t_0)) \\ & \text{subject to} && \dot{x}(t) = Ax(t) + Bu_o(t), x(0) = x_0, \end{aligned} \tag{P2}$$

$$\|x(t_0)\|_D^2 = d_1^2.$$

Then let $J^* : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ be such that $J^*(t_f) = \text{val}(\text{P2})$.

Problem 3: Find a terminal time t_f such that the following sum is minimized

$$\underset{t_f \in (t_0, t_0 + T]}{\text{minimize}} \quad J^*(t_f) + \phi(t_f - t_0). \tag{P3}$$

Note that (P3) is equivalent to (P). The key to solving problem (P) is to find a locally optimal terminal time t_f^* of (P3). We plug t_f^* in (P2) to find the optimal control of the outer stage. Then the switching state under the outer stage optimal control is plugged in (P1), together with t_f^* , to solve for the optimal control of the inner stage. We will suggest a method to determine t_f^* in the next section.

4.3 Main results

In this section, we solve problem (P) by first providing the solutions to (P1) and (P2). The solutions and properties of (P1) and (P2) are obtained through the equivalent problem of each. We propose an algorithm that can solve (P2) efficiently using strong duality. And we suggest the bisection method to search for a solution of (P3), using first-order necessary conditions. Once (P3) is solved, we show how to obtain the solution to the original problem (P).

4.3.1 The inner stage problem

By the Linear-quadratic regulator (LQR) theory [20], problem (P1) is equivalent to the following problem.

$$\begin{aligned} & \underset{r(t_f) \in \mathbb{R}^4}{\text{minimize}} && \frac{1}{2} \|x_f - r(t_f)\|_{\Delta^{-1}(t_f)}^2 \\ & \text{subject to} && \|r(t_f)\|_D^2 \leq d_2^2, \end{aligned} \tag{EP1}$$

where $x_f = \Phi(t_f, t_0)x(t_0)$ is the propagated state at t_f , with $\Phi(t, \tau) = e^{A(t-\tau)}$ being the state-transition matrix. And $\Delta(t_f) = \int_{t_0}^{t_f} e^{A(t_f-\tau)} B R^{-1} B^T e^{A^T(t_f-\tau)} d\tau$ is the weighted reachability gramian. The vector $r(t_f)$ is the desired terminal state, which is an optimization variable here.

Problem (EP1) seeks a desired terminal state $r(t_f)$ within the target area such that the distance between $r(t_f)$ and the propagated state is minimized, where the distance is measured by the inverse of the weighted reachability gramian.

Note that $\Delta(t_f)$ is positive definite because (A, B) , and hence $(A, BR^{-\frac{1}{2}})$, is a

reachable pair. Since $\Delta(t_f) \succ 0$ and $D \succeq 0$, (EP1) is a convex quadratic constrained quadratic programming (QCQP) which has a unique minimizer, denoted by $r^*(t_f)$. As the expression indicates, $r^*(t_f)$ is a vector-valued function of t_f . By [20], the optimal control $u_i^*(t)$ of (P1) is given by

$$u_i^*(t) = -R^{-1}B^T e^{A^T(t_f-t)} \Delta^{-1}(t_f)(x_f - r^*(t_f)), \quad (4.13)$$

for $t \in [t_0, t_f]$. And $\text{val}(\text{EP1}) = J_i^*(t_f, x(t_0))$.

The following Lemma states properties of the minimum control effort problem of the inner stage.

Lemma 4.1. *For $t_f \in (t_0, t_0 + T)$,*

1. *The function $J_i^*(t_f, x(t_0))$ is continuous w.r.t. t_f .*
2. *The optimal terminal state $r^*(t_f)$ is a vector-valued function that is continuous w.r.t. t_f .*
3. *The partial derivative of $J_i^*(t_f, x(t_0))$ w.r.t. t_f , denoted as $\dot{J}_i^*(t_f)$, is*

$$\dot{J}_i^*(t_f) = \begin{cases} \|r^*(t_f)\|_{\Theta_1}^2, & \text{if } x_{fp} \notin \mathcal{D}_2^p, \\ 0, & \text{if } x_{fp} \in \text{int}(\mathcal{D}_2^p), \end{cases} \quad (4.14)$$

where $\Theta_1 = -2(\lambda_1^*)^2 DBR^{-1}B^T D + \lambda_1^*(A^T D + DA)$ and $\lambda_1^* \in \mathbb{R}_+$ is the Lagrange multiplier satisfying $\Delta^{-1}(t_f)(r^*(t_f) - x_f) = -2\lambda_1^* D r^*(t_f)$. We use $x_{fp} \in \mathbb{R}^2$ to denote the projection of x_f onto the position subspace.

Proof. Assume $t_1, t_2 \in (t_0, t_0 + T)$. Let r_1^* and r_2^* denote the minimizers of (EP1) with a given initial state $x(t_0)$ and terminal times t_1 and t_2 , respectively. Let

$x_1 = \Phi(t_1, t_0)x(t_0)$ and $x_2 = \Phi(t_2, t_0)x(t_0)$. To simplify the notation, we use $W(t)$ to denote $\Phi^T(t, t_0)\Delta^{-1}(t)$.

1. We first show that $J_i^*(t_f, x(t_0))$ is continuous w.r.t. t_f :

$$\begin{aligned} & \lim_{t_2 \rightarrow t_1} J_i^*(t_1, x(t_0)) - J_i^*(t_2, x(t_0)) \\ &= \frac{1}{2} \lim_{t_2 \rightarrow t_1} \|r_1^* - x_1\|_{\Delta^{-1}(t_1)}^2 - \|r_2^* - x_2\|_{\Delta^{-1}(t_2)}^2 \end{aligned} \quad (4.15)$$

$$\geq \frac{1}{2} \lim_{t_2 \rightarrow t_1} \|r_1^* - x_1\|_{\Delta^{-1}(t_1)}^2 - \|r_1^* - x_2\|_{\Delta^{-1}(t_2)}^2 \quad (4.16)$$

$$\begin{aligned} &= \frac{1}{2} \lim_{t_2 \rightarrow t_1} \left\{ \|x(t_0)\|_{W(t_1)\Phi(t_1, t_0) - W(t_2)\Phi(t_2, t_0)}^2 - 2x^T(t_0)(W(t_1) - W(t_2))r_1^* \right. \\ & \quad \left. + \|r_1^*\|_{\Delta^{-1}(t_1)}^2 - \|r_1^*\|_{\Delta^{-1}(t_2)}^2 \right\} = 0, \end{aligned} \quad (4.17)$$

where we use the fact that $\Delta^{-1}(t_f)$ is entrywisely continuous w.r.t. to t_f .

Similarly, replacing r_1^* by r_2^* in (4.16), we get

$$\lim_{t_2 \rightarrow t_1} J_i^*(t_1, x(t_0)) - J_i^*(t_2, x(t_0)) \leq 0. \quad (4.18)$$

Hence,

$$\lim_{t_2 \rightarrow t_1} J_i^*(t_1, x(t_0)) - J_i^*(t_2, x(t_0)) = 0. \quad (4.19)$$

Because t_1 is arbitrarily chosen, we conclude $J_i^*(t_f, x(t_0))$ is continuous w.r.t. $t_f \in (t_0, t_0 + T)$.

2. We prove the continuity of minimizer $r^*(t_f)$ w.r.t. t_f by contradiction.

Assume $\lim_{t_2 \rightarrow t_1} r_2^* = \bar{r} \neq r_1^*$, where the limit is an elementwise limit. Then, by the continuity of $J_i^*(t_f, x(t_0))$, we have

$$\frac{1}{2} \|r_1^* - x_1\|_{\Delta^{-1}(t_1)}^2 = J_i^*(t_1, x(t_0)) = \frac{1}{2} \lim_{t_2 \rightarrow t_1} \|r_2^* - x_2\|_{\Delta^{-1}(t_2)}^2 = \frac{1}{2} \|\bar{r} - x_1\|_{\Delta^{-1}(t_1)}^2, \quad (4.20)$$

which contradicts the uniqueness of the minimizer of (EP1) because $\Delta^{-1}(t_1) \succ 0$.

As t_1 is arbitrarily chosen, We conclude the continuity of $r(t_f)$.

3. Notice $J_i^*(t_f, x(t_0)) = 0$ in the trivial case $x_f \in \mathcal{D}_2$, because the minimizer $r^*(t_f) = x_f$. Hence the derivative

$$\dot{J}_i^*(t_f, x(t_0)) = 0, \text{ if } x_{fp} \in \text{int}(\mathcal{D}_2^p). \quad (4.21)$$

We now assume $x_{fp} \notin \mathcal{D}_2^p$ and deduce a formula of the derivative of $J_i^*(t_f, x(t_0))$ w.r.t. t_f .

$$\begin{aligned} & \lim_{t_2 \rightarrow t_1^-} \frac{J_i^*(t_1, x(t_0)) - J_i^*(t_2, x(t_0))}{t_1 - t_2} \\ & \geq \frac{1}{2} \lim_{t_2 \rightarrow t_1^-} \frac{\|r_1^* - x_1\|_{\Delta^{-1}(t_1)}^2 - \|r_1^* - x_2\|_{\Delta^{-1}(t_2)}^2}{t_1 - t_2} \end{aligned} \quad (4.22)$$

$$\begin{aligned} & = \frac{1}{2} \lim_{t_2 \rightarrow t_1^-} \left\{ \frac{\|x(t_0)\|_{W(t_1)\Phi(t_1, t_0) - W(t_2)\Phi(t_2, t_0)}^2}{t_1 - t_2} - \frac{2x^T(t_0)(W(t_1) - W(t_2))r_1^*}{t_1 - t_2} \right. \\ & \quad \left. + \frac{\|r_1^*\|_{\Delta^{-1}(t_1) - \Delta^{-1}(t_2)}^2}{t_1 - t_2} \right\} \end{aligned} \quad (4.23)$$

$$= \frac{1}{2} \|x(t_0)\|_{\left(\frac{d}{dt}W(t)\Phi(t, t_0)\right)|_{t=t_1}}^2 - x^T(t_0) \left(\frac{d}{dt}W(t)\right)|_{t=t_1} r_1^* + \frac{1}{2} \|r_1^*\|_{\left(\frac{d}{dt}\Delta^{-1}(t)\right)|_{t=t_1}}^2. \quad (4.24)$$

On the other hand, replacing r_1^* by r_2^* in (4.22), we get the inequality in the other direction

$$\begin{aligned} & \lim_{t_2 \rightarrow t_1^-} \frac{J_i^*(t_1, x(t_0)) - J_i^*(t_2, x(t_0))}{t_1 - t_2} \\ & \leq \frac{1}{2} \|x(t_0)\|_{\left(\frac{d}{dt}W(t)\Phi(t, t_0)\right)|_{t=t_1}}^2 - x^T(t_0) \left(\frac{d}{dt}W(t)\right)|_{t=t_1} r_1^* + \frac{1}{2} \|r_1^*\|_{\left(\frac{d}{dt}\Delta^{-1}(t)\right)|_{t=t_1}}^2. \end{aligned} \quad (4.25)$$

We conclude equality from (4.24) and (4.25).

We can derive $\lim_{t_2 \rightarrow t_1^+} \frac{J_i^*(t_1, x(t_0)) - J_i^*(t_2, x(t_0))}{t_1 - t_2}$ in the same way, which yields the same

equality result. Hence we conclude

$$\begin{aligned}
& \frac{dJ_i^*(t_f, x(t_0))}{dt_f} \\
&= \lim_{t \rightarrow t_f} \frac{J_i^*(t_f, x(t_0)) - J_i^*(t, x(t_0))}{t_f - t} \\
&= \frac{1}{2} \|x(t_0)\|_{\left(\frac{d}{dt}W(t)\Phi(t, t_0)\big|_{t=t_f}\right)}^2 - x_0^T \left(\frac{d}{dt}W(t)\big|_{t=t_f}\right) r^*(t_f) + \frac{1}{2} \|r^*(t_f)\|_{\left(\frac{d}{dt}\Delta^{-1}(t)\big|_{t=t_f}\right)}^2 \\
&= -\frac{1}{2} \|r^*(t_f) - x_f\|_{\Delta^{-1}(t_f)BR^{-1}B^T\Delta^{-1}(t_f)}^2 - (r^*(t_f))^T A^T \Delta^{-1}(t_f) (r^*(t_f) - x_f) \\
&= \|r^*(t_f)\|_{-2(\lambda_1^*)^2 DBR^{-1}B^T D + \lambda_1^*(A^T D + DA)}, \tag{4.26}
\end{aligned}$$

where the last step uses the Karush-Kuhn-Tucker (KKT) conditions of (EP1):

$$\frac{1}{2} \Delta^{-1}(t_f) (r^*(t_f) - x_f) + \lambda_1^* D r^*(t_f) = 0, \tag{4.27}$$

with λ_1^* being the positive Lagrange multiplier associated with the global minimum. □

Now we introduce an assumption on the dynamics of the mobile agent, which is necessary for a geometric property of the optimal trajectory at the terminal time.

Assumption 4.2. $B^T = \begin{bmatrix} 0_{2 \times 2} & B_2^T \end{bmatrix}$, with $\text{rank}(B_2) = 2$.

Assumption 4.2 states that the control u can only directly steer the time derivative of the velocity in the state $x(t)$, which is the acceleration of the mobile agent.

Partition matrix A and vector $r^*(t_f)$ in the following way,

$$A = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}, r^*(t_f) = \begin{bmatrix} r_p^*(t_f) \\ r_v^*(t_f) \end{bmatrix}, \tag{4.28}$$

where $A_i \in \mathbb{R}^{2 \times 2}$, $i = 1, 2, 3, 4$, and $r_p^*(t_f), r_v^*(t_f) \in \mathbb{R}^2$. The vectors $r_p^*(t_f)$ and $r_v^*(t_f)$ represent the projections of $r^*(t_f)$ onto the position subspace and velocity subspace, respectively.

With Assumption 4.2 and partitions in (4.5) and (4.28), we can simplify the partial derivative in (4.14) to the following form:

$$\dot{J}_i^*(t_f) = 2\lambda_1^*(r_p^*(t_f))^T D_p(A_1 r_p^*(t_f) + A_2 r_v^*(t_f)). \quad (4.29)$$

The next lemma relates the sign of the partial derivative in (4.29) to a geometric property of the optimal trajectory at the terminal time.

Lemma 4.2. *Let Assumption 4.2 hold and consider the nontrivial case $x_f \notin \mathcal{D}_2$. Then at a fixed terminal time $t_f \in (t_0, t_0 + T)$, the optimal trajectory of system (4.1) under the optimal inner stage control (4.13) will enter the target area \mathcal{D}_2^p if and only if $\dot{J}_i^*(t_f) < 0$, will exit \mathcal{D}_2^p if and only if $\dot{J}_i^*(t_f) > 0$ and will be tangent to the boundary of \mathcal{D}_2^p if and only if $\dot{J}_i^*(t_f) = 0$.*

Proof. Since $x_f \notin \mathcal{D}_2$, by complementary slackness, $\lambda_1^* > 0$ and $\|r^*(t_f)\|_D^2 = \|r_p^*(t_f)\|_{D_p}^2 = d_2^2$, i.e., $r_p^*(t_f)$ is on the boundary of \mathcal{D}_2^p . Then $D_p r_p^*(t_f)$ is the outward normal vector of \mathcal{D}_2^p at $r_p^*(t_f)$.

On the other hand, writing dynamics (4.1) in the partitioned form,

$$\begin{bmatrix} \dot{x}_p^*(t_f) \\ \dot{x}_v^*(t_f) \end{bmatrix} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \begin{bmatrix} x_p^*(t_f) \\ x_v^*(t_f) \end{bmatrix} + \begin{bmatrix} 0_{2 \times 2} \\ B_2 \end{bmatrix} u_i^*(t_f), \quad (4.30)$$

the first row reads $\dot{x}_p^*(t_f) = A_1 x_p^*(t_f) + A_2 x_v^*(t_f)$ which is the velocity $x_v^*(t_f)$ of the terminal state. Using the fact that the optimal trajectory yields $x_p^*(t_f) = r_p^*(t_f)$ and $x_v^*(t_f) = r_v^*(t_f)$, we have $x_v^*(t_f) = A_1 r_p^*(t_f) + A_2 r_v^*(t_f)$.

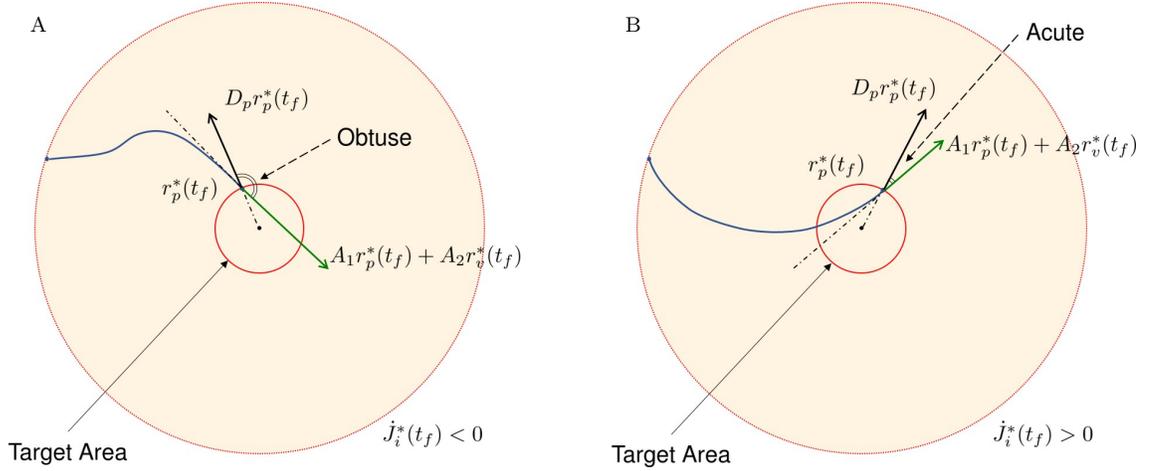


Figure 4.2: Relation between $\dot{J}_i^*(t_f)$ and the behavior of the optimal trajectory at the terminal time. (A) The optimal trajectory enters the target area at the terminal time when $\dot{J}_i^*(t_f) < 0$. (B) The optimal trajectory exits the target area when $\dot{J}_i^*(t_f) > 0$.

So, $\dot{J}_i^*(t_f) < 0$ if and only if the included angle is obtuse between two vectors $D_p r_p^*(t_f)$ and $A_1 r_p^*(t_f) + A_2 r_v^*(t_f)$. Since $D_p r_p^*(t_f)$ always points outwards \mathcal{D}_2^p , we conclude $A_1 r_p^*(t_f) + A_2 r_v^*(t_f)$ points inward \mathcal{D}_2^p , which is equivalent to that the optimal trajectory enters the target area at t_f . The cases of the derivative being nonnegative follow the same logic. \square

Figure 4.2 illustrates the relation between the sign of $\dot{J}_i^*(t_f)$ and the behavior of the optimal trajectory at the terminal time.

4.3.2 Augmented outer stage problem

By the LQR theory [20], for a fixed terminal state linear-quadratic problem, the optimal control is a linear combination of the state and the terminal state. This result and the linear dynamics imply that optimal cost is quadratic in the initial

state and the terminal state. This observation allows us to transform (P2) into the following equivalent problem.

$$\begin{aligned}
& \underset{r(t_0) \in \mathbb{R}^4, r(t_f) \in \mathbb{R}^4}{\text{minimize}} && \|r(t_0)\|_{\Xi_3}^2 + 2x_0^T \Xi_2 r(t_0) + \|x_0\|_{\Xi_1}^2 + \frac{1}{2} \|\Phi(t_f, t_0)r(t_0) - r(t_f)\|_{\Delta^{-1}(t_f)}^2 \\
& \text{subject to} && \|r(t_0)\|_D^2 = d_1^2, \\
& && \|r(t_f)\|_D^2 \leq d_2^2,
\end{aligned} \tag{EP2}$$

where the first three terms in the cost function constitute the minimum outer stage cost which is quadratic in x_0 and $r(t_0)$. The vector $r(t_0)$ is the desired switching state, which is an optimization variable here. The first constraint specifies that the desired switching state must be on the boundary of the time-costly area. The last term in the cost function and last constraint are adopted from (EP1). The expressions of Ξ_1 , Ξ_2 , and Ξ_3 are as follows

$$\Xi_1 = \frac{1}{2} \int_0^{t_0} \bar{\Phi}^T(t, 0) Q \bar{\Phi}(t, 0) + \bar{H}^T(t) R \bar{H}(t) dt, \tag{4.31a}$$

$$\Xi_2 = \frac{1}{2} \int_0^{t_0} -\bar{\Phi}^T(t, 0) Q \bar{L}(t) + \bar{H}^T(t) R (H(t) \bar{L}(t) - L(t)) dt, \tag{4.31b}$$

$$\Xi_3 = \frac{1}{2} \int_0^{t_0} \bar{L}^T(t) Q \bar{L}(t) + (H(t) \bar{L}(t) - L(t))^T R (H(t) \bar{L}(t) - L(t)) dt, \tag{4.31c}$$

with

$$\bar{H}(t) = -H(t) \bar{\Phi}(t, 0), \tag{4.32a}$$

$$\bar{L}(t) = \int_0^t \bar{\Phi}(t, \tau) B L(\tau) d\tau, \tag{4.32b}$$

and $\bar{\Phi}(t, \tau)$ being the state transition matrix associated with $\bar{A}(t) = A - BH(t)$ and hence the unique solution of

$$\frac{d}{dt}\bar{\Phi}(t, \tau) = \bar{A}(t)\bar{\Phi}(t, \tau), \forall t \geq \tau \geq 0, \quad (4.32a)$$

$$\bar{\Phi}(\tau, \tau) = I_4, \forall \tau \geq 0, \quad (4.32b)$$

and $H(t)$ as well as $L(t)$ being obtained by solving the following set of equations backwards in time,

$$-\dot{S}(t) = A^T S(t) + S(t)A - S(t)BR^{-1}B^T S(t) + Q, \quad (4.33a)$$

$$K(t) = R^{-1}B^T S(t), \quad (4.33b)$$

$$-\dot{V}(t) = (A - BK(t))^T V(t), \quad (4.33c)$$

$$\dot{P}(t) = V^T(t)BR^{-1}B^T V(t), \quad (4.33d)$$

where $t \leq t_0$, with boundary conditions $S(t_0) = 0_{4 \times 4}$, $V(t_0) = I_4$, $P(t_0) = 0_{4 \times 4}$, and

$$H(t) = K(t) - R^{-1}B^T V(t)P^{-1}(t)V^T(t), \quad (4.34a)$$

$$L(t) = R^{-1}B^T V(t)P^{-1}(t). \quad (4.34b)$$

Problem (EP2) seeks a switching state $r(t_0)$ on the boundary of the time-costly area and a terminal state $r(t_f)$ within the target area, such that the sum of the quadratic cost of the outer stage and the control effort of the inner stage is minimized. Note that $\text{val}(\text{EP2}) = J^*(t_f)$.

Denote the solution of (EP2) by $r^*(t_0)$ and $r^*(t_f)$. Then the optimal control u_o^* of (P2) is obtained by

$$u_o^*(t) = -H(t)x^*(t) - L(t)r^*(t_0), t \in [0, t_0), \quad (4.35)$$

where x^* denotes the optimal state under the optimal control u_o^* .

We now show how to solve (EP2). First rewrite (EP2) in a concise form to simplify notations. Let

$$\Gamma_1 = \begin{bmatrix} D & 0_{4 \times 4} \\ 0_{4 \times 4} & 0_{4 \times 4} \end{bmatrix}, \Gamma_2 = \begin{bmatrix} 0_{4 \times 4} & 0_{4 \times 4} \\ 0_{4 \times 4} & D \end{bmatrix}, \quad (4.36a)$$

$$M = \begin{bmatrix} M_1 & M_2 \\ M_2^T & M_3 \end{bmatrix}, y = \begin{bmatrix} r(t_0) \\ r(t_f) \end{bmatrix}, q = \begin{bmatrix} \Xi_2 x_0 \\ 0_{4 \times 1} \end{bmatrix}, \quad (4.36b)$$

where

$$M_1 = \Xi_3 + \frac{1}{2} \Phi^T(t_f, t_0) \Delta^{-1}(t_f) \Phi(t_f, t_0), \quad (4.37a)$$

$$M_2 = -\frac{1}{2} \Phi^T(t_f, t_0) \Delta^{-1}(t_f), \quad (4.37b)$$

$$M_3 = \frac{1}{2} \Delta^{-1}(t_f). \quad (4.37c)$$

Then (EP2) has the following form

$$\begin{aligned} & \underset{y \in \mathbb{R}^8}{\text{minimize}} && \|y\|_M^2 + 2q^T y \\ & \text{subject to} && \|y\|_{\Gamma_1}^2 = d_1^2, \\ & && \|y\|_{\Gamma_2}^2 \leq d_2^2, \end{aligned} \quad (\text{P4})$$

where we drop the term $\|x_0\|_{\Xi_1}^2$ in the cost function because it is a constant when x_0 is given.

Since $M \succ 0$, the cost function of (P4) is strictly convex. Hence, (P4) can be

solved by the following two relaxations,

$$\begin{aligned}
& \underset{y \in \mathbb{R}^8}{\text{minimize}} && \|y\|_M^2 + 2q^T y \\
& \text{subject to} && \|y\|_{\Gamma_1}^2 \leq d_1^2, \\
& && \|y\|_{\Gamma_2}^2 \leq d_2^2,
\end{aligned} \tag{P4-1}$$

$$\begin{aligned}
& \underset{y \in \mathbb{R}^8}{\text{minimize}} && \|y\|_M^2 + 2q^T y \\
& \text{subject to} && \|y\|_{\Gamma_1}^2 \geq d_1^2, \\
& && \|y\|_{\Gamma_2}^2 \leq d_2^2.
\end{aligned} \tag{P4-2}$$

Specifically, the optimal solution of (P4) is identical to the optimal solution of (P4- i), $i = 1, 2$, if (P4- i)'s first constraint is active at its optimal solution. (P4-1) is a convex QC2QP in which Slater's condition holds. Hence, strong duality holds for (P4-1) whose global optimal solution can be efficiently solved. Problem (P4-2) is nonconvex for which we can use Theorem 3.2 to check if strong duality holds by solving its Lagrange dual and semidefinite relaxation.

The Lagrange dual problem of (P4-2) is the following.

$$\begin{aligned}
& \underset{\mu_0, \mu_1, \mu_2 \in \mathbb{R}}{\text{maximize}} && \mu_0 \\
& \text{subject to} && Z(\mu_0, \mu_1, \mu_2) \succeq 0, \\
& && \mu_i \geq 0, i = 1, 2,
\end{aligned} \tag{P4-2D}$$

where

$$Z(\mu_0, \mu_1, \mu_2) = \begin{bmatrix} -\mu_0 + \mu_1 d_1^2 - \mu_2 d_2^2 & q^T \\ q & M - \mu_1 \Gamma_1 + \mu_2 \Gamma_2 \end{bmatrix}. \tag{4.38}$$

The semidefinite relaxation of (P4-2) is the following.

$$\begin{aligned}
& \underset{Y \in \mathcal{S}^9}{\text{minimize}} && M(q_0) \cdot Y \\
& \text{subject to} && M(q_i) \cdot Y \leq 0, i = 1, 2, \\
& && I_{00} \cdot Y = 1, \\
& && Y \succeq 0,
\end{aligned} \tag{P4-2R}$$

where

$$M(q_0) \stackrel{\text{def}}{=} \begin{bmatrix} 0 & q^T \\ q^T & M \end{bmatrix}, M(q_1) \stackrel{\text{def}}{=} \begin{bmatrix} d_1^2 & 0 \\ 0 & -\Gamma_1 \end{bmatrix}, M(q_2) \stackrel{\text{def}}{=} \begin{bmatrix} -d_2^2 & 0 \\ 0 & \Gamma_2 \end{bmatrix}, I_{00} \stackrel{\text{def}}{=} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}. \tag{4.39}$$

Let \hat{Y} and $(\hat{Z}, \hat{\mu}_0, \hat{\mu}_1, \hat{\mu}_2)$ denote a pair of optimal solutions for (P4-2R) and (P4-2D). By Theorem 3.2, strong duality holds for (P4-2) if and only if the pair \hat{Y} and $(\hat{Z}, \hat{\mu}_0, \hat{\mu}_1, \hat{\mu}_2)$ violates *Property I⁺*, which is restated as follows,

1. $\hat{\mu}_1 \hat{\mu}_2 \neq 0$;
2. $\text{rank}(\hat{Z}) = 7$;
3. $\text{rank}(\hat{Y}) = 2$ and there is a rank-one decomposition of \hat{Y} , $\hat{Y} = \hat{y}_1 \hat{y}_1^T + \hat{y}_2 \hat{y}_2^T$

such that

$$M(q_1) \cdot \hat{y}_i \hat{y}_i^T = 0, i = 1, 2, \tag{4.40}$$

$$(M(q_2) \cdot \hat{y}_1 \hat{y}_1^T)(M(q_2) \cdot \hat{y}_2 \hat{y}_2^T) < 0. \tag{4.41}$$

4. $M(q_1) \cdot \hat{y}_1 \hat{y}_2^T \neq 0$.

Now problem (EP2) can be solved in the following manner. First solve (P4)

for y^* using Algorithm 1. Then we can obtain the solution $r^*(t_0)$ and $r^*(t_f)$ of (EP2) by the partition of y in (4.36b).

Algorithm 1: Computing the optimal solution of (P4)

Input: $M, q, \Gamma_1, \Gamma_2, d_1, d_2$
Output: y^*

```

1 begin
2    $y^* \leftarrow$  solve (P4-1) by solving its dual ;
3   if  $y^*$  is active at the first constraint of (P4-1) then
4     return  $y^*$ ;
5   else
6      $(\hat{Z}, \hat{\mu}_0, \hat{\mu}_1, \hat{\mu}_2) \leftarrow$  solve (P4-2D);
7      $\hat{Y} \leftarrow$  solve (P4-2R);
8     if the pair  $(\hat{Z}, \hat{\mu}_0, \hat{\mu}_1, \hat{\mu}_2)$  and  $\hat{Y}$  violates Property  $I^+$  then
9        $y^* \leftarrow$  conduct a rank-one decomposition of  $\hat{Y}$ 
10    else
11       $y^* \leftarrow$  solve (P4-2) by a nonlinear solver (e.g. MATLAB
12        fmincon)
13    end
14  return  $y^*$ ;
15 end
```

The optimal value $J^*(t_f)$ of (P2) is

$$J^*(t_f) = \text{val}(\text{EP2}) = \|r^*(t_0)\|_{M_1}^2 + 2(r^*(t_0))^T M_2 r^*(t_f) + \|r^*(t_f)\|_{M_3}^2 + \|x_0\|_{\Xi_1}^2. \quad (4.42)$$

And the following lemma states the properties of (EP2).

Lemma 4.3. For $t_f \in (t_0, t_0 + T)$,

1. The function $J^*(t_f)$ is continuous w.r.t. t_f .
2. The optimal switching state $r^*(t_0)$ and the optimal terminal state $r^*(t_f)$ are vector-valued functions of t_f and are continuous w.r.t. t_f .

3. The derivative of $J^*(t_f)$ w.r.t. t_f , denoted by $\dot{J}^*(t_f)$, is

$$\dot{J}^*(t_f) = \begin{cases} \|r^*(t_f)\|_{\Theta_2}^2, & \text{if } r_{fp} \notin \mathcal{D}_2^p, \\ 0, & \text{if } r_{fp} \in \text{int}(\mathcal{D}_2^p), \end{cases} \quad (4.43)$$

where $\Theta_2 = -2(\lambda_2^*)^2 DBR^{-1}B^T D + \lambda_2^*(A^T D + DA)$ and λ_2^* satisfies

$$\Delta^{-1}(t_f)(r^*(t_f) - \Phi(t_f, t_0)r^*(t_0)) = -2\lambda_2^* D r^*(t_f), \quad (4.44)$$

and r_{fp} denotes to the projection of $\Phi(t_f, t_0)r^*(t_0)$ onto the position subspace.

Proof. We provide a sketch of the proof as the following. The proof uses the identical method applied in the proof of Lemma 4.1, i.e., using the optimality of minimizers at different terminal times to deduce symmetric inequalities and hence equalities. The derivation of λ_2^* (and hence Θ_2) refers to the zero gradient condition $\Delta^{-1}(t_f)(r^*(t_f) - \Phi(t_f, t_0)r^*(t_0)) + 2\lambda_2^* D r^*(t_f) = 0$ in the KKT conditions² of (EP2), where $\lambda_2^* \in \mathbb{R}_+$ denotes the Lagrange multiplier associated with the second constraint. \square

The next lemma connects the geometric property of the optimal trajectory at the terminal time with the sign of the derivative in (4.43).

Lemma 4.4. *Let Assumption 4.2 hold and consider the nontrivial case $r_{fp} \notin \mathcal{D}_2^p$. Then at a fixed terminal time $t_f \in (t_0, t_0 + T)$, the optimal trajectory of system (4.1) under the optimal controls (4.13) and (4.35) will enter the target area \mathcal{D}_2^p if and only if $\dot{J}^*(t_f) < 0$, will exit \mathcal{D}_2^p if and only if $\dot{J}^*(t_f) > 0$ and will be tangent to the boundary of \mathcal{D}_2^p if and only if $\dot{J}^*(t_f) = 0$.*

²Linear independence constraint qualification (LICQ) holds for (EP2)

Proof. A sketch of the proof is the following. We adopt the identical method used in the proof of Lemma 4.2, except that x_f and λ_1^* are replaced by $\Phi(t_f, t_0)r^*(t_0)$ and λ_2^* , respectively. \square

4.3.3 Solution of the original problem (P)

To find a solution of (P), we shall find a locally optimal terminal time $t_f^* \in (t_0, t_0 + T)$ of (P3). The first-order necessary condition states that a local minimizer $t_f^* \in (t_0, t_0 + T)$ of (P3) satisfies

$$\dot{J}^*(t_f^*) + \dot{\phi}(t_f^* - t_0) = 0, \quad (4.45)$$

where $\dot{\phi}(t_f^* - t_0) = \left. \frac{d\phi(t)}{dt} \right|_{t=t_f^*-t_0}$. We suggest the bisection method because the values of $J^*(t_f)$ and $\dot{J}^*(t_f)$ cannot be evaluated until problem (EP2) is solved for a specific value of t_f . The initial left endpoint of bisection interval is t_0 because $\lim_{t_f \rightarrow t_0^+} J^*(t_f) = \infty$ and $\lim_{t_f \rightarrow t_0^+} \dot{J}^*(t_f) = -\infty$. The initial right endpoint t_r is determined heuristically by searching the minimum $t_r \in (t_0, t_0 + T]$ such that the derivative $\dot{J}^*(t_r)$ turns positive. Once t_f^* is found, the optimal controls $u_o^*(t)$ and $u_i^*(t)$ are obtained by (4.35) and (4.13) using $r^*(t_f^*)$ and $r^*(t_0)$, respectively, where $r^*(t_f^*)$ and $r^*(t_0)$ are solved in (EP2) with $t_f = t_f^*$.

Remark 4.1. *Since $\phi(\cdot)$ is a strictly increasing function with continuous derivative, it must hold that $\dot{J}^*(t_f^*) < 0$. By Lemma 4.4, this implies the optimal terminal time trajectory must enter the target area at t_f^* .*

4.4 Numerical example

In this section, we illustrate the procedure of finding an optimal terminal time by the following numerical example. The results in this section are obtained by CVX [21] (solving (P4-1)(P4-2D)(P4-2R)). The followings are the values and functions applied in the simulation.

$$A = \begin{bmatrix} 0_{2 \times 2} & I_2 \\ 0_{2 \times 2} & 0_{2 \times 2} \end{bmatrix}, B = \begin{bmatrix} 0_{2 \times 2} \\ I_2 \end{bmatrix}, Q = I_4, R = I_2, D_p = I_2, t_0 = 3, T = 2,$$

$$\epsilon = 10^{-4}, d_1 = 4, d_2 = 1, x_0 = \begin{bmatrix} 8 & -3 & 0 & 2 \end{bmatrix}^T, \phi(t) = 200t.$$

The bisection interval and minimum cost at each iteration are shown in Figure 4.3. The procedure stops at the 12th iteration because the tolerance ϵ has been reached such that difference between values of $J^*(t_f) + \phi(t_f - t_0)$ in two consecutive iterations is less than or equal to ϵ . A locally optimal terminal time $t_f^* = 3.417$ has been found with the optimal switching state

$$r^*(t_0) = \begin{bmatrix} 3.963 & -0.542 & -5.673 & 0.803 \end{bmatrix}^T,$$

the optimal terminal state

$$r^*(t_f^*) = \begin{bmatrix} 0.968 & -0.125 & -7.851 & 1.086 \end{bmatrix}^T,$$

and the minimum cost being 187.486. The optimal values of (P3) is shown in Figure 4.4, which indicates the t_f^* we found attains a local minimum. Figure 4.5 shows the optimal trajectory. Notice that the optimal trajectory *enters* the target area, as indicated by Remark 4.1.

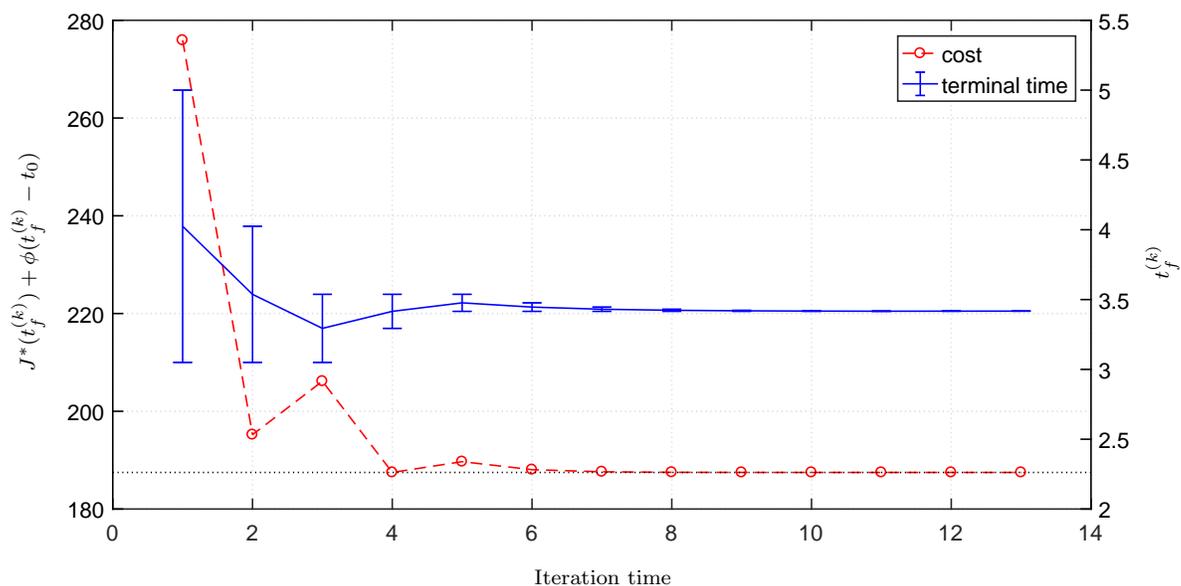


Figure 4.3: Minimum cost and terminal time with bisection intervals through iterations. The vertical bars attached to the plot of terminal time show the bisection intervals applied at each iteration. The terminal time at the k^{th} iteration is denoted by $t_f^{(k)}$.

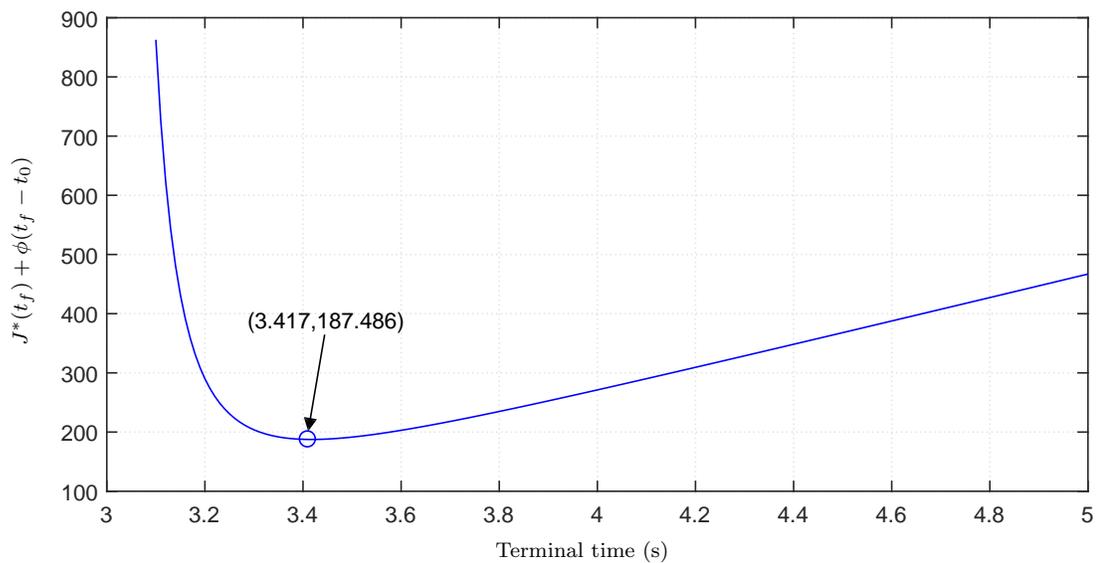


Figure 4.4: Optimal values of (P3) when the terminal time t_f varies

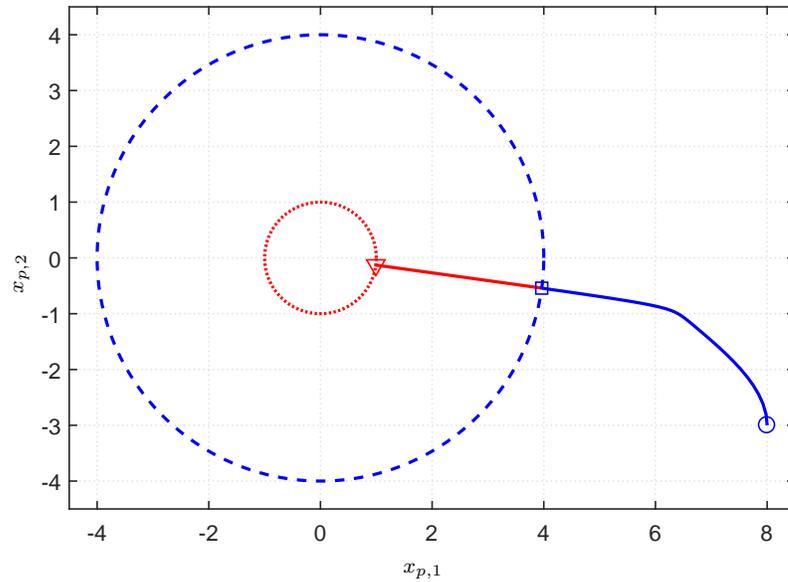


Figure 4.5: The optimal trajectory is the curve starting from the initial position (marked by the small blue circle), crossing the boundary of the time-costly area at the position marked by the blue square, and *entering* the target area at the position marked by the red triangle. The blue curve and red curve correspond to the trajectory by the outer stage controller and inner stage controller, respectively. The dashed blue circle and dotted red circle represent the boundaries of the time-costly area and target area, respectively.

Chapter 5: Two-stage Optimal Control for Target Reaching Inside a Denied Area

In this Chapter, we consider another case of the special area: denied area. The denied area, which generalizes the GPS-denied area commonly known in the robotics research, is defined such that the mobile agent cannot receive measurements of its position nor velocity once it enters the area. The difference between the time-costly area and the denied area is that the former stresses the criticality of time while the latter stresses the unavailability of measurements which are necessary for feedback control. We also consider perturbation in this problem for which we propose a robust controller.

5.1 Related literature

Research on operations of a mobile agent inside a denied area has a rich literature. From the perspective of localization and navigation, the main effort is placed on maintaining a local estimation of the position and pose of the mobile agent when the global estimation based on GPS is unavailable in the denied area. Such estimation typically uses onboard sensors to collect information of the surroundings (e.g. by camera and lidar) and to measure the relative motion (e.g. by odometer

and inertial measurement unit). Then the estimation is performed using a probabilistic method (e.g. extended Kalman filter [22]), or using nonlinear optimization techniques (e.g. factor graphs [23]). Since the lack of GPS measurement induces nonobservability of the position and heading [24, 25], the estimation needs to refer to a local reference frame for the full-state observability [26]. In this chapter, our focus is on a control strategy that anticipates the future operations inside the denied area, instead of estimating the state of the mobile agent there. Hence, we allow the control inside the denied area to be open-loop, but pre-determined before its entry to such area.

5.2 Discrete-time two-stage optimal control problem

We start by restating the two-stage optimal control problem in Chapter 4 under discrete-time dynamics. Consider the following linear discrete-time dynamics of a mobile agent,

$$x_{k+1} = A^{\mathfrak{D}}x_k + B^{\mathfrak{D}}u_k, \quad k = 0, 1, \dots, \quad (5.1)$$

where $x_k \in \mathbb{R}^4$ is the state variable, $u_k \in \mathbb{R}^2$ is the control variable, and the system matrices $(A^{\mathfrak{D}}, B^{\mathfrak{D}})$ are discretized from (4.2), with sampling time T_s and method introduced in [27],

$$\begin{bmatrix} A^{\mathfrak{D}} & B^{\mathfrak{D}} \\ * & I_2 \end{bmatrix} = \exp \left(\begin{bmatrix} A & B \\ 0_{2 \times 4} & 0_{2 \times 2} \end{bmatrix} T_s \right), \quad N = \frac{t_0}{T_s}. \quad (5.2)$$

The goal of this problem is to steer the mobile agent to the target that is enclosed within the denied area. We stick to the decomposition introduced in Chapter

4 such that the problem has an outer stage and an inner stage, where the time-costly area is replaced by the denied area. The outer stage seeks a sequence of control $\mathbf{u}_{0:N_s-1}^o$ such that the mobile agent is steered to reach the boundary of the denied area at the switching time N_s , with a minimum quadratic cost $\sum_{k=0}^{N_s-1} \|u_k^o\|_{R^{\mathfrak{D}}}^2 + \|x_k\|_{Q^{\mathfrak{D}}}^2$, where $R^{\mathfrak{D}} = T_s R$ and $Q^{\mathfrak{D}} = T_s Q$. The inner stage seeks a sequence of control $\mathbf{u}_{N_s:N-1}^i$ such that the mobile agent is steered to reach the target area at the terminal time N , with a minimum control effort $\sum_{k=N_s}^{N-1} \|u_k^i\|_{R^{\mathfrak{D}}}^2$. Notice that measurement of position and velocity is unavailable upon the mobile agent's arrival at the denied area. Hence the inner stage only consider the cost containing a function of the control, not the states which are unavailable for evaluation.

The two-stage problem has the following form

$$\begin{aligned}
& \underset{\mathbf{u}_{0:N-1}}{\text{minimize}} && \frac{1}{2} \sum_{k=0}^{N_s-1} \|u_k^o\|_{R^{\mathfrak{D}}}^2 + \|x_k\|_{Q^{\mathfrak{D}}}^2 + \frac{1}{2} \sum_{k=N_s}^{N-1} \|u_k^i\|_{R^{\mathfrak{D}}}^2 \\
& \text{subject to} && x_{k+1} = A^{\mathfrak{D}} x_k + B^{\mathfrak{D}} u_k, k = 0, 1, \dots, N-1, \\
& && u_k = \begin{cases} u_k^o, & \text{if } k = 0, 1, \dots, N_s-1, \\ u_k^i, & \text{if } k = N_s, N_s+1, \dots, N-1, \end{cases} \quad (\text{P5}) \\
& && \|x_{N_s}\|_D^2 = d_1^2, \\
& && \|x_N\|_D^2 \leq d_2^2.
\end{aligned}$$

By [20], this problem can be transformed to an equivalent form

$$\begin{aligned}
& \underset{r_{N_s}, r_N \in \mathbb{R}^4}{\text{minimize}} && \|x_0\|_{\Xi_1^{\mathfrak{D}}}^2 + 2x_0^T \Xi_2^{\mathfrak{D}} r_{N_s} + \|r_{N_s}\|_{\Xi_3^{\mathfrak{D}}}^2 + \frac{1}{2} \|(A^{\mathfrak{D}})^{N-N_s} r_{N_s} - r_N\|_{(\Delta^{\mathfrak{D}}(N_s, N))^{-1}}^2 \\
& \text{subject to} && \|r_{N_s}\|_D^2 = d_1^2, \\
& && \|r_N\|_D^2 \leq d_2^2.
\end{aligned} \tag{EP5}$$

where

$$\Xi_1^{\mathfrak{D}} = \frac{1}{2} \sum_{k=0}^{N-1} (\bar{\Phi}^{\mathfrak{D}}(k, 0))^T Q^{\mathfrak{D}} \bar{\Phi}^{\mathfrak{D}}(k, 0) + (\bar{H}_k^{\mathfrak{D}})^T R^{\mathfrak{D}} \bar{H}_k^{\mathfrak{D}}, \tag{5.3a}$$

$$\Xi_2^{\mathfrak{D}} = \frac{1}{2} \sum_{k=0}^{N-1} -(\bar{\Phi}^{\mathfrak{D}}(k, 0))^T Q^{\mathfrak{D}} \bar{L}_k^{\mathfrak{D}} - (\bar{H}_k^{\mathfrak{D}})^T R^{\mathfrak{D}} (H_k^{\mathfrak{D}} \bar{L}_k^{\mathfrak{D}} - L_k^{\mathfrak{D}}), \tag{5.3b}$$

$$\Xi_3^{\mathfrak{D}} = \frac{1}{2} \sum_{k=0}^{N-1} (\bar{L}_k^{\mathfrak{D}})^T Q^{\mathfrak{D}} \bar{L}_k^{\mathfrak{D}} + (H_k^{\mathfrak{D}} \bar{L}_k^{\mathfrak{D}} - L_k^{\mathfrak{D}})^T R^{\mathfrak{D}} (H_k^{\mathfrak{D}} \bar{L}_k^{\mathfrak{D}} - L_k^{\mathfrak{D}}), \tag{5.3c}$$

$$\Delta^{\mathfrak{D}}(N_s, N) = \sum_{k=N_s}^{N-1} (A^{\mathfrak{D}})^{N-1-k} B^{\mathfrak{D}} (R^{\mathfrak{D}})^{-1} (B^{\mathfrak{D}})^T ((A^{\mathfrak{D}})^T)^{N-1-k}, \tag{5.3d}$$

with

$$K_k = ((B^{\mathfrak{D}})^T S_{k+1} B^{\mathfrak{D}} + R^{\mathfrak{D}})^{-1} (B^{\mathfrak{D}})^T S_{k+1} A^{\mathfrak{D}}, \quad (5.4a)$$

$$S_k = (A^{\mathfrak{D}})^T S_{k+1} (A^{\mathfrak{D}} - B^{\mathfrak{D}} K_k) + Q^{\mathfrak{D}}, \quad (5.4b)$$

$$V_k = (A^{\mathfrak{D}} - B^{\mathfrak{D}} K_k)^T V_{k+1}, \quad (5.4c)$$

$$P_k = P_{k+1} - V_{k+1}^T B^{\mathfrak{D}} ((B^{\mathfrak{D}})^T S_{k+1} B^{\mathfrak{D}} + R^{\mathfrak{D}})^{-1} (B^{\mathfrak{D}})^T V_{k+1}, \quad (5.4d)$$

$$K_k^u = ((B^{\mathfrak{D}})^T S_{k+1} B^{\mathfrak{D}} + R^{\mathfrak{D}})^{-1} (B^{\mathfrak{D}})^T, \quad (5.4e)$$

$$H_k^{\mathfrak{D}} = K_k - K_k^u V_{k+1} P_k^{-1} V_k^T, \quad (5.4f)$$

$$L_k^{\mathfrak{D}} = K_k^u V_{k+1} P_k^{-1}, \quad (5.4g)$$

$$\bar{\Phi}^{\mathfrak{D}}(k, j) = \begin{cases} \prod_{i=j}^{k-1} (A^{\mathfrak{D}} - B^{\mathfrak{D}} H_i^{\mathfrak{D}}) & , \text{ if } k > j, \\ I & , \text{ if } k = j, \end{cases} \quad (5.4h)$$

$$\bar{H}_k^{\mathfrak{D}} = H_k^{\mathfrak{D}} \bar{\Phi}^{\mathfrak{D}}(k, 0), \quad (5.4i)$$

$$\bar{L}_k^{\mathfrak{D}} = \sum_{j=0}^{k-1} \bar{\Phi}^{\mathfrak{D}}(k, j) B L_j^{\mathfrak{D}}, \quad (5.4j)$$

and boundary conditions $S_{N_o} = 0_{4 \times 4}$, $V_{N_o} = I$ and $P_{N_o} = 0_{4 \times 4}$ and here $N_o = N_s$.

Problem (EP5) has an identical structure to (EP2). Hence we can recast (EP5) in the form of (P4) and refer to Algorithm 1 for a solution. Denote the solution of (EP5) as \bar{r}_{N_s} and \bar{r}_N . Then the optimal controls of (P5), denoted by \bar{u}_k^o and \bar{u}_k^i are given by

$$\bar{u}_k^o = -H_k^{\mathfrak{D}} \bar{x}_k - L_k^{\mathfrak{D}} \bar{r}_{N_s}, k = 0, 1, \dots, N_s - 1, \quad (5.5a)$$

$$\bar{u}_k^i = -(R^{\mathfrak{D}})^{-1} (B^{\mathfrak{D}})^T ((A^{\mathfrak{D}})^T)^{N-k-1} (\Delta^{\mathfrak{D}}(N_s, N))^{-1} ((A^{\mathfrak{D}})^{N-N_s} \bar{r}_{N_s} - \bar{r}_N), \quad (5.5b)$$

for $k = N_s, N_s + 1, \dots, N - 1$, where \bar{x}_k denotes the state steered by control input

\bar{u}_{k-1}^o .

The advantage of casting (P5) into (EP5) is that the latter significantly decreases the size of the optimization problem: (EP5) only computes the optimal switching state and the optimal terminal state online, using offline computable matrix coefficients $\Xi_1^{\mathfrak{D}}, \Xi_2^{\mathfrak{D}}, \Xi_3^{\mathfrak{D}}$ and $\Delta^{\mathfrak{D}}(N_s, N)$. And the optimal control sequence can be recovered using (5.5a) and (5.5b) once the two optimal states are solved.

5.3 Problem formulation with perturbation

In this section, we illustrate how to modify the problem formulation to incorporate the perturbation. First, consider the dynamics of the mobile agent with an additive perturbation, i.e.,

$$x_{k+1} = A^{\mathfrak{D}}x_k + B^{\mathfrak{D}}u_k + w_k, k = 0, 1, \dots, \quad (5.6)$$

where $w_k \in \mathbb{R}^4$ is the perturbation. The perturbation is bounded within a convex polytope $\mathcal{W} \subseteq \mathbb{R}^4$ with m vertices $\{v_1, v_2, \dots, v_m\}$, but are otherwise unknown. The polytope \mathcal{W} has an equivalent \mathcal{H} -representation by linear inequalities, i.e.,

$$\mathcal{W} = \{x \in \mathbb{R}^4 | Gx \leq g\}, \quad (5.7)$$

where G and g have conformed dimensions. We assume \mathcal{W} contains the origin.

Perturbation can fail the controller (5.5), which is designed using deterministic dynamics, as shown in Figure 5.1. The challenges brought by perturbations are summarized into the following aspects.

1. Position of stage switching. The constraint that the switching state is on the boundary of the denied area is hardly feasible because the boundary has a zero

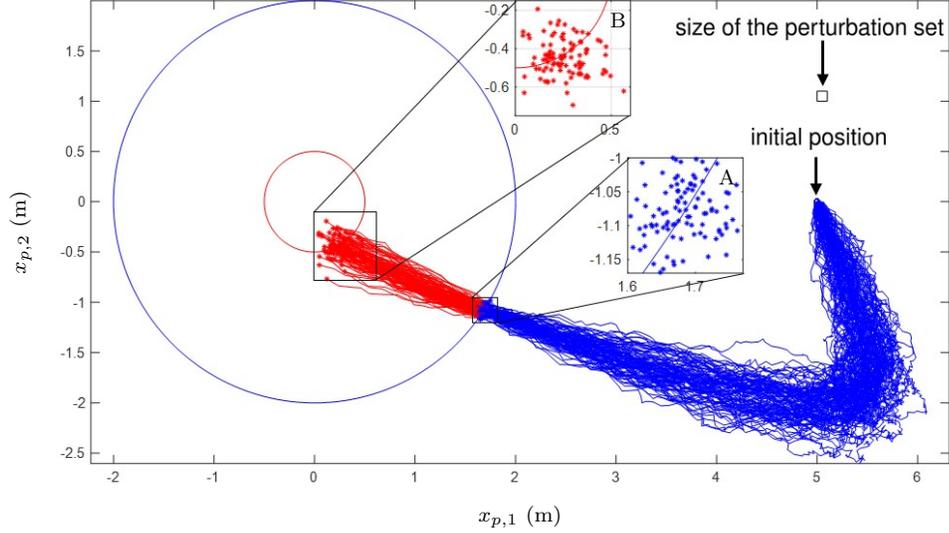


Figure 5.1: Illustration of the failure of controller (5.5) subject to perturbations. The figure shows trajectories of the mobile agent, with dynamics (5.6), controlled by (5.5) in 100 simulations. The blue trajectories are steered by (5.5a) while the red ones are steered by (5.5b). (A) Enlarged view of switching positions near the boundary of the denied area. Only the ones on the right, which are outside the denied area, are proper switching positions. (B) Enlarged view of terminal positions near the target area. Only the ones on top left, which are inside the target area, are proper terminal positions.

measure in \mathbb{R}^4 . In Figure 5.1(A), there is no switching position that arrives right on the boundary of the denied area.

2. Time of stage switching. Such time needs to be selected such that the switching happens sufficiently close but outside the denied area. In Figure 5.1(A), almost half of all trajectories have the switching positions inside the denied area, which is improper.
3. Target reaching. The mobile agent needs to enter the target despite perturba-

tions. In Figure 5.1(B), almost half of all trajectories do not reach the target area, despite the fact that their switching positions might not be proper either.

The goal of the problem is to find a control sequence such that the mobile agent with dynamics (5.6) can be steered to reach the target area enclosed within the denied area, subject to the perturbation. As before, the problem can be decomposed into two stages, but these two stages here differ with the stages defined under the perturbation-free dynamics. The presence of the perturbation makes it impossible to control the mobile agent to precisely arrive at the boundary of the denied area and hence to switch stage. The time of stage switching must be carefully determined at which the mobile agent is outside, but sufficiently close to, the denied area. Therefore, under current settings, the switching time is a decision variable. Once the mobile agent switches stage, it relies on the controller of the second stage to reach the target area.

Because of the advantage of efficient online computation, we stick to the controls u_k^o and u_k^i in the form of (5.5a) and (5.5b), i.e.,

$$u_k^o = -H_k^{\mathfrak{D}} x_k - L_k^{\mathfrak{D}} r_{N_s}, k = 0, 1, \dots, N_s - 1, \quad (5.8a)$$

$$u_k^i = -(R^{\mathfrak{D}})^{-1} (B^{\mathfrak{D}})^T ((A^{\mathfrak{D}})^T)^{N-k-1} (\Delta^{\mathfrak{D}}(N_s, N))^{-1} ((A^{\mathfrak{D}})^{N-N_s} x_0^i - r_N), \quad (5.8b)$$

for $k = N_s, N_s + 1, \dots, N$, where x_0^i denotes an arbitrary initial state of the inner stage. The deterministic switching state r_{N_s} and the deterministic terminal state r_N are to be determined later to adapt to the perturbation, together with new optimization variables introduced.

We can now make predictions of states by set propagation, given dynamics

(5.6) and controls in (5.8). As the control sequence $\mathbf{u}_{0:N_s-1}^o$ steers the mobile agent to the deterministic switching state r_{N_s} , the switching state $x_{N_s}^o$ under a realization of perturbations $\mathbf{w}_{0:N_s-1}$ is

$$x_{N_s}^o = \bar{\Phi}^{\mathfrak{D}}(N_s, 0)x_0 - \sum_{k=0}^{N_s-1} \bar{\Phi}^{\mathfrak{D}}(N_s, k)B^{\mathfrak{D}}L_k^{\mathfrak{D}}r_{N_s} + \sum_{k=0}^{N_s-1} \bar{\Phi}^{\mathfrak{D}}(N_s, k)w_k. \quad (5.9)$$

Since $w_k \in \mathcal{W}$ for all k , we can characterize the predicted set of switching states (PSSS), denoted by $\mathcal{X}_{N_s}^o$, as the following,

$$\begin{aligned} \mathcal{X}_{N_s}^o &= (\bar{\Phi}^{\mathfrak{D}}(N_s, 0)x_0 - \sum_{k=0}^{N_s-1} \bar{\Phi}^{\mathfrak{D}}(N_s, k)B^{\mathfrak{D}}L_k^{\mathfrak{D}}r_{N_s}) \oplus \bigoplus_{k=0}^{N_s-1} \bar{\Phi}^{\mathfrak{D}}(N_s, k)\mathcal{W} \\ &= r_{N_s} \oplus \mathcal{W}_{N_s}^o, \end{aligned} \quad (5.10)$$

where $\mathcal{W}_{N_s}^o$ is the propagated set of perturbation at N_s , i.e.,

$$\mathcal{W}_{N_s}^o = \bigoplus_{k=0}^{N_s-1} \bar{\Phi}^{\mathfrak{D}}(N_s, k)\mathcal{W}. \quad (5.11)$$

As the control sequence $\mathbf{u}_{N_s:N-1}^i$ steers the mobile agent to a deterministic terminal state r_N , the terminal state under a realization of perturbations $\mathbf{w}_{N_s:N-1}$ is

$$\begin{aligned} x_N^i &= (A^{\mathfrak{D}})^{N-N_s}x_0^i - \sum_{k=N_s}^{N-1} (A^{\mathfrak{D}})^{N-1-k}B^{\mathfrak{D}}(R^{\mathfrak{D}})^{-1}(B^{\mathfrak{D}})^T((A^{\mathfrak{D}})^T)^{N-1-k}r_N \\ &\quad + \sum_{k=N_s}^{N-1} (A^{\mathfrak{D}})^{N-1-k}w_k. \end{aligned} \quad (5.12)$$

Since $w_k \in \mathcal{W}$ for all k , we can characterize the predicted set of terminal states (PSTS), denoted by \mathcal{X}_N^i , as the following,

$$\begin{aligned} \mathcal{X}_N^i &= ((A^{\mathfrak{D}})^{N-N_s}x_0^i - \sum_{k=N_s}^{N-1} (A^{\mathfrak{D}})^{N-1-k}B^{\mathfrak{D}}(R^{\mathfrak{D}})^{-1}(B^{\mathfrak{D}})^T((A^{\mathfrak{D}})^T)^{N-1-k}r_N) \\ &\quad \oplus \bigoplus_{k=N_s}^{N-1} (A^{\mathfrak{D}})^{N-1-k}\mathcal{W} \end{aligned} \quad (5.13)$$

$$= r_N \oplus \mathcal{W}_N^i, \quad (5.14)$$

where \mathcal{W}_N^i is the propagated set of perturbation at N , i.e.,

$$\mathcal{W}_N^i = \bigoplus_{k=N_s}^{N-1} (A^{\mathfrak{D}})^{N-1-k} \mathcal{W}. \quad (5.15)$$

Notice that the PSSS $\mathcal{X}_{N_s}^o$ and PSTS \mathcal{X}_N^i are the propagated set of perturbation $\mathcal{W}_{N_s}^o$ and \mathcal{W}_N^i shifted by the deterministic switching state r_{N_s} and the deterministic terminal state r_N , respectively.

The sets $\mathcal{W}_{N_s}^o$ and \mathcal{W}_N^i are convex polytopes and are pre-computable once N and N_s are given. We adopt both the \mathcal{V} -representation and \mathcal{H} -representation of $\mathcal{W}_{N_s}^o$ and \mathcal{W}_N^i . Therefore,

$$\mathcal{W}_{N_s}^o = \text{convh}\{v_1^o, v_2^o, \dots, v_{m(N_s)}^o\} = \{x \in \mathbb{R}^4 | G_{N_s}^o x \leq g_{N_s}^o\}, \quad (5.16)$$

$$\mathcal{W}_N^i = \text{convh}\{v_1^i, v_2^i, \dots, v_{m(N)}^i\} = \{x \in \mathbb{R}^4 | G_N^i x \leq g_N^i\}, \quad (5.17)$$

where $m(N_s)$ and $m(N)$ denote the number of vertices of $\mathcal{W}_{N_s}^o$ and \mathcal{W}_N^i ; $G_{N_s}^o, g_{N_s}^o, G_N^i$, and g_N^i are of conformed dimensions.

We evaluate the cost incurred in the outer stage by the sum of quadratic functions of control actions $\mathbf{u}_{0:N_s-1}^o$ and corresponding states $\mathbf{x}_{0:N_s-1}^o$ under the deterministic dynamics (5.1). The cost has the following form,

$$\frac{1}{2} \sum_{k=0}^{N_s-1} \|u_k^o\|_{R^{\mathfrak{D}}}^2 + \|x_k\|_{Q^{\mathfrak{D}}}^2. \quad (5.18)$$

Since control u_k^o in (5.8a) is applied, then the quadratic cost equals

$$\|x_0^o\|_{\Xi_1}^2 + 2r_{N_s}^T \Xi_2 x_0^o + \|r_{N_s}\|_{\Xi_3}^2, \quad (5.19)$$

where $x_0^o \in \mathbb{R}^4$ is the initial state which is known.

We need to place $\mathcal{X}_{N_s}^o$ outside the denied area because this is where stage switching happens. In this way, the switching state x_{N_s} will stay outside the denied

area because $x_{N_s} \in \mathcal{X}_{N_s}^o$. We can formulate this requirement in the following manner. Find a vector $r_s \in \mathbb{R}^4$ on the boundary of the denied area, such that the tangent plane (which is also the supporting plane) of the denied area at r_s separates the denied area and $\mathcal{X}_{N_s}^o$, i.e.,

$$(r_{N_s} + v_k^o - r_s)^T D r_s \geq 0, k = 1, 2, \dots, m(N_s), \quad (5.20)$$

$$\|r_s\|_D^2 = d_1^2. \quad (5.21)$$

The geometric illustration of the above two equations is shown in Figure 5.2.

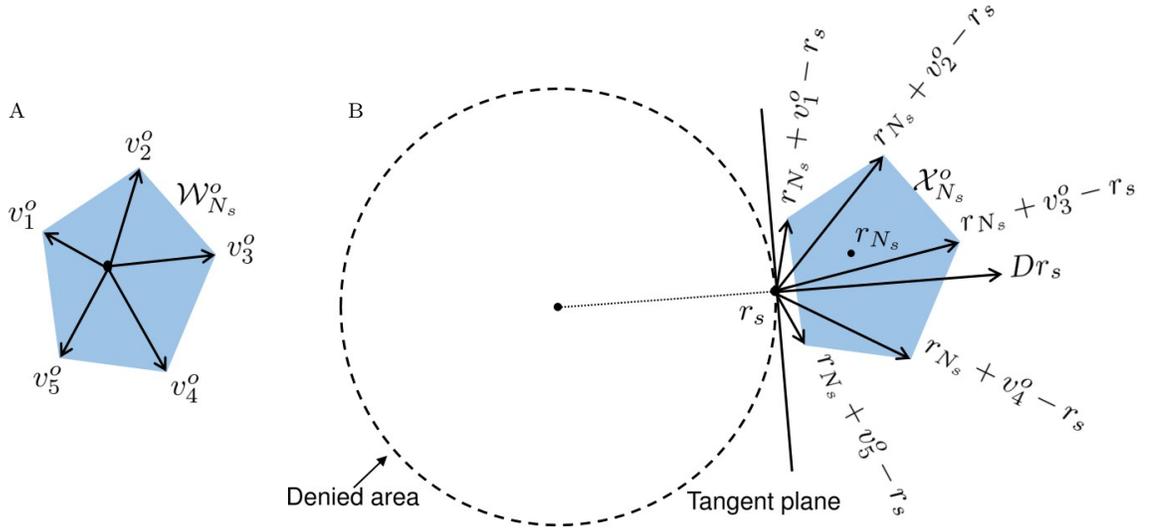


Figure 5.2: Geometric illustration of (5.20) and (5.21). (A) The vertices $\{v_1^o, v_2^o, v_3^o, v_4^o, v_5^o\}$ of $\mathcal{W}_{N_s}^o$ are shown. (B) The set $\mathcal{X}_{N_s}^o$ is placed outside the denied area if the included angle between the normal vector $D r_s$ and $r_{N_s} + v_k^o - r_s$ is acute for $k = 1, 2, \dots, 5$.

Then $\mathcal{X}_{N_s}^o$ is placed outside the denied area, due to the convexity of the denied area and $\mathcal{W}_{N_s}^o$. Vectors r_s and r_{N_s} are the optimization variables to be determined since $\mathcal{W}_{N_s}^o$ is fixed for a given N_s . The outer stage problem can be formulated in

the following form,

$$\begin{aligned}
& \underset{r_{N_s}, r_s \in \mathbb{R}^4}{\text{minimize}} && \|x_0^o\|_{\Xi_1}^2 + 2r_{N_s}^T \Xi_2 x_0^o + \|r_{N_s}\|_{\Xi_3}^2 \\
& \text{subject to} && (r_{N_s} + v_k^o - r_s)^T D r_s \geq 0, k = 1, 2, \dots, m(N_s), \\
& && \|r_s\|_D^2 = d_1^2.
\end{aligned} \tag{P6}$$

We evaluate the cost incurred in the second stage by the control effort

$$\frac{1}{2} \sum_{k=N_s}^{N-1} \|u_k^i\|_{\mathbb{R}^{\mathfrak{D}}}^2, \tag{5.22}$$

and since u_k^i in (5.8b) is applied, this cost equals

$$\frac{1}{2} \|(A^{\mathfrak{D}})^{N-N_s} x_0^i - r_N\|_{(\Delta^{\mathfrak{D}}(N_s, N))^{-1}}^2, \tag{5.23}$$

where $x_0^i \in \mathbb{R}^4$ denotes an arbitrary initial state of the inner stage, which is known.

We will place \mathcal{X}_N^i inside the denied area so that the terminal state x_N can reach the target area because $x_N \in \mathcal{X}_N^i$. As \mathcal{W}_N^i is fixed for a given N , we need to find r_N such that all the vertices of \mathcal{X}_N^i are inside the target area, i.e.,

$$\|r_N + v_k^i\|_D^2 \leq d_2^2, k = 1, 2, \dots, m(N), \tag{5.24}$$

where r_N is an optimization variable to be determined. Due to convexity of \mathcal{W}_N^i and the target area, \mathcal{X}_N^i is placed inside the denied area. The geometric illustration of (5.24) is shown in Figure 5.3.

The inner stage problem can be formulated as

$$\begin{aligned}
& \underset{r_N \in \mathbb{R}^4}{\text{minimize}} && \frac{1}{2} \|(A^{\mathfrak{D}})^{N-N_s} x_0^i - r_N\|_{(\Delta^{\mathfrak{D}}(N_s, N))^{-1}}^2 \\
& \text{subject to} && \|r_N + v_k^i\|_D^2 \leq d_2^2, k = 1, 2, \dots, m(N).
\end{aligned} \tag{P7}$$

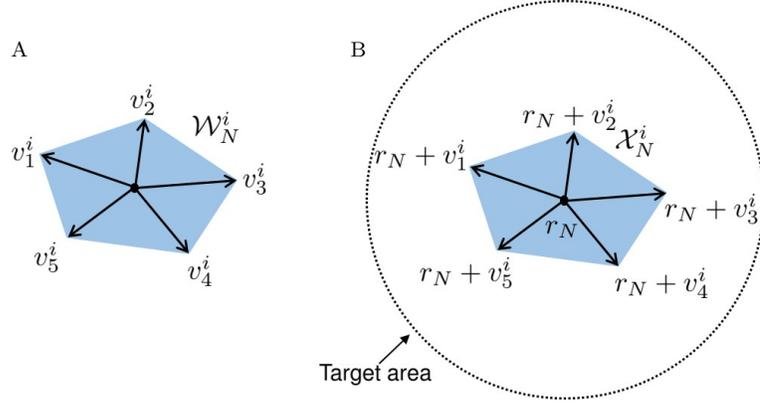


Figure 5.3: Geometric illustration of (5.24). (A) The vertices $\{v_1^i, v_2^i, v_3^i, v_4^i, v_5^i\}$ of \mathcal{W}_N^i are shown. (B) The set \mathcal{X}_N^i is placed inside the target area if all vertices $r_N + v_k^i, k = 1, 2, \dots, 5$, of \mathcal{X}_N^i are inside such area.

Now, we combine (P6) and (P7) to form a complete problem as follows.

$$\begin{aligned}
& \underset{r_{N_s}, r_s, r_0^i, r_N \in \mathbb{R}^4}{\text{minimize}} && \|x_0^o\|_{\Xi_1}^2 + 2r_{N_s}^T \Xi_2 x_0^o + \|r_{N_s}\|_{\Xi_3}^2 + \frac{1}{2} \|((A^{\mathfrak{D}})^{N-N_s} r_0^i - r_N)\|_{(\Delta^{\mathfrak{D}}(N_s, N))^{-1}}^2 \\
& \text{subject to} && (r_{N_s} + v_k^o - r_s)^T D r_s \geq 0, k = 1, 2, \dots, m(N_s), \\
& && \|r_s\|_D^2 = d_1^2, \\
& && G_{N_s}^o (r_0^i - r_{N_s}) \leq g_{N_s}^o, \\
& && \|r_N + v_k^i\|_D^2 \leq d_2^2, k = 1, 2, \dots, m(N).
\end{aligned} \tag{P8}$$

The cost function is the sum of the ones in (P6) and (P7). Notice that the inner stage initial state is denoted by r_0^i , which is an optimization variable. It is constrained to stay within the PSSS $\mathcal{X}_{N_s}^o$, as formulated in the third constraint. This constraint is added to find a realization of the initial state for the inner stage (which is also the switching state) which yields the minimum overall cost compared to other states inside the PSSS. The first two constraints come from (P6).

Problem (P8) seeks:

1. a deterministic switching state r_{N_s} such that $\mathcal{X}_{N_s}^o$ is placed outside the denied area;
2. an initial state r_0^i of the inner stage such that r_0^i is a realization of the switching state which resides in $\mathcal{X}_{N_s}^o$;
3. a deterministic terminal state r_N such that \mathcal{X}_N^i is placed inside the target area;
4. a lower bound of the overall cost among all realizations of the inner stage initial states.

Problem (P8) is a nonconvex QCQP, which we may obtain a numerical solution that yields a local minimum by a nonlinear solver. Let $(r_{N_s}^*, r_s^*, r_0^{i*}, \bar{r}_N)$ denote a local minimizer of (P8). For implementation, we apply a control sequence $\mathbf{u}_{0:N_s-1}^{o*}$ to the outer stage states, where

$$u_k^{o*} = -H_k^{\mathfrak{D}} x_k^* - L_k^{\mathfrak{D}} r_{N_s}^*, k = 0, 1, \dots, N_s - 1. \quad (5.25)$$

The state is steered to $x_{N_s}^* \in \mathcal{X}_{N_s}^{o*} \stackrel{\text{def}}{=} r_{N_s}^* \oplus \mathcal{W}_{N_s}^o$. Note that $x_{N_s}^*$ is may not be identical to $r_{N_s}^*$ due to perturbations. Then we set $x_{N_s}^* = x_0^i$ as the initial state of the inner stage and solve (P7) for an optimal deterministic terminal state, denoted by r_N^* . The inner stage optimal control sequence $\mathbf{u}_{N_s:N-1}^{i*}$ is applied to the mobile agent, where

$$u_k^{i*} = -(R^{\mathfrak{D}})^{-1} (B^{\mathfrak{D}})^T ((A^{\mathfrak{D}})^T)^{N-k-1} (\Delta^{\mathfrak{D}}(N_s, N))^{-1} ((A^{\mathfrak{D}})^{N-N_s} x_0^i - r_N^*), \quad (5.26)$$

for $k = N_s, N_s+1, \dots, N-1$. And the mobile agent will be steered to $x_N^* \in r_N^* \oplus \mathcal{W}_N^i$, which is inside the target area.

The control sequences $\mathbf{u}_{0:N_s-1}^{o*}$ and $\mathbf{u}_{N_s:N-1}^{i*}$ can carry the mobile agent to the target and guarantee a proper stage switching outside the denied area when (P8) is solvable. But the outer stage optimal control does not use feedback efficiently. It only use the measurement at the initial time to plan for the future control actions once and for all. Though u_k^{o*} has the feedback form, i.e., it is linear function of state x_k^* at time k , it is only a control action that is optimal when planned at the initial time. Ideally, we would like the measurements to aid the controller to decide a current action such that the future controls are optimal based on current measurements. We will propose a robust controller in the next section to stress this issue.

5.4 Robust controller using variable horizon model predictive control

In this section, we introduce a robust controller using model predictive control with a variable horizon. The controller only works in the outer stage. It plans the mobile agent's future trajectory at each step, seeking an optimal switching time and a sequence of control actions such that a cost is minimized. And it will implement the first control action in the optimal sequence to the actuator of the mobile agent. The model predictive controller in the outer stage is switched to an inner stage controller when the criterion for stage switching is met. Such criterion is determined by the variable horizon scheme. The inner stage controller remains (5.26), which has been introduced in the previous section.

Even though the switching time N_s is allowed to vary, we stick to the controls

in the forms of (5.25) in the outer stage. Hence, we adopt (P8) with necessary changes to accommodate the variable switching time.

We now introduce new notations under the variable switching time. We use subscript $k[N_s]|j$ to denote a variable at a future time k , which is originated from the current time j , with switching time N_s . We will omit $|j$ to indicate that such variable is independent to a current time j . And we use $N_s|j$ as a short for $N_s[N_s]|j$. For example, (5.8a) is now written as

$$u_{k[N_s]|j}^o = -H_{k[N_s]}^{\mathfrak{D}} x_{k[N_s]|j} - L_{k[N_s]}^{\mathfrak{D}} r_{N_s|j}, k = j, j + 1, \dots, N_s - 1, \quad (5.27)$$

where the initial value $x_{j[N_s]|j} = x_j$ is the current state and $r_{N_s|j}$ is the deterministic switching state to be determined later. The matrices $H_{k[N_s]}^{\mathfrak{D}}$ and $L_{k[N_s]}^{\mathfrak{D}}$ are

$$H_{k[N_s]}^{\mathfrak{D}} = H_{N_o - (N_s - k)}^{\mathfrak{D}}, \quad (5.28a)$$

$$L_{k[N_s]}^{\mathfrak{D}} = L_{N_o - (N_s - k)}^{\mathfrak{D}}, \quad (5.28b)$$

for $k = 0, 1, \dots, N_s - 1$, where the terms on the right hand side of both equations come from (5.4a)-(5.4g) with a sufficiently large boundary time N_o . The equalities in (5.28) hold because (5.4a)-(5.4e) are solved backwards in time. Hence, if (5.4a)-(5.4e) are solved for a sufficiently large N_o , then we can obtain the values of $H_{k[N_s]}^{\mathfrak{D}}$ and $L_{k[N_s]}^{\mathfrak{D}}$ by referring to the last $(N_s - k)^{\text{th}}$ values in the sequences $\mathbf{H}_{0:N_o}^{\mathfrak{D}}$ and $\mathbf{L}_{0:N_o}^{\mathfrak{D}}$, i.e., $H_{N_o - (N_s - k)}^{\mathfrak{D}}$ and $L_{N_o - (N_s - k)}^{\mathfrak{D}}$, respectively. This is another advantage of using the control in form of (5.8a) since equations (5.4a)-(5.4e) only need solving *once* and *offline* with a sufficiently large N_o , instead of being solved each time when N_s varies.

Remark 5.1. $H_{k[N_s-1]}^{\mathfrak{D}} = H_{k+1[N_s]}^{\mathfrak{D}}$ and $L_{k[N_s-1]}^{\mathfrak{D}} = L_{k+1[N_s]}^{\mathfrak{D}}$.

With (5.28), we have a new formula for the PSSS $\mathcal{X}_{N_s}^o$, which is now denoted by $\mathcal{X}_{N_s|j}^o$, with switching time N_s ,

$$\mathcal{X}_{N_s|j}^o = r_{N_s|j} \oplus \mathcal{W}_{N_s|j}^o, \quad (5.29)$$

$$\mathcal{W}_{N_s|j}^o = \bigoplus_{i=j}^{N_s-1} \bar{\Phi}_{[N_s]}^{\mathfrak{D}}(N_s, i+1)\mathcal{W}, \quad (5.30)$$

where

$$\bar{\Phi}_{[N_s]}^{\mathfrak{D}}(n_2, n_1) = \begin{cases} \prod_{k=n_1}^{n_2-1} (A^{\mathfrak{D}} - B^{\mathfrak{D}} H_{k[N_s]}^{\mathfrak{D}}), & \text{if } n_1 < n_2, \\ I_4, & \text{if } n_1 = n_2. \end{cases} \quad (5.31)$$

Since $\mathcal{W}_{N_s|j}^o$ is still a convex polytope, we denote its vertices by v_k^o , $k = 1, 2, \dots, m(N_s|j)$, i.e.,

$$\mathcal{W}_{N_s|j}^o = \text{convh}\{v_1^o, v_2^o, \dots, v_{m(N_s|j)}^o\}. \quad (5.32)$$

Before we proceed to the new problem formulation, we first introduce properties of the propagated set of perturbation $\mathcal{W}_{N_s|j}^o$. The criterion for stage switching relies on these properties.

Lemma 5.1. *For $j = 0, 1, \dots, N_s - 2$, we have*

1. $\mathcal{W}_{N_s|j+1}^o \subseteq \mathcal{W}_{N_s|j}^o$
2. $\mathcal{W}_{N_s|j+1}^o = \mathcal{W}_{N_s-1|j}^o$
3. $\mathcal{W}_{N_s-1|j}^o \subseteq \mathcal{W}_{N_s|j}^o$

Proof. 1. By definition (5.30)

$$\mathcal{W}_{N_s|j}^o = \mathcal{W}_{N_s|j+1}^o \oplus \bar{\Phi}_{[N_s]}^{\mathfrak{D}}(N_s, j+1)\mathcal{W}. \quad (5.33)$$

Because \mathcal{W} contains the origin, we conclude that $\mathcal{W}_{N_s|j+1}^o \subseteq \mathcal{W}_{N_s|j}^o$.

2. By definition,

$$\mathcal{W}_{N_s-1|j}^o = \bigoplus_{i=j}^{N_s-2} \bar{\Phi}_{[N_s-1]}^{\mathfrak{D}}(N_s-1, i+1)\mathcal{W} \quad (5.34)$$

$$= \mathcal{W} \oplus \bigoplus_{i=j}^{N_s-3} \prod_{k=i+1}^{N_s-2} (A^{\mathfrak{D}} - B^{\mathfrak{D}} H_{k[N_s-1]}^{\mathfrak{D}})\mathcal{W}. \quad (5.35)$$

On the other hand,

$$\mathcal{W}_{N_s|j+1}^o = \bigoplus_{i=j+1}^{N_s-1} \bar{\Phi}_{[N_s]}^{\mathfrak{D}}(N_s, i+1)\mathcal{W} \quad (5.36)$$

$$= \mathcal{W} \oplus \bigoplus_{i=j+1}^{N_s-2} \prod_{k=i+1}^{N_s-1} (A^{\mathfrak{D}} - B^{\mathfrak{D}} H_{k[N_s]}^{\mathfrak{D}})\mathcal{W} \quad (5.37)$$

$$= \mathcal{W} \oplus \bigoplus_{\bar{i}=j}^{N_s-3} \prod_{\bar{k}=\bar{i}+2}^{N_s-1} (A^{\mathfrak{D}} - B^{\mathfrak{D}} H_{\bar{k}[N_s]}^{\mathfrak{D}})\mathcal{W} \quad (5.38)$$

$$= \mathcal{W} \oplus \bigoplus_{\bar{i}=j}^{N_s-3} \prod_{\bar{k}=\bar{i}+1}^{N_s-2} (A^{\mathfrak{D}} - B^{\mathfrak{D}} H_{\bar{k}+1[N_s]}^{\mathfrak{D}})\mathcal{W} \quad (5.39)$$

$$= \mathcal{W}_{N_s-1|j}^o, \quad (5.40)$$

where we change variable $\bar{i} = i - 1$ on the third line and $\bar{k} = k - 1$ on the fourth line. The last step uses the fact that $H_{\bar{k}[N_s-1]}^{\mathfrak{D}} = H_{\bar{k}+1[N_s]}^{\mathfrak{D}}$, as shown in Remark 5.1.

3. This is a direct consequence of 1 and 2. \square

Lemma 5.1 states that if current time j is fixed, then the propagated set of perturbation $\mathcal{W}_{N_s|j}$ grows in size as N_s increases. The size of the propagated set of perturbation represents the level of uncertainty of the predicted switching state.

Now, we introduce the new problem formulation. Recall that we need to place

the PSSS $\mathcal{X}_{N_s|j}^o$ outside the denied area to guarantee a proper stage switching, i.e.,

$$(r_{N_s|j} + v_k^o - r_{s|j})^T D r_{s|j} \geq 0, k = 1, 2, \dots, m(N_s), \quad (5.41)$$

$$\|r_{s|j}\|_D^2 = d_1^2, \quad (5.42)$$

which rewrites (5.20) and (5.21) with new notations $r_{N_s|j}$ and $r_{s|j}$. But this is not enough because the PSSS could be placed far away from, while remaining outside, the denied area. It does not make sense to make the position of switching far away from the denied area. Hence it is reasonable to restrict the distance between the PSSS and the origin within a certain limit. This limit is selected as the Euclidean distance in the position subspace between the current stage x_j and the origin. Here, we use notation $\|\cdot\|_p$ to denote the Euclidean distance in the position subspace, i.e., for $x \in \mathbb{R}^4$,

$$\|x\|_p^2 = x^T \begin{bmatrix} I_2 & 0_{2 \times 2} \\ 0_{2 \times 2} & 0_{2 \times 2} \end{bmatrix} x. \quad (5.43)$$

Now, we restrict the distance to the origin of all vectors in the PSSS to be less than that of x_j , i.e.,

$$\|x_j\|_p^2 \geq \|r_{N_s|j} + v_k^o\|_p^2, k = 1, 2, \dots, m(N_s). \quad (5.44)$$

This constraint requires the position of any realization of the switching state $x_{N_s} \in \mathcal{X}_{N_s}^o = r_{N_s|j} \oplus \text{convh}\{v_1^o, v_2^o, \dots, v_{m(N_s|j)}^o\}$, subject to perturbations, is closer to the origin than the position of the current state.

Constraint (5.44) together with the constraints (5.41)-(5.42) form an annular region to place the PSSS $\mathcal{X}_{N_s}^o$, as shown in Figure 5.4. However, the size of PSSS grows over N_s because the size of PSSS is determined by the propagated set of

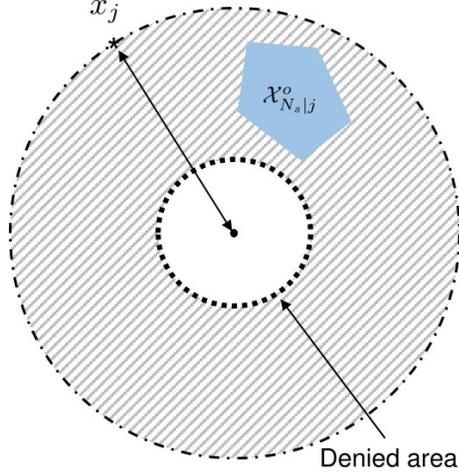


Figure 5.4: Annular region to place the PSSS $\mathcal{X}_{N_s}^o$ is the shaded area which is between the ball, whose radius is the distance from the origin to x_j in the position subspace, and the denied area.

perturbation $\mathcal{W}_{N_s}^o$. So there ought to be an upper bound $\bar{N}_s(j)$ on the switching time so that we can always find $r_{N_s|j}$ and $r_{s|j}$ satisfying (5.41)-(5.44) for some $N_s \leq \bar{N}_s(j)$.

Definition 5.1. Define the set of switching times $\mathbb{N}_s(j)$ as

$$\mathbb{N}_s(j) \stackrel{\text{def}}{=} \{N_s = j + 1, \dots, N_s^{\max} | \exists r_{N_s|j}, r_{s|j} \in \mathbb{R}^4 \text{ satisfying (5.41)-(5.44)}\},$$

where N_s^{\max} is the maximum allowed switching time that is given.

If $\mathbb{N}_s(j) \neq \emptyset$, then the upper bound $\bar{N}_s(j)$ is defined as the maximum N_s in $\mathbb{N}_s(j)$, or, equivalently,

$$\bar{N}_s(j) \stackrel{\text{def}}{=} \min\{N_s \in \{j+1, j+2, \dots, N_s^{\max}\} | \nexists r_{N_s|j}, r_{s|j} \in \mathbb{R}^4 \text{ satisfying (5.41)-(5.44)}\} - 1. \tag{5.45}$$

The effectiveness of $\bar{N}_s(j)$ as an upper bound is shown by the following proposition.

Proposition 5.1. *Suppose $\mathbb{N}_s(j) \neq \emptyset$.*

1. *Let $\hat{N}_s \in \{j+1, j+2, \dots, N_s^{max}\}$ be such that $\nexists r_{\hat{N}_s|j}, r_{s|j} \in \mathbb{R}^4$ satisfying (5.41)-(5.44). Then $\forall N_s > \hat{N}_s$, $\nexists r_{N_s|j}, r_{s|j} \in \mathbb{R}^4$ satisfying (5.41)-(5.44).*
2. *$\forall N_s \leq \bar{N}_s(j)$, $\exists r_{N_s|j}, r_{s|j} \in \mathbb{R}^4$ satisfying (5.41)-(5.44).*

Proof. 1. We prove by contradiction. Let $N_s \geq \hat{N}_s + 1$ and assume $\exists r_{N_s|j}, r_{s|j} \in \mathbb{R}^4$ satisfying (5.41)-(5.44). Then $\forall a_1, a_2, \dots, a_{m(N_s)} \geq 0$, such that

$$\sum_{k=1}^{m(N_s)} a_k = 1, \quad (5.46)$$

we have

$$0 \geq \sum_{k=1}^{m(N_s)} a_k \|r_{N_s|j} + v_k^o\|_p^2 - a_k \|x_j\|_p^2 \quad (5.47)$$

$$\geq \left\| \sum_{k=1}^{m(N_s)} a_k r_{N_s|j} + a_k v_k^o \right\|_p^2 - \|x_j\|_p^2 \quad (5.48)$$

$$\Rightarrow \|x_j\|_p^2 \geq \|r_{N_s|j} + \sum_{k=1}^{m(N_s)} a_k v_k^o\|_p^2, \quad (5.49)$$

i.e., for an arbitrary vector $v = \sum_{k=1}^{m(N_s)} a_k v_k^o \in \mathcal{W}_{N_s|j}^o$,

$$\|x_j\|_p^2 \geq \|r_{N_s|j} + v\|_p^2. \quad (5.50)$$

By Lemma 5.1, $\mathcal{W}_{\hat{N}_s|j} \subseteq \mathcal{W}_{N_s|j}$, i.e., $v_k^o \in \mathcal{W}_{N_s|j}, \forall k = 1, 2, \dots, m(\hat{N}_s)$, where v_k^o 's are the vertices of $\mathcal{W}_{\hat{N}_s|j}$, so we have

$$\|x_j\|_p^2 \geq \|r_{N_s|j} + v_k^o\|_p^2, \hat{k} = 1, 2, \dots, m(\hat{N}_s). \quad (5.51)$$

Then, by making $r_{\hat{N}_s|j} = r_{N_s|j}$, we have (5.44) satisfied, i.e.,

$$\|x_j\|_p^2 \geq \|r_{\hat{N}_s|j} + v_k^o\|_p^2, \hat{k} = 1, 2, \dots, m(\hat{N}_s). \quad (5.52)$$

On the other hand, since $\mathcal{W}_{\hat{N}_s|j}^o \subseteq \mathcal{W}_{N_s|j}^o$, each vertex v_k^o of $\mathcal{W}_{\hat{N}_s|j}^o$ can be represented by a convex combination of vertices v_k^o 's of $\mathcal{W}_{N_s|j}^o$, i.e., for each v_k^o ,

there exist $b_1, \dots, b_{m(N_s|j)} \geq 0$, $\sum_{k=1}^{m(N_s|j)} b_k = 1$ such that

$$v_k^o = \sum_{k=1}^{m(N_s|j)} b_k v_k^o. \quad (5.53)$$

Then by (5.41), for any $\hat{k} \in \{1, 2, \dots, m(\hat{N}_s|j)\}$,

$$(r_{N_s|j} + v_k^o - r_{s|j})^T D r_{s|j} \geq 0 \quad (5.54)$$

$$\Rightarrow \sum_{k=1}^{m(N_s|j)} b_k (r_{N_s|j} + v_k^o - r_{s|j})^T D r_{s|j} \geq 0 \quad (5.55)$$

$$\Rightarrow (r_{N_s|j} + v_{\hat{k}}^o - r_{s|j})^T D r_{s|j} \geq 0. \quad (5.56)$$

By making $r_{\hat{N}_s|j} = r_{N_s|j}$, $r_{\hat{s}|j} = r_{s|j}$, we know (5.41)-(5.42) hold for \hat{N}_s , i.e.,

$$(r_{\hat{N}_s|j} + v_{\hat{k}}^o - r_{\hat{s}|j})^T D r_{\hat{s}|j} \geq 0, \hat{k} = 1, 2, \dots, m(\hat{N}_s), \quad (5.57)$$

$$\|r_{\hat{s}|j}\|_D^2 = d_1^2, \quad (5.58)$$

which is a contradiction.

2. This is a direct consequence of 1. By contradiction, assume $\exists N'_s \leq \bar{N}_s(j)$ such that $\nexists r_{N'_s|j}, r_{s'|j} \in \mathbb{R}^4$ satisfying (5.41)-(5.44). Then, by 1, $\forall N_s \geq N'_s$, $\nexists r_{N_s|j}, r_{s|j} \in \mathbb{R}^4$ satisfying (5.41)-(5.44). This is a contradiction since $\bar{N}_s(j) \geq N'_s$.

□

If $\mathbb{N}_s(j) = \emptyset$, then either $j = N_s^{\max}$ or there exists no $r_{N_s|j}, r_{s|j} \in \mathbb{R}^4$ satisfying (5.41)-(5.44) for $N_s \in \{j+1, \dots, N_s^{\max}\}$. Intuitively, the former means that the

current time j reaches the maximum allowed time for stage switching while the latter means that there is no chance we can place the PSSS within the annular region specified by (5.41)-(5.44). In both cases, the model predictive controller terminates and the inner stage controller takes over.

Therefore, the emptiness of $\mathbb{N}_s(j)$ characterizes the criterion of stage switching, which we summarize in the following definition.

Definition 5.2. *The criterion for stage switching is when there is no feasible future switching time, i.e., when $\mathbb{N}_s(j) = \emptyset$. The outer stage shall terminate at the current step j and switch to the inner stage.*

Now we present modified outer stage problem.

$$\begin{aligned}
& \underset{r_{N_s|j}, r_{s|j}, r_{0|j}^i, r_{N|j} \in \mathbb{R}^4, N_s \in \mathbb{R}}{\text{minimize}} && \|x_j\|_{\Xi_1}^2 + 2r_{N_s|j}^T \Xi_2 x_j + \|r_{N_s|j}\|_{\Xi_3}^2 \\
& && + \frac{1}{2} \|(A^{\mathfrak{D}})^{N-N_s} r_{0|j}^i - r_{N|j}\|_{(\Delta^{\mathfrak{D}}(N_s, N))^{-1}}^2 \\
& \text{subject to} && (r_{N_s|j} + v_k^o - r_{s|j})^T D r_{s|j} \geq 0, k = 1, 2, \dots, m(N_s|j), \\
& && \|r_{s|j}\|_D^2 = d_1^2, \\
& && \|x_j\|_p^2 \geq \|r_{N_s|j} + v_k^o\|_p^2, k = 1, 2, \dots, m(N_s|j), \\
& && G_{N_s}^o(r_{0|j}^i - r_{N_s|j}) \leq g_{N_s}^o, \\
& && \|r_{N|j} + v_k^i\|_D^2 \leq d_2^2, k = 1, 2, \dots, m(N), \\
& && N_s \in \{j+1, \dots, N_s^{\max}\}.
\end{aligned} \tag{P-MPC}$$

We do not explicitly require $N_s \in \mathbb{N}_s(j)$ because it holds automatically when (P-MPC) is feasible.

If (P-MPC) is feasible, with N_s^* denoting the optimal switching time and $r_{N_s^*|j}^*$ denoting a solution of (P-MPC), we apply the first control action of sequence,

$$u_{k[N_s^*|j]}^{o*} = -H_{k[N_s^*|j]}^{\mathfrak{D}} x_{k[N_s^*|j]}^* - L_{k[N_s^*|j]}^{\mathfrak{D}} r_{N_s^*|j}^*, \quad (5.59)$$

to the actuator of the mobile agent.

If (P-MPC) is infeasible for some j , then the criterion for stage switching is met and the outer stage terminates. The inner stage starts with initial state being x_j and the optimal inner stage control is given by (5.26), with $x_0^i = x_j$ and r_N^* solved in (P7).

Problem (P-MPC) is a mixed-integer programming because of the last constraint. We can solve (P-MPC) by solving at most $N_s^{\max} - j$ subproblems, i.e.,

$$\begin{aligned} & \underset{r_{N_s|j}, r_{s|j}, r_{0|j}^i, r_{N|j} \in \mathbb{R}^4}{\text{minimize}} && \|x_j\|_{\Xi_1}^2 + 2r_{N_s|j}^T \Xi_2 x_j + \|r_{N_s|j}\|_{\Xi_3}^2 \\ & && + \frac{1}{2} \|(A^{\mathfrak{D}})^{N-N_s} r_{0|j}^i - r_{N|j}\|_{(\Delta^{\mathfrak{D}}(N_s, N))^{-1}}^2 \\ & \text{subject to} && (r_{N_s|j} + v_k^o - r_{s|j})^T D r_{s|j} \geq 0, k = 1, 2, \dots, m(N_s|j), \\ & && \|r_{s|j}\|_D^2 = d_1^2, \\ & && \|x_j\|_p^2 \geq \|r_{N_s|j} + v_k^o\|_p^2, k = 1, 2, \dots, m(N_s|j), \\ & && G_{N_s}^o(r_{0|j}^i - r_{N_s|j}) \leq g_{N_s}^o, \\ & && \|r_{N|j} + v_k^i\|_D^2 \leq d_2^2, k = 1, 2, \dots, m(N). \end{aligned} \quad (\text{P-MPC}(N_s))$$

where N_s is fixed. We do not necessarily need to solve (P-MPC(N_s)) for all $N_s \in \{j+1, \dots, N_s^{\max}\}$. Rather, we only need to solve (P-MPC(N_s)) in an ascending order of N_s : $j+1 \rightarrow j+2 \rightarrow \dots \rightarrow N_s^{\max}$. Following this order, if (P-MPC(N_s))

is non-feasible for some $N_s = \tilde{N}_s$, then, by Proposition 5.1, there is no need to carry on computing the rest of the subproblems ($\text{P-MPC}(N_s)$) in which $N_s \in \{\tilde{N}_s + 1, \dots, N_s^{\max}\}$. Let $J_{\text{MPC}}(N_s) \stackrel{\text{def}}{=} \text{val}(\text{P-MPC}(N_s))$. Then we can obtain the optimal switching time N_s^* such that

$$N_s^* = \arg \min_{N_s \in \{j+1, \dots, \tilde{N}_s-1\}} J_{\text{MPC}}(N_s). \quad (5.60)$$

5.5 Simulation result

In this section, we show the simulation result of the variable horizon model predictive controller in the previous section.

We use the following parameters for simulation.

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, D = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (5.61a)$$

$$Q = I_4, R = I_2, d_1 = 4, d_2 = 2, x_0 = \begin{bmatrix} 10 & 0 & 2 & -5 \end{bmatrix}^T. \quad (5.61b)$$

The set of perturbation \mathcal{W} is characterized by the inequality

$$\mathcal{W} = \{x \in \mathbb{R}^4 | Gx \leq g\}, \quad (5.62)$$

where

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 \end{bmatrix}, g = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix}, \quad (5.63)$$

i.e., \mathcal{W} is a 4-dimensional cube with side length 0.2.

We use a sample time $T_s = 0.1$ s. And the values of $A^{\mathfrak{D}}, B^{\mathfrak{D}}$ follow (5.2) and $R^{\mathfrak{D}} = T_s R$ and $Q^{\mathfrak{D}} = T_s Q$. And we set the maximum allowed time for switching $N_s^{\max} = 30$, and the terminal time $N = N_s + 5$, i.e., the duration for the inner stage is 5 sample times.

We solve (P-MPC(N_s)) by MATLAB fmincon using the YALMIP [28]. At each step j , we solve (P-MPC(N_s)) for $N_s \in \{j + 1, \dots, N_s^{\max}\}$ in an ascending order, until (P-MPC(N_s)) is infeasible. Then we select the optimal switching time N_s^* using (5.60). Define a successful trial in which the mobile agent switches stage at a position that is sufficiently close to outside the denied area, and reaches the target eventually. Then the trajectory of a successful trial is shown in Figure 5.5. The initial position and trajectories planned at each step are displayed in Figure 5.6. The optimal switching time and optimal cost at each step are displayed in Figure 5.7.

A collection of 16 successful trials is shown in Figure 5.8. As can be seen, all

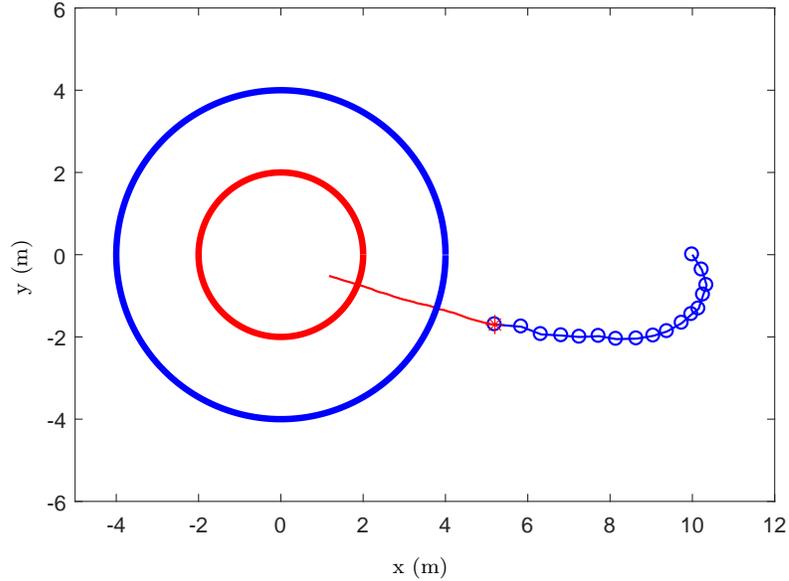


Figure 5.5: Trajectory of a successful trial. The target area (marked by the red circle) resides inside the denied area (marked by the big blue circle). Each small blue circle corresponds to the position at each step in the outer stage. The inner stage starts at the red asterisk and steers the mobile agent to the target.

such trajectories switch stage outside the denied area and reach the target eventually.

However, some trials ended in undesirable trajectories due to solver incapacibilities. As the criteria for stage switching is the infeasibility of $(\text{P-MPC}(N_s))$, if the solver claims the problem is infeasible, then the controller will switch, regardless of the actual feasibility of the problem. An example is shown in Figure 5.9 for illustration. In this trial, the solver claims the problem is infeasible at $j = 5$ but an examination on the data indicates the contrary.

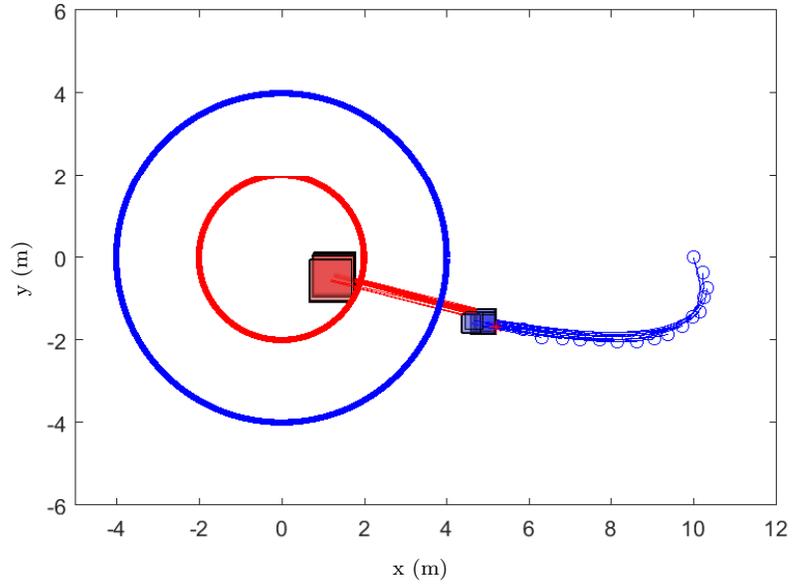


Figure 5.6: Collection of planned trajectories at all steps of a successful trial. The target area (marked by the red circle) resides inside the denied area (marked by the big blue circle). Each planned trajectory starts with the measured state at each step (marked by small blue circles). At the each step, the outer stage problem is solved where the PSSS $\mathcal{X}_{N_s^*}^o$ (marked by the blue rectangle) is placed outside the denied area and the PSTS \mathcal{X}_N^i (marked by the red rectangle) is placed inside the target area. The blue and red curves correspond to the predicted trajectories under closed-loop control (5.59) and openloop control (5.26), respectively. The controller switches stage at the position marked by the red asterisk.

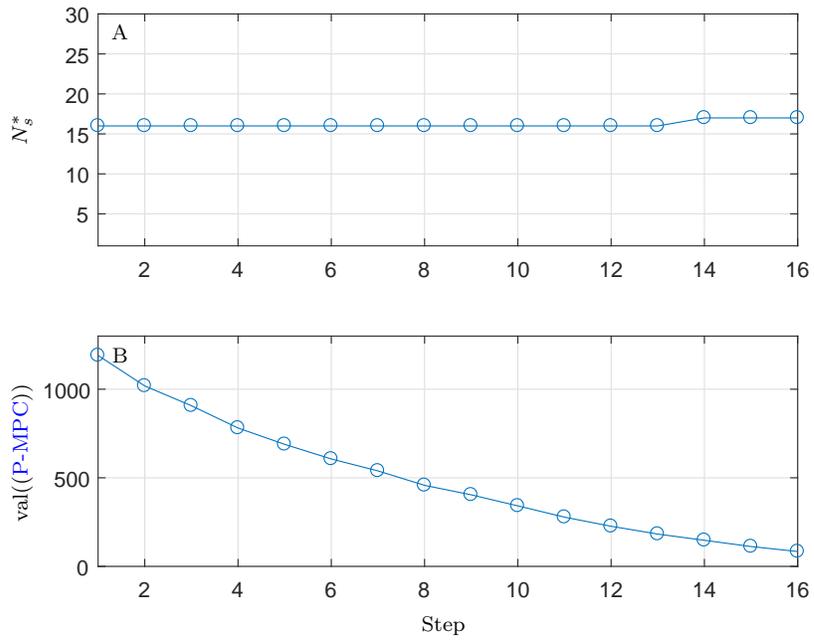


Figure 5.7: Optimal switching time N_s^* and optimal cost of (P-MPC) at each step.

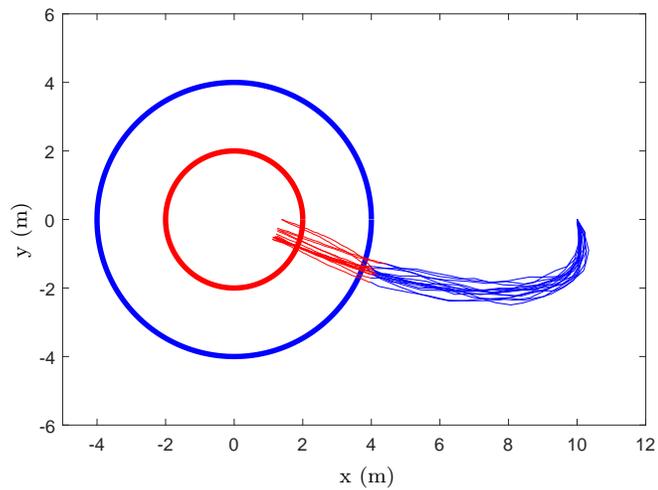


Figure 5.8: Collection of 16 successful trials. All trials have switching positions outside the denied area and have reached the target.

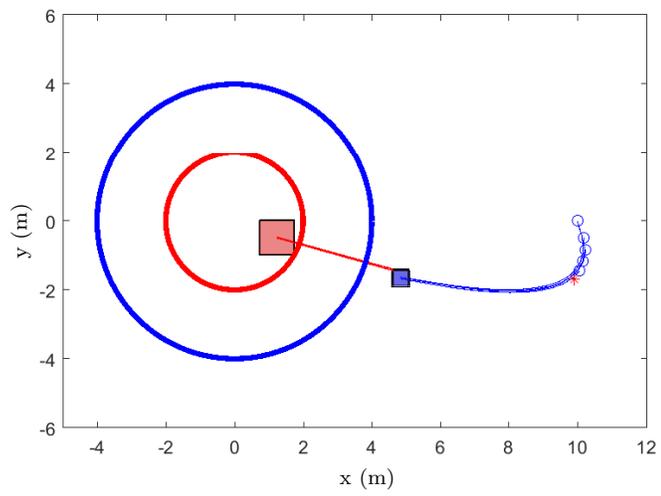


Figure 5.9: Collection of planned trajectories at all steps of an unsuccessful trial. The stage switching happens at the position marked by the red asterisk, which is far from the denied area.

Chapter 6: Quadrotor Experiment

6.1 Introduction

Quadrotor is a type of aerial vehicle that has four rotors providing thrusts and torques for movement of 6 degree-of-freedom (DOF). The simple mechanical structure of the quadrotor makes it an ideal testbed for control algorithms. Though the quadrotor has nonlinear dynamics, a linear stabilizing controller can be built based on the linearized model in hover state. Hence, various quadrotor platforms and controllers have been designed and built for different tasks. Since a considerable amount of research has been done on the control of the quadrotor, we only name a few here. Early studies on mathematical modeling and aerodynamic effects can be found in [29, 30], followed by research on trajectory generation and control [31–35]. And beyond the control of a single quadrotor, swarm control has been investigated in [36, 37].

In this chapter, we will introduce the quadrotor testbeds used in the CPS and Cooperative Autonomy Laboratory: an AscTec Hummingbird and an Ar.Drone. We establish the dynamical model and design controllers for each quadrotor. We also show the theoretical study in Chapter 4 and Chapter 5 can be utilized to build a controller that can steer a quadrotor to the target which is enclosed within a denied

area.

6.2 AscTec Hummingbird

The AscTec Hummingbird is a small, lightweight, and agile quadrotor built by Ascending Technologies. It consists of a carbon-fiber frame, four brushless motors, a lithium polymer (LiPo) battery, two ARM7 processors, a GPS receiving module, an RC receiver, a pair of XBee modules, and inertial sensors. The inertial sensors include a 3-DOF accelerometer, a 3-DOF gyroscope, a pressure sensor, and a compass.

The AscTec Hummingbird provides an onboard Autopilot, which is functioned on a High Level processor (HLP) and a Low Level processor (LLP). The LLP handles sensor data processing, data fusion as well as a built-in fast and stable attitude control algorithm with an update rate of 1 kHz. The HLP is free for custom C code which enables customized functions. More importantly, the HLP provides protections while testing custom code as the pilot can always switch back to the stable controller on LLP to recover from critical flight situations [38].

Ascending Technologies provides free software development kit (SDK) for users to program the HLP. The SDK is a C code framework in an Eclipse environment with cross-compiler and debugger [39]. It also includes the AscTec Communication Interface (ACI) for communications between the AscTec Hummingbird and a local machine [40]. It enables users to request variables, send commands, and set parameters easily, through a pair of XBee modules.

The SDK also provides the AscTec Simulink Toolkit which works in combination with the SDK. It enables users to develop control systems in Simulink and automatically generate C code which can be flashed to the HLP. The toolkit provides a model for communication, which sends live commands from ground PC and monitor custom debug data from the Hummingbird [41].

6.2.1 Dynamics and mathematical model

The coordinate systems are shown in Figure 6.1. The world frame, \mathcal{W} , is defined by axes $(x_{\mathcal{W}}, y_{\mathcal{W}}, z_{\mathcal{W}})$ with $z_{\mathcal{W}}$ pointing downward. The body frame, \mathcal{B} , is attached to the center of mass of the quadrotor where $x_{\mathcal{B}}$ coincides with the desired forward direction, $y_{\mathcal{B}}$ is in the plane of motors and perpendicular to $x_{\mathcal{B}}$, and $z_{\mathcal{B}}$ follows the right-hand rule and is pointing vertically down during perfect hovering.

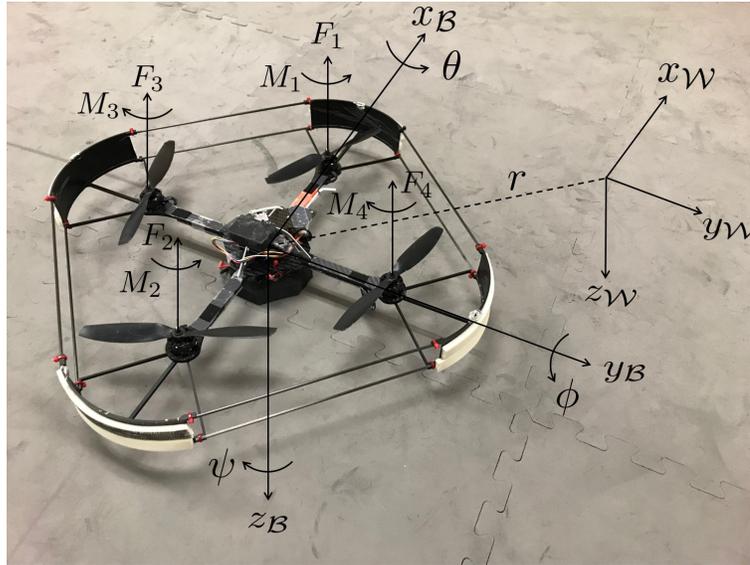


Figure 6.1: Coordinate systems and forces/torques acting on the AscTec Hummingbird frame

Newton-Euler formulation of the quadrotor model in this thesis follows [42].

The Newton equation in the world frame is

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} + {}^{\mathcal{W}}R_{\mathcal{B}} \begin{bmatrix} 0 \\ 0 \\ -\sum_{i=1}^4 F_i \end{bmatrix}, \quad (6.1)$$

where $r^T = \begin{bmatrix} x & y & z \end{bmatrix}$ is the vector of position of the quadrotor in \mathcal{W} , m is the quadrotor mass, g is the gravitational acceleration, F_i is the thrust generated by rotor i along $-z_{\mathcal{B}}$ axis, and ${}^{\mathcal{W}}R_{\mathcal{B}}$ is the rotation matrix from \mathcal{B} to \mathcal{W} . The rotation is derived from the $X - Y - Z$ Euler angle formulation [43] as the following

$$\begin{aligned} {}^{\mathcal{W}}R_{\mathcal{B}} &= R_z(\psi)R_y(\phi)R_x(\theta) \\ &= \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\phi & 0 & s_\phi \\ 0 & 1 & 0 \\ -s_\phi & 0 & c_\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\theta & -s_\theta \\ 0 & s_\theta & c_\theta \end{bmatrix} \\ &= \begin{bmatrix} c_\psi c_\phi & c_\psi s_\phi s_\theta - s_\psi c_\theta & c_\psi s_\phi c_\theta + s_\psi s_\theta \\ s_\psi c_\phi & s_\psi s_\phi s_\theta + c_\psi c_\theta & s_\psi s_\phi c_\theta - c_\psi s_\theta \\ -s_\phi & c_\phi s_\theta & c_\phi c_\theta \end{bmatrix}. \end{aligned} \quad (6.2)$$

The Euler angles θ , ϕ , and ψ correspond to the roll, pitch, and yaw, respectively.

The Euler equation in the body frame is

$$J\dot{\omega} = M - M_g - \omega \times J\omega, \quad (6.3)$$

where J is the inertial tensor of the quadrotor about $(x_{\mathcal{B}}, y_{\mathcal{B}}, z_{\mathcal{B}})$ axes, ω is the vector of angular velocities $(\omega_x, \omega_y, \omega_z)$ in \mathcal{B} , M is the vector of torques produced by rotors, M_g is the vector of gyroscopic torques.

Neither (6.1) nor (6.3) considers the effects of blade flapping. By assuming symmetry in weight distribution about $x_{\mathcal{B}}$ and $y_{\mathcal{B}}$ axes and low speed operation, we ignore the gyroscopic torques and consider the simplified terms

$$J = \begin{bmatrix} J_x & & \\ & J_y & \\ & & J_z \end{bmatrix}, M = \begin{bmatrix} l(F_3 - F_4) \\ l(F_1 - F_2) \\ -M_1 - M_2 + M_3 + M_4 \end{bmatrix}, \quad (6.4)$$

where J_x, J_y , and J_z are the quadrotor moments of inertia about $(x_{\mathcal{B}}, y_{\mathcal{B}}, z_{\mathcal{B}})$ axes, M_i correspond to the torques produced by rotor i in the body frame, and l is the distance from the center of the propeller to the center of mass of the quadrotor. The thrust and moments produced by rotors are modeled by a quadratic function of the rotor rotational speed, i.e.,

$$F_i = k_F \omega_i^2, M_i = k_M \omega_i^2, i = 1, 2, 3, 4, \quad (6.5)$$

where k_F and k_M are coefficients for thrust and torque, respectively, and ω_i is the rotational speed of rotor i .

By [44], the angular velocity ω has the following relation to the derivative of Euler angles

$$\begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s_{\theta} t_{\phi} & c_{\theta} t_{\phi} \\ 0 & c_{\theta} & -s_{\theta} \\ 0 & \frac{s_{\theta}}{c_{\phi}} & \frac{c_{\theta}}{c_{\phi}} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}. \quad (6.6)$$

Now, combining (6.1), (6.3), and (6.6), we arrive at the simplified nonlinear

model

$$\ddot{x} = -(c_\psi s_\phi c_\theta + s_\psi s_\theta)(F_1 + F_2 + F_3 + F_4) \frac{1}{m}, \quad (6.7a)$$

$$\ddot{y} = -(s_\psi s_\phi c_\theta - c_\psi s_\theta)(F_1 + F_2 + F_3 + F_4) \frac{1}{m}, \quad (6.7b)$$

$$\ddot{z} = g - c_\phi c_\theta (F_1 + F_2 + F_3 + F_4) \frac{1}{m}, \quad (6.7c)$$

$$\dot{\omega}_x = (\omega_y \omega_z (J_y - J_z) + l(F_3 - F_4)) \frac{1}{J_x}, \quad (6.7d)$$

$$\dot{\omega}_y = (\omega_x \omega_z (J_z - J_x) + l(F_1 - F_2)) \frac{1}{J_y}, \quad (6.7e)$$

$$\dot{\omega}_z = (-M_1 - M_2 + M_3 + M_4) \frac{1}{J_z}, \quad (6.7f)$$

$$\dot{\theta} = \omega_x + s_\theta t_\phi \omega_y + c_\theta t_\phi \omega_z, \quad (6.7g)$$

$$\dot{\phi} = c_\theta \omega_y - s_\theta \omega_z, \quad (6.7h)$$

$$\dot{\psi} = \frac{s_\theta}{c_\phi} \omega_y + \frac{c_\theta}{c_\phi} \omega_z. \quad (6.7i)$$

6.2.2 Simulink model

We establish a simulink model described by (6.7a)-(6.7i) as shown in Figure 6.3, with parameter values from [42],

$$m = 0.668 \text{ kg},$$

$$J_x = J_y = 0.0039 \text{ kg} \cdot \text{m}^2,$$

$$J_z = 0.0049 \text{ kg} \cdot \text{m}^2,$$

$$k_F = 6.11 \times 10^{-8} \text{ N} \cdot \text{rpm}^2,$$

$$k_M = 1.5 \times \text{N} \cdot \text{m/rpm}^2.$$

The model has rotor speed $\omega_i, i = 1, 2, 3, 4$, as inputs and $(x, y, z, \theta, \phi, \psi)$ as

well as the time derivative of each as outputs. This Simulink model is a useful tool for controller design and verification for future research.

6.2.3 Quadrotor controller

The quadrotor is controlled by nested feedback loops, similar to those in [32], as shown in Figure 6.2.

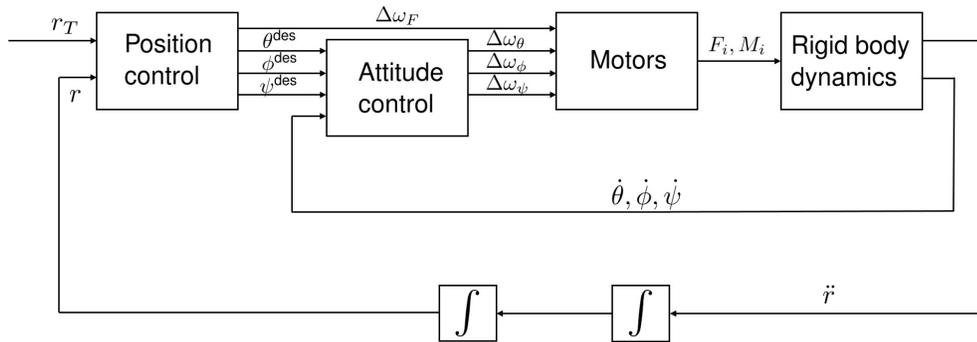


Figure 6.2: The nested control loops for position and attitude control.

The attitude controller of the inner loop uses onboard accelerometer and gyroscope to control the roll, pitch, and yaw and runs approximately at 1kHz [38]. The position controller in the outer loop uses the position measurements (by the Optitrack motion capture systems) and velocity estimation to control the quadrotor to follow trajectories.

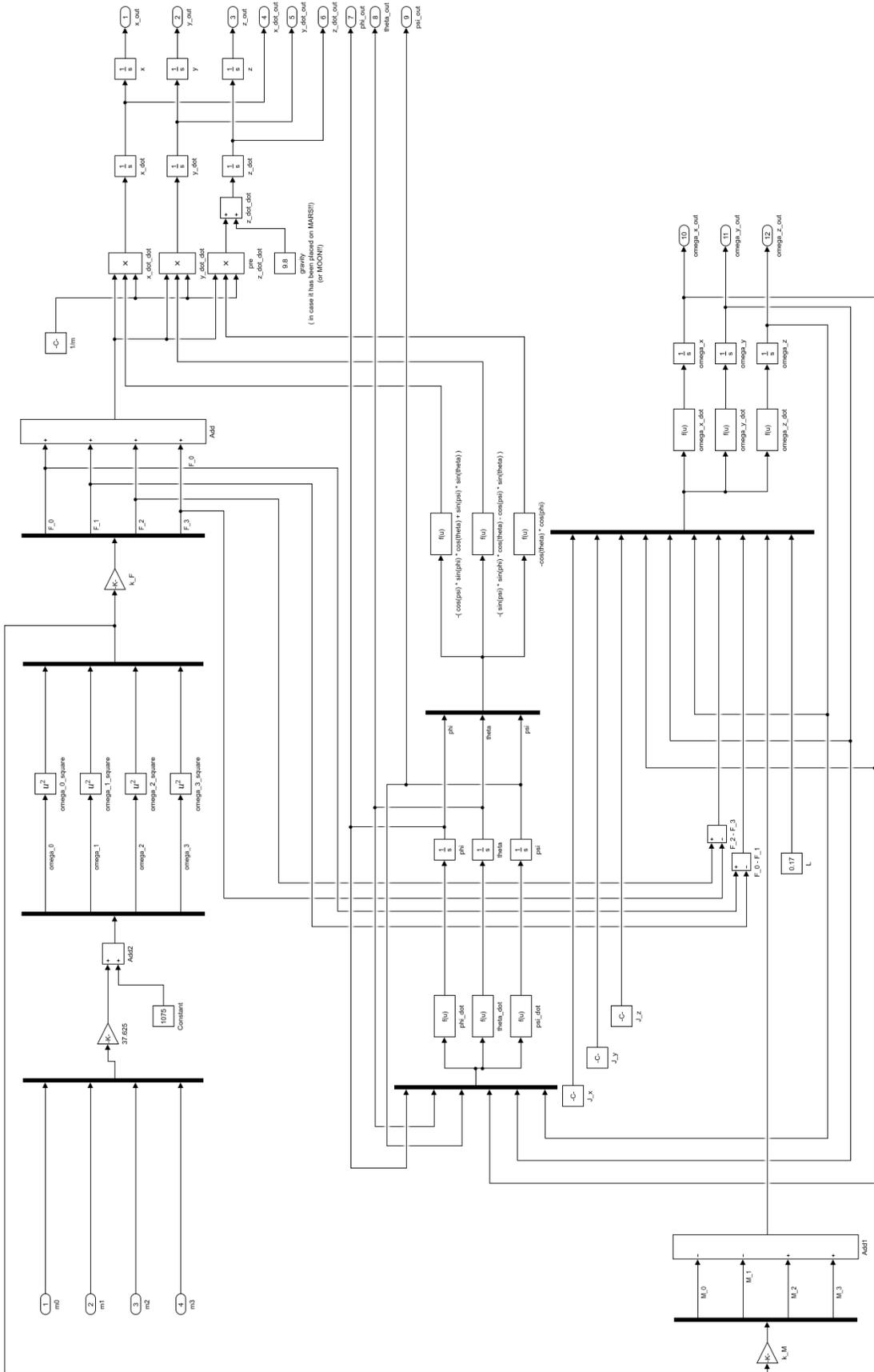


Figure 6.3: Simulink diagram of the nonlinear dynamics (6.7).

Our controllers are derived by linearizing the dynamics (6.7) at an operating point that corresponds to the nominal hover state, $r = r_0, \theta \approx 0, \phi \approx 0, \psi = \psi_0$, and $\dot{\theta} = \dot{\phi} = 0$, where r_0 and ψ_0 are given and roll as well as pitch are small such that $c_\phi \approx 1, s_\theta \approx 1, s_\phi \approx \phi, s_\theta \approx \theta$.

At the hover state, the nominal thrust from each propeller must satisfy

$$F_h = \frac{mg}{4}, \quad (6.8)$$

which specifies the nominal rotor speed as

$$\omega_h = \sqrt{\frac{mg}{4k_F}}. \quad (6.9)$$

6.2.3.1 Attitude control

The attitude controller aims to have the Euler angles (θ, ϕ, ψ) follow the desired values $(\theta^{\text{des}}, \phi^{\text{des}}, \psi^{\text{des}})$. This is achieved by controlling the speeds of each rotor such that the collective torques produced about each axes rotate the quadrotor into desired attitude.

Combining (6.7d)-(6.7f) with (6.5), we have the relation between rotor speeds ω_i 's and angular velocities ω_x, ω_y , and ω_z ,

$$\dot{\omega}_x = \frac{lk_F(\omega_3^2 - \omega_4^2) + \omega_y\omega_z(J_y - J_z)}{J_x}, \quad (6.10a)$$

$$\dot{\omega}_y = \frac{lk_F(\omega_1^2 - \omega_2^2) + \omega_x\omega_z(J_z - J_x)}{J_y}, \quad (6.10b)$$

$$\dot{\omega}_z = \frac{-k_M(-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2) + \omega_x\omega_y(J_x - J_y)}{J_z}. \quad (6.10c)$$

We assume $J_x = J_y$ by symmetry and assume the angular velocity ω_z is small, such that the products involving ω_z in (6.10) are small compared to other terms.

The desired rotor speed, denoted by $\omega_i^{\text{des}}, i = 1, 2, 3, 4$, has the following relation to the deviations from the hover state, denoted by $\Delta\omega_F, \Delta\omega_\theta, \Delta\omega_\phi$, and $\Delta\omega_\psi$,

$$\begin{bmatrix} \omega_1^{\text{des}} \\ \omega_2^{\text{des}} \\ \omega_3^{\text{des}} \\ \omega_4^{\text{des}} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & -1 \\ 1 & -1 & 0 & -1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \omega_h + \Delta\omega_F \\ \Delta\omega_\theta \\ \Delta\omega_\phi \\ \Delta\omega_\psi \end{bmatrix}, \quad (6.11)$$

where the nominal rotor speed required for the hover state is ω_h . The deviation $\Delta\omega_F$ results from the net force along the z_B axis, which will be introduced in the position controller. The rest of the deviations, $(\Delta\omega_\theta, \Delta\omega_\phi, \Delta\omega_\psi)$, cause nonzero net torques on the quadrotor frame, which further cause attitude changes in roll, pitch, and yaw, respectively. We use a proportional-derivative (PD) control law to generate the deviations, i.e.,

$$\Delta\omega_\theta = k_{p,\theta}(\theta^{\text{des}} - \theta) + k_{d,\theta}(\omega_y^{\text{des}} - \omega_y), \quad (6.12a)$$

$$\Delta\omega_\phi = k_{p,\phi}(\phi^{\text{des}} - \phi) + k_{d,\phi}(\omega_x^{\text{des}} - \omega_x), \quad (6.12b)$$

$$\Delta\omega_\psi = k_{p,\psi}(\psi^{\text{des}} - \psi) + k_{d,\psi}(\omega_z^{\text{des}} - \omega_z), \quad (6.12c)$$

where we take $\omega_x^{\text{des}} = \omega_y^{\text{des}} = \omega_z^{\text{des}} = 0$ for the hover state and use the approximation $\dot{\theta} \approx \omega_y$, $\dot{\phi} \approx \omega_x$, and $\dot{\psi} \approx \omega_z$.

Substitute (6.12) into (6.11) yields the desired rotor speeds.

6.2.3.2 Position control

We use pitch and roll to control the horizontal position in $x_{\mathcal{W}}$ and $y_{\mathcal{W}}$, $\Delta\omega_\psi$ to control yaw, and $\Delta\omega_F$ to control vertical position. Let $r_T(t)$ and $\psi_T(t)$ denote the trajectory and yaw, respectively, for the quadrotor to follow. Note $\psi_T(t) = \psi_T$ and $r_T(t) = r_T$ for the hovering controller.

We linearize (6.1) about the hover state to get the relation between the desired accelerations and desired roll, pitch, as well as deviation $\Delta\omega_F$, i.e.,

$$\ddot{x}^{\text{des}} = -g(c_{\psi^{\text{des}}}\phi^{\text{des}} + s_{\psi^{\text{des}}}\theta^{\text{des}}), \quad (6.13a)$$

$$\ddot{y}^{\text{des}} = -g(s_{\psi^{\text{des}}}\phi^{\text{des}} - c_{\psi^{\text{des}}}\theta^{\text{des}}), \quad (6.13b)$$

$$\ddot{z}^{\text{des}} = -\frac{8k_F\omega_h}{m}\Delta\omega_F. \quad (6.13c)$$

We can invert (6.13) and compute the desired roll θ^{des} , desired pitch ϕ^{des} , and deviation $\Delta\omega_F$ for the attitude controller, from the desired accelerations \ddot{x}^{des} , \ddot{y}^{des} , and \ddot{z}^{des} ,

$$\theta^{\text{des}} = -\frac{1}{g}(\ddot{x}^{\text{des}}s_{\psi^{\text{des}}} - \ddot{y}^{\text{des}}c_{\psi^{\text{des}}}), \quad (6.14a)$$

$$\phi^{\text{des}} = -\frac{1}{g}(\ddot{y}^{\text{des}}s_{\psi^{\text{des}}} + \ddot{x}^{\text{des}}c_{\psi^{\text{des}}}), \quad (6.14b)$$

$$\Delta\omega_F = -\frac{m}{8k_F\omega_h}\ddot{z}^{\text{des}}. \quad (6.14c)$$

Now, the desired accelerations are computed through a PD controller

$$\ddot{r}_{i,T} - \ddot{r}_i^{\text{des}} + k_{d,i}(\dot{r}_{i,T} - \dot{r}_i) + k_{p,i}(r_{i,T} - r_i) = 0, \quad i = 1, 2, 3, \quad (6.15)$$

where for the hover controller, $\ddot{r}_{i,T} = \dot{r}_{i,T} = 0$.

6.2.3.3 Luenberger observer

The position measurement of the quadrotor is provided by the Optitrack motion capture systems at 100 Hz. Ideally, we can compute the difference between consecutive position measurements to estimate the velocity of the quadrotor. But due to packet loss in the communication between PC and Optitrack, such velocity estimation contains frequent and unpredictable spikes. Hence, by (6.14) and (6.15), the spikes pass through desired angles θ^{des} and ϕ^{des} and further to desired rotor speed ω_i^{des} , which eventually cause observable shaking in the motion of the quadrotor.

We can build a Luenberger observer to estimate the velocity from other measurable quantities, since the dynamics of the quadrotor is known and the linearized model is valid near the hovering state. Consider the following linearized model near the hover state,

$$\ddot{x} = -g\phi, \quad (6.16a)$$

$$\ddot{y} = g\theta, \quad (6.16b)$$

$$\ddot{z} = -\frac{k_F}{m}(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) + g, \quad (6.16c)$$

$$\ddot{\theta} = \frac{lk_F}{J_x}(\omega_3^2 - \omega_4^2), \quad (6.16d)$$

$$\ddot{\phi} = \frac{lk_F}{J_y}(\omega_1^2 - \omega_2^2), \quad (6.16e)$$

$$\ddot{\psi} = \frac{k_M}{J_z}(-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2). \quad (6.16f)$$

We can write (6.16) as a linear time-invariant state space model. The state variable, denoted by s , contains the positions (x, y, z) , Euler angles (θ, ϕ, ψ) and

their derivatives, i.e.,

$$s^T = \begin{bmatrix} x & y & z & \theta & \phi & \psi & \dot{x} & \dot{y} & \dot{z} & \dot{\theta} & \dot{\phi} & \dot{\psi} \end{bmatrix}. \quad (6.17)$$

The control variable, denoted by u , is a vector of the squares of rotor speeds, i.e.,

$$u^T = \begin{bmatrix} \omega_1^2 & \omega_2^2 & \omega_3^2 & \omega_4^2 \end{bmatrix}. \text{ The linearized dynamics (6.16) have the following form}$$

$$\dot{s}(t) = As(t) + Bu(t) + \bar{g}, \quad (6.18)$$

where

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{k_F}{m} & -\frac{k_F}{m} & -\frac{k_F}{m} & -\frac{k_F}{m} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{lk_F}{J_x} & -\frac{lk_F}{J_x} \\ \frac{lk_F}{J_y} & -\frac{lk_F}{J_y} & 0 & 0 \\ \frac{k_M}{J_z} & \frac{k_M}{J_z} & -\frac{k_M}{J_z} & -\frac{k_M}{J_z} \end{bmatrix}, \quad (6.19)$$

and

$$\bar{g}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.20)$$

is the constant bias induced by gravitational acceleration.

The output, denoted by q , contains the positions (x, y, z) , and Euler angle (θ, ϕ, ψ) , i.e.,

$$q = Cs, \quad (6.21)$$

where

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (6.22)$$

Discretize (6.18) to obtain the discrete-time system matrices $A^{\mathfrak{D}}, B^{\mathfrak{D}}$ and $\bar{g}^{\mathfrak{D}}$

with sampling time T_s , i.e.,

$$\begin{bmatrix} A^{\mathfrak{D}} & B^{\mathfrak{D}} \\ * & I_4 \end{bmatrix} = \exp \left(\begin{bmatrix} A & B \\ 0_{4 \times 12} & 0_{4 \times 4} \end{bmatrix} T_s \right), \quad \begin{bmatrix} * & \bar{g}^{\mathfrak{D}} \\ * & 1 \end{bmatrix} = \exp \left(\begin{bmatrix} A & \bar{g} \\ 0_{1 \times 12} & 0 \end{bmatrix} T_s \right), \quad (6.23)$$

We now stick to the discrete-time state space model

$$\begin{cases} s_{k+1} = A^{\mathfrak{D}} s_k + B^{\mathfrak{D}} u_k + \bar{g}^{\mathfrak{D}} \\ q_k = C s_k \end{cases}, k = 0, 1, \dots \quad (6.24)$$

The Luenberger observer has the following form

$$\begin{cases} \hat{s}_{k+1} = A^{\mathfrak{D}} \hat{s}_k + B^{\mathfrak{D}} u_k + L(q_k - C \hat{s}_k) + \bar{g}^{\mathfrak{D}} \\ \hat{q}_k = C \hat{s}_k \end{cases}, k = 0, 1, \dots, \quad (6.25)$$

where we design values of L to place eigenvalues of the $(A^{\mathfrak{D}} - LC)$ at the following locations

$$\left[0.7 \quad 0.7 \quad 0.2 \quad 0.05 \quad 0.05 \quad 0.05 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.4 \quad 0.4 \quad 0.4 \right]. \quad (6.26)$$

6.2.4 Software and integration

We implement the attitude controller in the Simulink model provided by the AscTec SDK Simulink Toolkit, 'onboard_matlab.mdl'. The model generates C code which is then flashed into the HLP. We implement the position controller in another Simulink model in the toolkit, 'UART_communication.mdl'. The model contains a 'real-time clock' which synchronizes Simulink time with real time. The sample time of this model is set to 0.02 second. The Optitrack measurement is integrated in this model by referring to the code produced by Or Hirshfeld [45], with necessary changes accommodating the coordinate systems we use.

All the quadrotor experiments are conducted using a PC with a 64-bit Windows 7 operating system, an Intel Core i7-2600K 3.7GHz processor, and 16 GB RAM. All the simulink models are ran by MATLAB R2016b.

6.2.5 Controller performance

Packet loss in the communication between PC and quadrotor occurs frequently and unpredictably during experiments. It renders the quadrotor in an uncontrolled state either when the packets containing attitude commands cannot reach the quadrotor, or when the packets containing the rotor speed measurements, which is necessary for the Luenberger observer, cannot reach PC. Figure 6.4 shows a 2-second window when packet loss occurs intermittently and irregularly. Hence, for safety considerations, we only test the position controller for hovering.

We show the performance of the hovering controller as the following. The

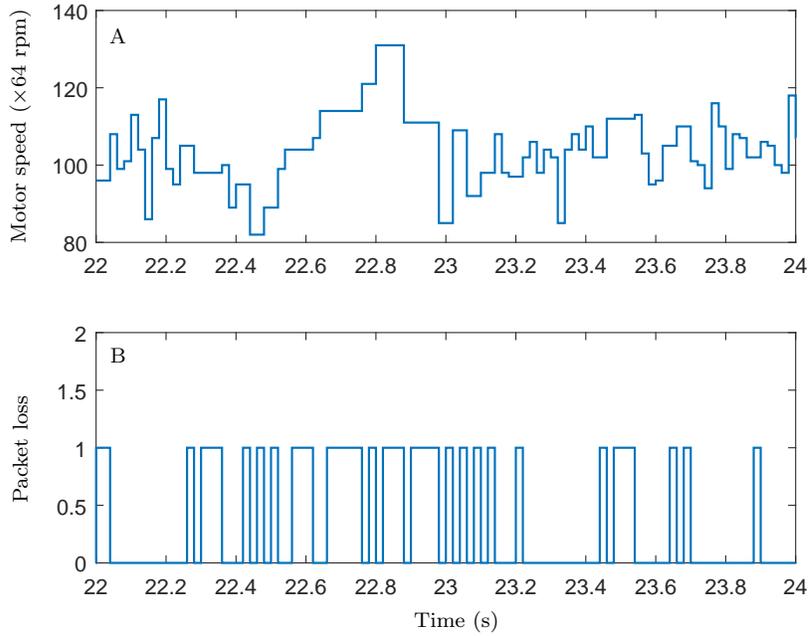


Figure 6.4: Speed of rotor 1 received by PC when packet loss occurs. (A) Simulink holds the previously received rotor speed value if the current packet is lost. (B) Value 1 indicates packet loss occurs while value 0 indicates packet is transmitted successfully.

desired hovering point is at $(x, y, z) = (0.1, 0, -0.3)$ m. The error is controlled within ± 0.05 m in the steady state, as shown in Figure 6.5 (A). The performance of the attitude controller is shown in Figure 6.5 (B)(C), where the measured attitude values follow the desired value closely. The Luenberger observer can reduce spikes greatly, compared to estimation by difference, as can be seen from Figure 6.6. It is worth noting that the Luenberger observer does not cause significant lags which is a typical drawback of a moving average filter.

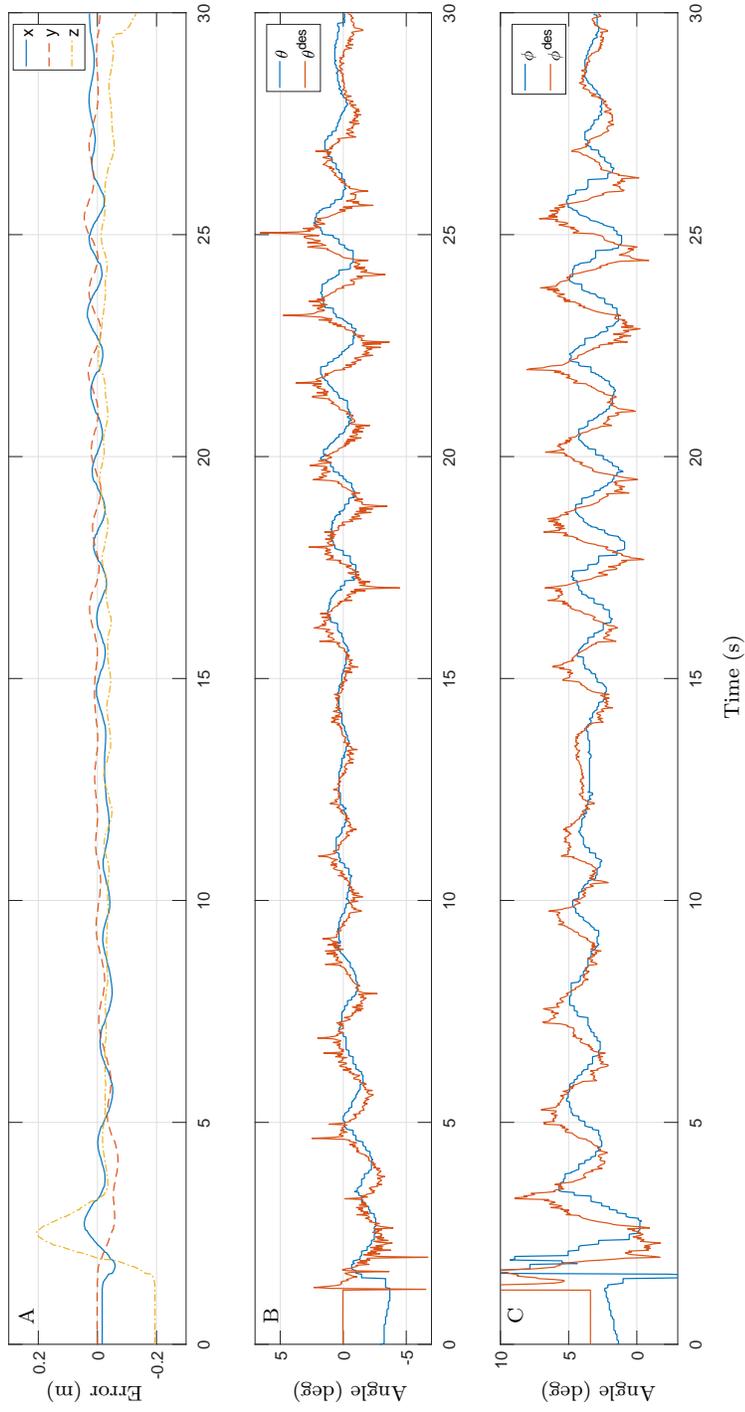


Figure 6.5: Performance of the position controller and the attitude controller. (A) The errors of the position controller are shown along (x, y, z) axes. (B) The desired roll θ^{des} is compared with measured roll θ . (C) The desired pitch ϕ^{des} is compared with measured pitch ϕ .

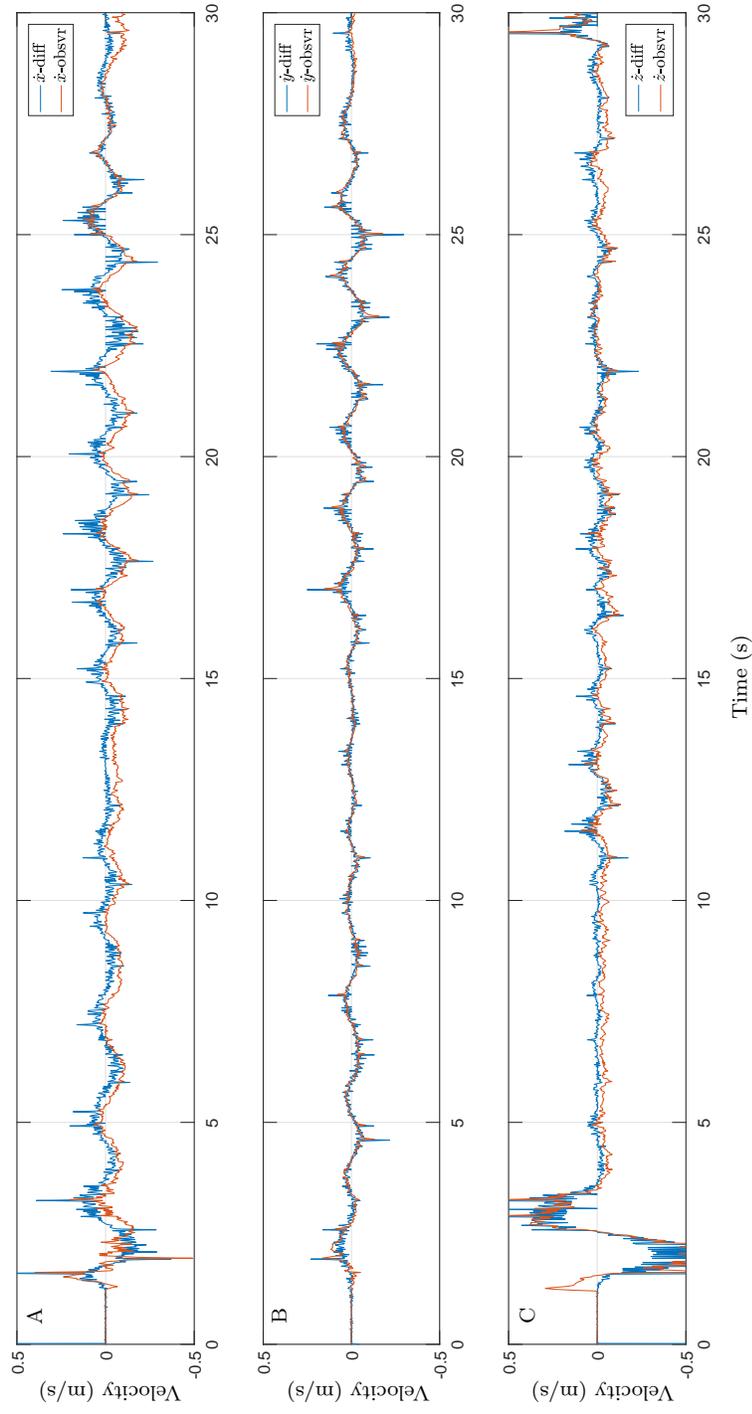


Figure 6.6: Velocity estimations by difference and by the Luenberger observer. The former is labelled by 'z-diff' while the latter is labeled by 'z-obsvr'. The estimations along (x, y, z) axes are displayed in (A), (B), and (C), respectively.

6.3 Ar.Drone

Ar.Drone is a small quadrotor built by Parrot. It consists of a carbon-fiber frame, four brushless motors, a LiPo battery, an ARM9 processor, sensors, and removable hulls. The sensors include a 3-DOF accelerometer, 3-DOF gyroscope, 2 ultrasonic sensors, and two cameras. The ARM9 processor runs a Linux based real-time operating system for onboard data fusion and motor control. More detailed introduction to the Ar.Drone can be found in [46, 47].

The Ar.Drone creates a WiFi network for communications between itself and remote devices. Through this network, the Ar.Drone can receive flying commands or modifying parameters from the remote device while the remote device can receive the navigation data measured by the onboard sensors of Ar.Drone.

Parrot provides an SDK [39] for users to program their applications on the remote device. Especially, users can send text strings following certain syntax, AT commands, to control the pitch, roll, yaw rate, and ascent speed of the Ar.Drone, which enables custom controllers on the Ar.Drone.

6.3.1 Dynamics and mathematical model

The dynamics of Ar.Drone is similar to the dynamics of AscTec Hummingbird except that Ar.Drone adopts '×' configuration while AscTec Hummingbird adopts '+' configuration. The coordinate systems are shown in Figure 6.7. We adopt (6.7) as the simplified dynamics of Ar.Drone with necessary modifications accommodating changes in the '+' configuration.

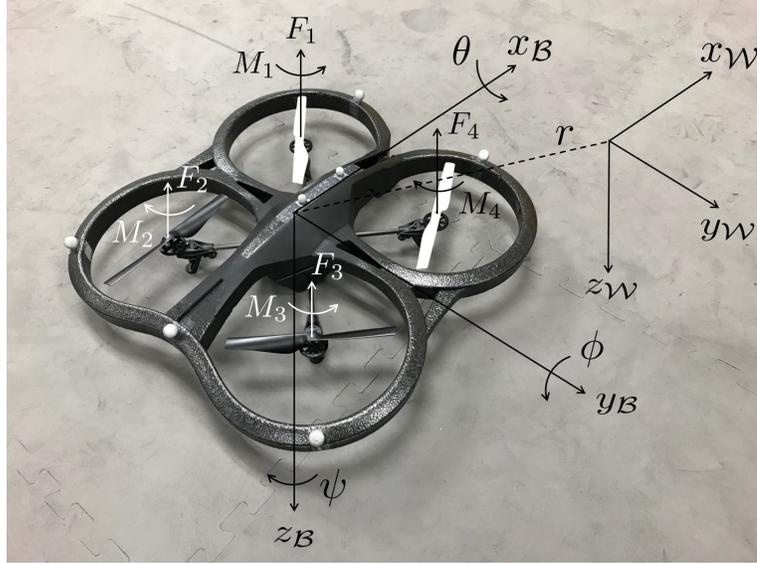


Figure 6.7: Coordinate systems and forces/torques acting on the Ar.Drone frame.

6.3.2 Quadrotor controller

We only need to build a position controller as in Section 6.2.3.2, since Ar.Drone contains an onboard attitude controller. We adopt a PD controller in the same form as in (6.15) to compute desired accelerations \ddot{x}^{des} and \ddot{y}^{des} and use (6.14a) as well as (6.14b) to compute desired roll θ^{des} and pitch ϕ^{des} , respectively. The desired ascent speed $-\dot{z}^{\text{des}}$ and desired yaw rate $\dot{\psi}^{\text{des}}$ are computed by the following PD control laws

$$-\dot{z}^{\text{des}} = k_{p,z}(z_T - z) + k_{d,z}(\dot{z}_T - \dot{z}), \quad (6.27)$$

$$\dot{\psi}^{\text{des}} = k_{p,\psi}(\psi_T - \psi) + k_{d,\psi}(\dot{\psi}_T - \dot{\psi}). \quad (6.28)$$

The desired values $(\theta^{\text{des}}, \phi^{\text{des}}, -\dot{z}^{\text{des}}, \dot{\psi}^{\text{des}})$ are then sent to the onboard controller.

6.3.3 Software and integration

We implement the position controller in a Simulink model. We use the same method to integrate Optitrack measurements as in Section 6.2.4. We adopt the MATLAB codes [48] for navdata decoding and AT command generation from the Advanced Control Systems Lab of the University of Oklahoma. The AT commands are sent from PC to the Ar.Drone via a custom C coded program.

6.3.4 Controller performance

The performance of the hovering controller is as follows. The desired hovering position is at $(x, y, z) = (0, 0, -1)$ m and $\psi = 0$ deg. The error in the steady state is controlled within ± 0.1 m in x and y , within ± 0.01 m in z , and within ± 5 deg in yaw, as shown in Figure 6.8.

The performance of the trajectory following controller is as follows. The desired trajectory is $x(t) = 0.5 \cos(2\pi t/5)$ m, $y(t) = \sin(2\pi t/5)$ m, $z(t)$ being a triangular wave with peak-to-peak 1.25 m, bias -1.375 m, and period 50 s, and $\psi(t) = 0$ deg. As can be seen from Figure 6.9, the position controller is able to steer the Ar.Drone to follow the trajectory closely.

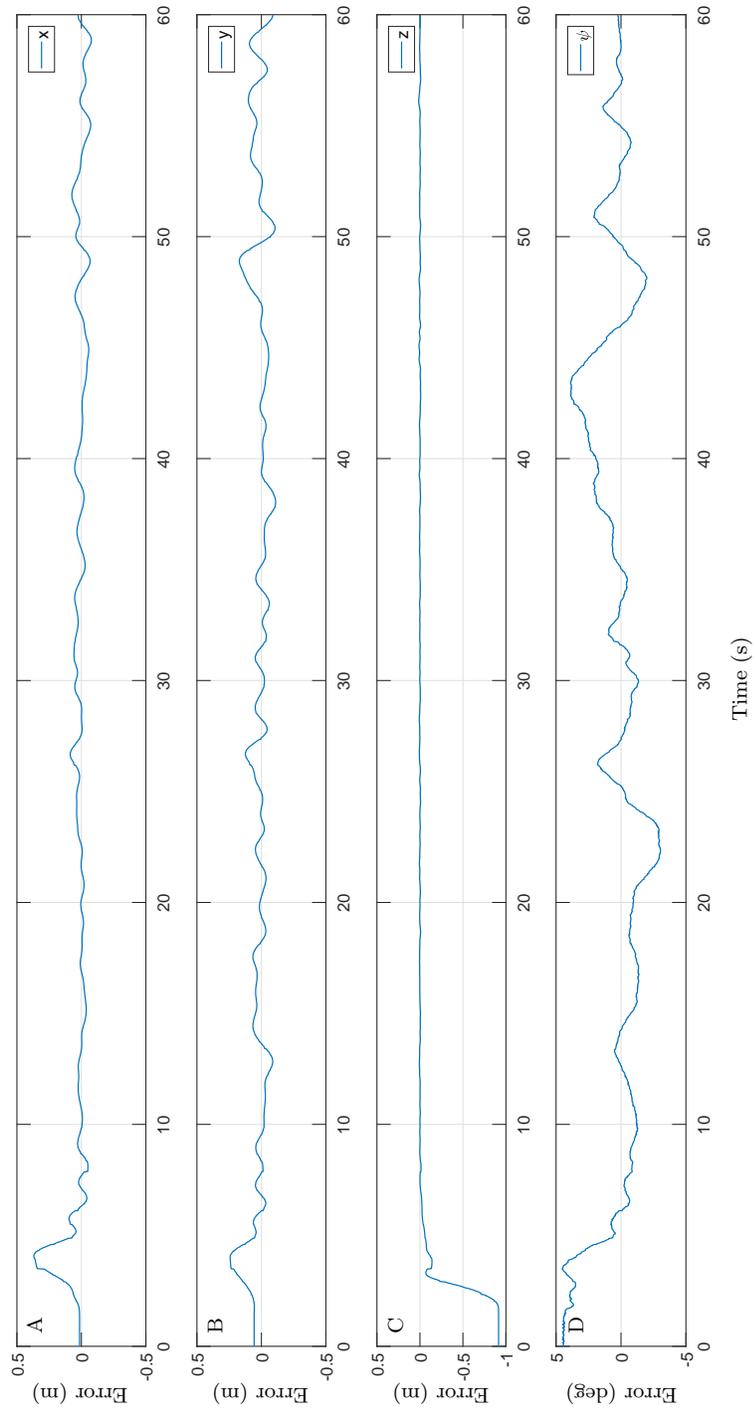


Figure 6.8: Performance of the position controller for hovering of Ar.Drone. The errors along (x, y, z) axes are shown in (A),(B), and (C). The error of yaw is shown in (D).

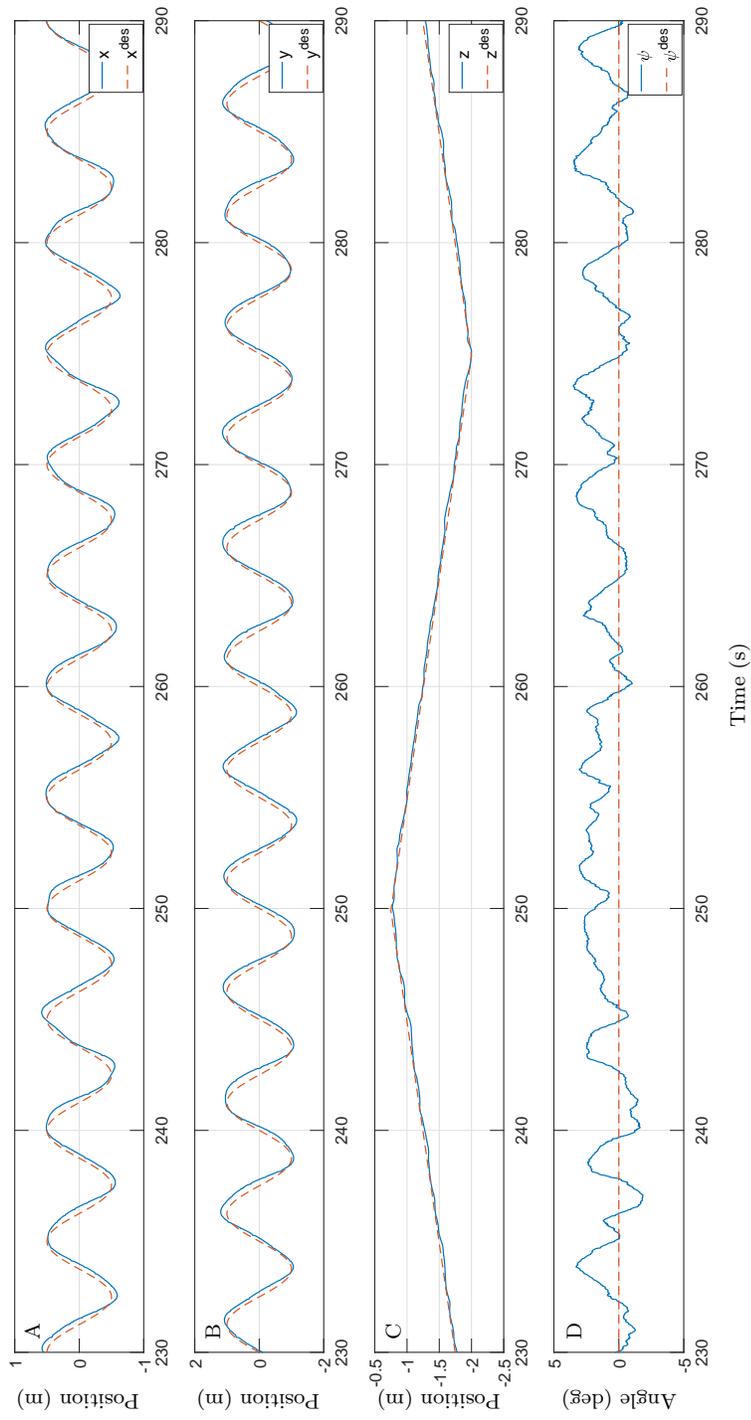


Figure 6.9: Performance of the position controller for trajectory following of Ar.Drone. The desired trajectory and actual trajectory along (x, y, z) axes are shown in (A),(B), and (C). respectively. The desired yaw and actual yaw are shown in (D).

6.4 Denied area experiment

6.4.1 Problem formulation

In this section, we show the design and implementation of a controller that steers the quadrotor to reach a target area which is enclosed within a denied area. The denied area and target all reside in the horizontal plane. So we only steer the quadrotor in the horizontal plane while the altitude and yaw remain fixed. And the controllers are based on the linearized model of the quadrotor in the hovering state.

The denied area is defined as previously in Chapter 5. Namely, in such area, the quadrotor will not be able to obtain its horizontal position which is necessary for a closed-loop control. Such denied area can be implemented by discarding horizontal position measurement once the quadrotor is detected to be within the denied area. As the theoretical results in Chapter 4 and Chapter 5 indicate, the controller will have two stages. The outer stage controller, which works in a closed-loop fashion, steers the quadrotor near, but outside, the the denied area; then the inner stage controller, which works in an openloop fashion, ensues to steer the quadrotor towards the target.

The linearized model has a state space containing (x, y) and its derivative, i.e., the state variable is

$$s^T = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}. \quad (6.29)$$

The control u contains the desired pitch and the desired roll, i.e.,

$$u^T = \begin{bmatrix} \phi^{\text{des}} & \theta^{\text{des}} \end{bmatrix}. \quad (6.30)$$

The state is directly observable, i.e., the output $q = s$. The state space model is the following

$$\begin{cases} \dot{s}(t) = As(t) + Bu(t) \\ q(t) = s(t) \end{cases}. \quad (6.31)$$

The position is measured by the Optitrack motion capture systems. The velocity is estimated by first computing the difference between consecutive position measurements, and then passing the difference through a finite impulse response filter whose discrete transfer function is the following

$$0.5 + 0.3z^{-1} + 0.15z^{-2} + 0.05z^{-3}. \quad (6.32)$$

6.4.2 Outer stage controller

We design the outer stage controller as a motion planner, where we solve (EP5) for the optimal switching state \bar{r}_{N_s} and terminal state \bar{r}_N . We only keep the switching state to generate the control \bar{u}_k^o in (5.5b) and the corresponding desired states \bar{s}_k for $k = 0, 1, \dots, N_s - 1$. Then the desired states $\bar{\mathbf{s}}_{0:N_s-1}$ are sent to the position controller as the desired positions and desired velocities to follow. The outer stage controller terminates when time reaches N_s and the inner stage controller takes over, if the quadrotor is outside the denied area. Otherwise the controller fails because the inner stage controller requires the position and velocity measurement when the outer stage terminates, which is forbidden inside the denied area.

6.4.3 Inner stage controller

The inner stage controller computes desired angles θ^{des} and ϕ^{des} in an openloop fashion and send them directly to the attitude controller. Since feedback cannot be applied for perturbation rejection, the controller must integrate the information of perturbation. Here, we adopt the problem formulation in (P7), which deals with perturbation by set propagation. The controller solves (P7) and send control actions in (5.25) to the onboard controller.

Following (P7), it remains to obtain the propagated set of perturbation \mathcal{W}_N^i , which originates from the set of perturbation \mathcal{W} . We will introduce next how to determine \mathcal{W} experimentally.

Consider the perturbed dynamics in (5.6). We can feed in the recorded state measurements and control actions to compute the perturbation, i.e., for an arbitrary $N_c > 0$,

$$w_k = s_{k+1} - (A^{\mathfrak{D}} s_k + B^{\mathfrak{D}} u_k), k = 0, 1, \dots, N_c, \quad (6.33)$$

where $A^{\mathfrak{D}}$ and $B^{\mathfrak{D}}$ are discretized matrices of A and B , respectively. Then,

$$\mathcal{W} = \text{convh}\{w_1, \dots, w_{N_c}\}, \quad (6.34)$$

and \mathcal{W}_N^i is computed by (5.15). Because we only need the position portion of \mathcal{W}_N^i in the optimization, we adopt an approximation method in computation to save time. We split \mathcal{W} into a position portion ${}^p\mathcal{W}$ and a velocity portion ${}^v\mathcal{W}$, which correspond to the projection of \mathcal{W} onto the position subspace and velocity subspace, respectively.

Then the approximate predicted set of perturbation ${}^p\bar{\mathcal{W}}_N$ in position subspace is computed by the following iterative procedure,

$$\begin{cases} {}^p\bar{\mathcal{W}}_{k+1} = A_1^{\mathfrak{D}} {}^p\bar{\mathcal{W}}_k \oplus A_2^{\mathfrak{D}} {}^v\bar{\mathcal{W}}_k \oplus B_1^{\mathfrak{D}} u_k \\ {}^v\bar{\mathcal{W}}_{k+1} = A_3^{\mathfrak{D}} {}^p\bar{\mathcal{W}}_k \oplus A_4^{\mathfrak{D}} {}^v\bar{\mathcal{W}}_k \oplus B_2^{\mathfrak{D}} u_k \end{cases}, {}^p\bar{\mathcal{W}}_{N_s} = {}^p\mathcal{W}, {}^v\bar{\mathcal{W}}_{N_s} = {}^v\mathcal{W}, \quad (6.35)$$

for $k = N_s, \dots, N - 1$, where $A_i^{\mathfrak{D}} \in \mathbb{R}^{2 \times 2}$, $i = 1, 2, 3, 4$, and $B_j^{\mathfrak{D}} \in \mathbb{R}^{2 \times 2}$, $j = 1, 2$, are partitions of $A^{\mathfrak{D}}$ and $B^{\mathfrak{D}}$, respectively, i.e.,

$$A^{\mathfrak{D}} = \begin{bmatrix} A_1^{\mathfrak{D}} & A_2^{\mathfrak{D}} \\ A_3^{\mathfrak{D}} & A_4^{\mathfrak{D}} \end{bmatrix}, B = \begin{bmatrix} B_1^{\mathfrak{D}} \\ B_2^{\mathfrak{D}} \end{bmatrix}. \quad (6.36)$$

The projection of \mathcal{W}_N^i onto the position subspace is a subset of the approximation set ${}^p\bar{\mathcal{W}}_N$. Let the \mathcal{V} -representation of ${}^p\bar{\mathcal{W}}_N$ be the following

$${}^p\bar{\mathcal{W}}_N = \text{convh}\{v_1, \dots, v_m\}. \quad (6.37)$$

We also approximate the target area by a polytope, which is an ellipse in the original problem (P7). Denote \mathcal{H} -representation of the polytopically approximated target area by $\{r \in \mathbb{R}^2 | G_t r \leq g_t\}$ where G_t and g_t have conformed dimensions. Then, the problem to solve is

$$\begin{aligned} & \underset{s_N \in \mathbb{R}^4}{\text{minimize}} \quad \frac{1}{2} \|(A^{\mathfrak{D}})^{N-N_s} s_{N_s} - s_N\|_{(\Delta^{\mathfrak{D}}(N_s, N))^{-1}}^2 \\ & \text{subject to} \quad G_t(s_{N,p} + v_k) \leq g_t, k = 1, \dots, m, \end{aligned} \quad (\text{P9})$$

where the projection of s_N onto the position subspace is denoted by $s_{N,p}$ and the inner stage initial state s_{N_s} is the state at the time of switching. Problem (P9) is a quadratic programming. Denote the solution of (P9) by s_N^* . Then the optimal control is in the same form of (5.26),

$$u_k^* = -(R^{\mathfrak{D}})^{-1} (B^{\mathfrak{D}})^T ((A^{\mathfrak{D}})^T)^{N-k-1} (\Delta^{\mathfrak{D}}(N_s, N))^{-1} ((A^{\mathfrak{D}})^{N-N_s} s_{N_s} - s_N^*), \quad (6.38)$$

for $k = N_s, N_s + 1, \dots, N - 1$.

6.4.4 System identification

Since we keep the yaw angle fixed at around 0 deg, we can decouple the state s and control u into two pairs: $(x, \dot{x}, \phi^{\text{des}})$ and $(y, \dot{y}, \theta^{\text{des}})$. Then we can rewrite (6.31)

into the following decoupled form

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = A_x \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + B_x \phi^{\text{des}}, \quad (6.39a)$$

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \end{bmatrix} = A_y \begin{bmatrix} y \\ \dot{y} \end{bmatrix} + B_y \theta^{\text{des}}. \quad (6.39b)$$

We suppose the system matrices A_x, A_y, B_x, B_y have following forms:

$$A_x = \begin{bmatrix} 0 & 1 \\ 0 & a_x \end{bmatrix}, A_y = \begin{bmatrix} 0 & 1 \\ 0 & a_y \end{bmatrix}, B_x = \begin{bmatrix} 0 \\ b_x \end{bmatrix}, B_y = \begin{bmatrix} 0 \\ b_y \end{bmatrix}. \quad (6.40)$$

The top left and bottom left entries of A_x are 0 since the position will not directly act on the velocity or position. The top right entry of A_x is 1 because of equality $\dot{x} = \dot{x}$. The top element of B_x is 0 because the desired pitch will not directly act on velocity. The desired pitch rather acts on acceleration, which makes the bottom element b_y to be determined. The bottom right entry of A_x corresponds to delay or velocity induced drag, which is to be determined. And the above argument applies to structures of A_y and B_y .

Hence, we can represent (6.39) by transfer functions in the \mathbf{s} -domain

$$\frac{\dot{x}(\mathbf{s})}{\phi^{\text{des}}(\mathbf{s})} = \frac{b_x}{\mathbf{s} - a_x}, \quad (6.41a)$$

$$\frac{\dot{y}(\mathfrak{s})}{\theta^{\text{des}}(\mathfrak{s})} = \frac{b_y}{\mathfrak{s} - a_y}. \quad (6.41b)$$

We use the System Identification Toolbox of MATLAB to identify a_x, a_y, b_x, b_y and obtain the following results

$$a_x = -0.35, a_y = 0.34, b_x = -4.87, b_y = 5.01. \quad (6.42)$$

Plugging (6.42) back to (6.39), we complete the system identification of the linearized dynamics of the Ar.Drone. From now on, we use (A, B) to denote the system matrices in (6.31) with values obtained in (6.42).

6.4.5 Perturbation set determination

We use (6.33) to collect vectors in the perturbation set and determine the perturbation set by (6.34). Then we adopt (6.35) to obtain the approximate propagated set of perturbation at terminal time N . Figure 6.10 displays the approximate propagated set of perturbation at various terminal times.

6.4.6 Optimization solver

We use Algorithm 1 to solve problem (EP5) for the outer stage problem, where the solver for semidefinite programming is SeDuMi 1.3 [49]. As for problem (P9), we use CVXGEN [50] to generate a C-coded QP solver. All the solvers are integrated in the Simulink model introduced in Section 6.3.3.

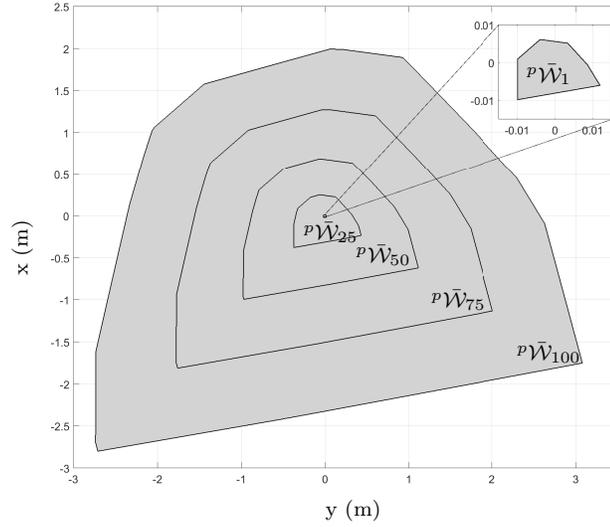


Figure 6.10: Size of the approximate propagated set of perturbation ${}^p\bar{\mathcal{W}}_N$. For the sake of simplicity, we take $N_s = 0$ in this figure.

6.4.7 Experiment results

The sample time is $T_s = 0.02$ s and we set the switching time to be 1.5 seconds and terminal time to be 2.5 seconds. The denied area and target area are both centered at the origin with radius 1.5 meters and 0.6 meter, respectively. We select ${}^p\bar{\mathcal{W}}_{25}$ as the propagated set of perturbation, which corresponds to an inner stage window of 0.5 second. We do not use ${}^p\bar{\mathcal{W}}_{50}$ since the approximation method makes it larger than the actual set ${}^p\mathcal{W}_{50}$. We use the following values for Q and R ,

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}, R = I_2. \quad (6.43)$$

The quadrotor is maintained at 1 meter altitude and yaw is fixed at around 0 deg as previously indicated.

The experiment is started manually by the operator through the Simulink interface. The operator initiate the outer stage at random time when the quadrotor is outside the denied area. Once the outer stage starts, problem (EP5) will be solved and a trajectory is planned with desired position and velocity generated for the following 1.5 seconds. In Simulink, the desired trajectory is fed to the position controller and corresponding attitude commands are sent to the onboard attitude controller for trajectory following. After 1.5 seconds, the inner stage starts automatically and it solves problem (P9). Then a sequence of desired attitude angles are generated and fed into the onboard attitude controller. In this stage, no position measurement in the horizontal plane takes part in the computation of the controller. The inner stage terminates in 1 second and return the control back to the position controller which holds the quadrotor at its current location until following commands arrive.

Figure 6.11 shows the trajectory of the quadrotor in the experiment. We can see the quadrotor follows the trajectory in the outer stage, then switch to inner stage near the optimal switching position and head towards the predicted set of terminal states inside the target area. The openloop commands of the inner stage are shown in Figure 6.12.

The average solver time of SeDuMi 1.3 is 65 ms (≈ 3 sample times) while average solver time of CVXGEN is 7.1 ms (≈ 0.3 sample time). It is worth noting that the solvers we use can solve the optimization problems fast without causing

huge delays that threaten stability.

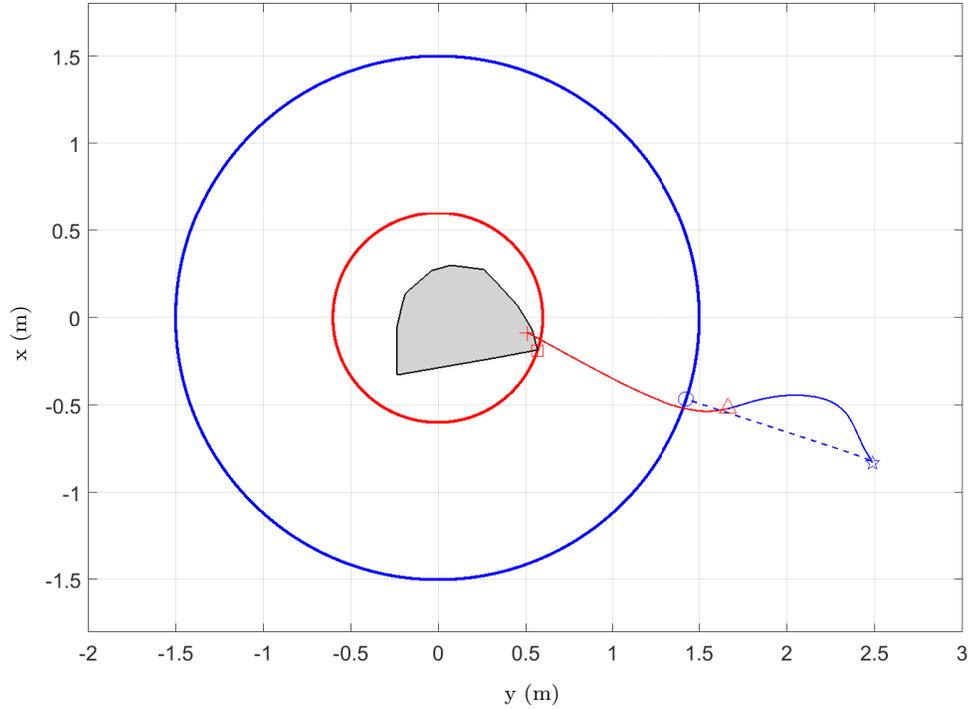


Figure 6.11: Trajectory of the Ar.Drone in an experiment. The big blue circle and big red circle are the denied area and target area, respectively. The Ar.Drone starts at the position marked by a blue pentagon. The optimal switching position and terminal position are marked by the blue circle and the red square, respectively. The optimal switching position is used to generate the optimal trajectory which is the dashed blue curve. The controller switches at the position marked by the red triangle. Then the predicted set of terminal states (marked by the polytope) is optimally placed in the target area and the Ar.Drone flies towards the predicted set. The trajectory in the inner stage is the solid red curve. The Ar.Drone arrives in the target area at the position marked by the red asterisk when the inner stage terminates.

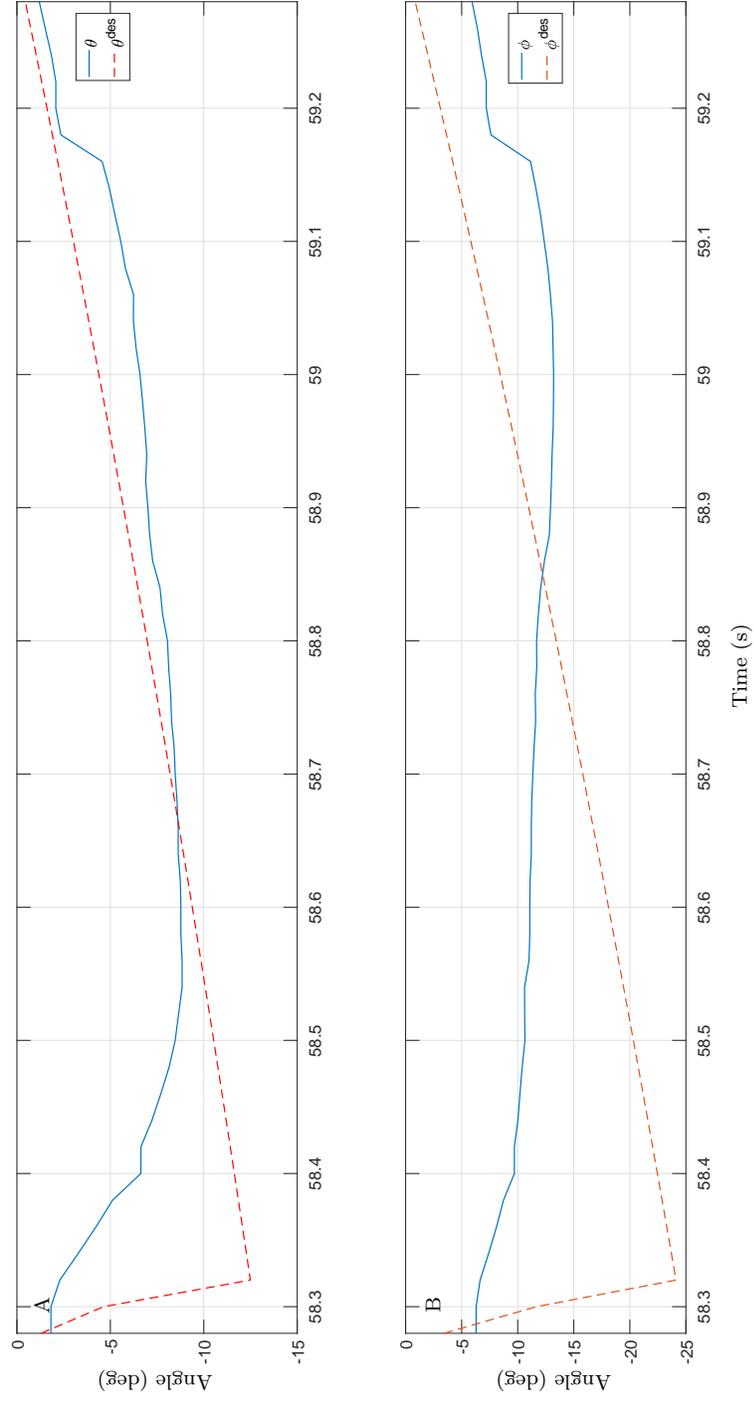


Figure 6.12: Attitude commands and actual attitude measurements in the inner stage. The outer stage starts at 58.28 s and ends at 59.28 s

Chapter 7: Conclusion and Future Work

In this thesis, we study the two-stage optimal control problem in which a mobile agent is steered to reach a target that is enclosed within a special area, within which it is either localization denied or time-costly.

We first formulate a two-stage optimal control problem in which the special area is time-costly. We consider deterministic dynamics of the mobile agent and convert the optimization problem into an equivalent nonconvex QC2QP using the LQR theory. The equivalent problem seeks the optimal switching state and the optimal terminal state, from which the optimal control can be obtained. We study the general QC2QP independently and prove the necessary and sufficient condition for strong duality. Then we propose solutions methods for the equivalent problem and suggest searching for the optimal terminal time using bisection. We demonstrate the trajectory of the mobile agent using the optimal controller in a numerical example.

Next, we formulate another two-stage optimal control problem in which the special area denies localization of the mobile agent. And we consider perturbations in the dynamics. The perturbation is handled by a robust controller where a variable horizon model predictive control problem is solved. The formulation takes into account a proper stage switching close to, but outside, the denied area as well as

a proper target reaching subject to the perturbation. The variable horizon enables the controller to decide the optimal switching time. And the controller is obtained by placing the predicted set of switching states and predicted set of terminal states at designated positions with the minimum cost. We demonstrate the performance of the robust controller in a simulation and give analysis on the failure of some trials.

We construct a quadrotor controller using the theoretical results. Experimental results show this controller can steer the quadrotor to reach a target area that is enclosed within a denied area. Moreover, this controller runs in real-time using off-the-shelf fast solvers. We also display the modeling and controller design of two quadrotor testbeds in the CPS and Cooperative Autonomy Laboratory.

We show experimental results of which a bat attempts to reach a target that resides in a denied area. The artificial denied area is created by broadcasting white noise when the bat enters such area. The trajectory and maneuver of the bat in the successful trials can benefit future research from a biological perspective.

There remain many interesting future research directions in this problem. On the one hand, more bat experiments can be conducted to statistically summarize the influence of the denied area on bat's target reaching. Meanwhile, the bat's trajectory and maneuvers may inspire new problem formulations and controller design based on nonlinear dynamics. On the other hand, results obtained in this thesis can be further investigated in various directions. One of those is to prove the robust controller in Chapter 5 provides guarantees on proper stage switching and target reaching. Also, one may explore efficient solvers or equivalent forms which can solve (P-MPC) for the global minimum. Another direction is in quadrotor experiment. Due to the size

of Ar.Drone, the controller proposed in Chapter 6 can only be tested for a denied area whose size is not large enough compared to that of the Ar.Drone. We hope future experiments can be conducted with smaller quadrotor platforms.

Bibliography

- [1] D. R. Griffin, F. A. Webster, and C. R. Michael, “The echolocation of flying insects by bats,” *Animal behaviour*, vol. 8, no. 3-4, pp. 141–154, 1960.
- [2] N. Ulanovsky and C. F. Moss, “What the bat’s voice tells the bat’s brain,” *Proceedings of the National Academy of Sciences*, vol. 105, no. 25, pp. 8491–8498, 2008.
- [3] L. Jakobsen and A. Surlykke, “Vespertilionid bats control the width of their biosonar sound beam dynamically during prey pursuit,” *Proceedings of the National Academy of Sciences*, vol. 107, no. 31, pp. 13930–13935, 2010.
- [4] A. J. Corcoran, W. E. Conner, and J. R. Barber, “Anti-bat tiger moth sounds: Form and function,” *Current Zoology*, vol. 56, no. 3, 2010.
- [5] W. Masters and K. Raver, “The degradation of distance discrimination in big brown bats (*Eptesicus fuscus*) caused by different interference signals,” *Journal of Comparative Physiology A*, vol. 179, no. 5, pp. 703–713, 1996.
- [6] L. A. Miller, “Arctiid moth clicks can degrade the accuracy of range difference discrimination in echolocating big brown bats, *Eptesicus fuscus*,” *Journal of Comparative Physiology A*, vol. 168, no. 5, pp. 571–579, 1991.
- [7] A. J. Corcoran and W. E. Conner, “Bats jamming bats: Food competition through sonar interference,” *Science*, vol. 346, no. 6210, pp. 745–747, 2014.
- [8] W. Ai and S. Zhang, “Strong duality for the CDT subproblem: a necessary and sufficient condition,” *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1735–1756, 2009.
- [9] J.-M. Peng and Y.-x. Yuan, “Optimality conditions for the minimization of a quadratic with two quadratic constraints,” *SIAM Journal on Optimization*, vol. 7, no. 3, pp. 579–594, 1997.

- [10] M. Celis, J. Dennis, and R. Tapia, “A trust region strategy for nonlinear equality constrained optimization,” *Numerical optimization*, vol. 1984, pp. 71–82, 1985.
- [11] J. Yuan, M. Wang, W. Ai, and T. Shuai, “New results on narrowing the duality gap of the extended Celis–Dennis–Tapia problem,” *SIAM Journal on Optimization*, vol. 27, no. 2, pp. 890–909, 2017.
- [12] A. Beck and Y. C. Eldar, “Strong duality in nonconvex quadratic optimization with two quadratic constraints,” *SIAM Journal on Optimization*, vol. 17, no. 3, pp. 844–860, 2006.
- [13] Y. Huang and S. Zhang, “Complex matrix decomposition and quadratic programming,” *Mathematics of Operations Research*, vol. 32, no. 3, pp. 758–768, 2007.
- [14] I. M. Bomze and M. L. Overton, “Narrowing the difficulty gap for the Celis–Dennis–Tapia problem,” *Mathematical Programming*, vol. 151, no. 2, pp. 459–476, 2015.
- [15] Y. Ye and S. Zhang, “New results on quadratic minimization,” *SIAM Journal on Optimization*, vol. 14, no. 1, pp. 245–267, 2003.
- [16] A. Lemon, A. M.-C. So, Y. Ye, *et al.*, “Low-rank semidefinite programming: Theory and applications,” *Foundations and Trends® in Optimization*, vol. 2, no. 1-2, pp. 1–156, 2016.
- [17] A. Richards and J. P. How, “Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility,” in *American Control Conference, 2003. Proceedings of the 2003*, vol. 5, pp. 4034–4040, IEEE, 2003.
- [18] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.
- [19] K. Tomiyama, “Two-stage optimal control problems and optimality conditions,” *Journal of Economic Dynamics and Control*, vol. 9, no. 3, pp. 317–337, 1985.
- [20] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal control*. John Wiley & Sons, 2012.
- [21] M. Grant, S. Boyd, and Y. Ye, “CVX: Matlab software for disciplined convex programming,” 2008.
- [22] R. C. Leishman and T. W. McLain, “Multiplicative extended kalman filter for relative rotorcraft navigation,” *Journal of Aerospace Information Systems*, vol. 12, no. 12, pp. 728–744, 2014.

- [23] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation,” Georgia Institute of Technology, 2015.
- [24] E. Jones, A. Vedaldi, and S. Soatto, “Inertial structure from motion with autocalibration,” in *Workshop on Dynamical Vision*, vol. 25, 2007.
- [25] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, “Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments,” in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 957–964, IEEE, 2012.
- [26] D. O. Wheeler, D. P. Koch, J. S. Jackson, T. W. McLain, and R. W. Beard, “Relative navigation: A keyframe-based approach for observable gps-degraded navigation,” *IEEE Control Systems*, vol. 38, no. 4, 2018.
- [27] R. A. DeCarlo, *Linear systems: A state variable approach with numerical implementation*. Prentice-Hall, Inc., 1989.
- [28] J. Löfberg, “YALMIP : A toolbox for modeling and optimization in matlab,” in *In Proceedings of the CACSD Conference*, (Taipei, Taiwan), 2004.
- [29] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, “Quadrotor helicopter flight dynamics and control: Theory and experiment,” in *AIAA Guidance, Navigation and Control Conference and Exhibit*, p. 6461, 2007.
- [30] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, “Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering,” in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pp. 3277–3282, IEEE, 2009.
- [31] M. Hehn and R. D’Andrea, “Quadrocopter trajectory generation and control,” in *IFAC world congress*, vol. 18, pp. 1485–1491, 2011.
- [32] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, “The grasp multiple micro-uav testbed,” *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.
- [33] D. Mellinger, N. Michael, and V. Kumar, “Trajectory generation and control for precise aggressive maneuvers with quadrotors,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.
- [34] D. Mellinger, M. Shomin, and V. Kumar, “Control of quadrotors for robust perching and landing,” in *Proceedings of the International Powered Lift Conference*, pp. 205–225, 2010.
- [35] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, “Precision flight control for a multi-vehicle quadrotor helicopter testbed,” *Control engineering practice*, vol. 19, no. 9, pp. 1023–1036, 2011.

- [36] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, “Towards a swarm of agile micro quadrotors,” *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, 2013.
- [37] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, “Crazyswarm: A large nano-quadcopter swarm,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pp. 3299–3304, IEEE, 2017.
- [38] A. Maksymiw, “AscTec AutoPilot.” <http://wiki.asctec.de/display/AR/AscTecAutoPilot>, Jan 2014.
- [39] S. Piskorski, N. Brulez, P. Eline, and F. DHaeyer, “AR.Drone Developer Guide.” <https://jpchanson.github.io/ARdrone/ParrotDevGuide.pdf>, May 2012.
- [40] A. Maksymiw and J. Link, “AscTec Communication Interface.” <http://wiki.asctec.de/display/AR/AscTec+Communication+Interface>, Mar 2014.
- [41] A. Maksymiw and A. Ryll, “Simulink Toolkit Manual.” <http://wiki.asctec.de/display/AR/Simulink+Toolkit+Manual>, Apr 2016.
- [42] C. P. De Prins, *Development of a quadcopter test environment and research platform*. PhD thesis, University of Maryland, College Park, 2015.
- [43] J. Diebel, “Representing attitude: Euler angles, unit quaternions, and rotation vectors,” *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.
- [44] M. Schreier, “Modeling and adaptive control of a quadrotor,” in *Mechatronics and Automation (ICMA), 2012 International Conference on*, pp. 383–390, IEEE, 2012.
- [45] O. Hirshfeld, “Motive-API-interface-to-Matlab-and-simulink-in-Real-time.” <https://github.com/orhirshfeld/Motive-API-interface-to-Matlab-and-simulink-in-Real-time>, August 2015.
- [46] P.-J. Bristeau, F. Callou, D. Vissiere, N. Petit, *et al.*, “The navigation and control technology inside the ar. drone micro uav,” in *18th IFAC world congress*, vol. 18, pp. 1477–1484, Milano Italy, 2011.
- [47] T. Krajník, V. Vonásek, D. Fišer, and J. Faigl, “AR-drone as a platform for robotic research and education,” in *International conference on research and education in robotics*, pp. 172–186, Springer, 2011.
- [48] A. L’Afflitto, “Control multiple AR.Drone 2.0 with Vicon feedback.” <https://www.mathworks.com/matlabcentral/fileexchange/63026-control-multiple-ar-drone-2-0-with-vicon-feedback#feedbacks>, May 2017.
- [49] J. Sturm, “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones,” *Optimization Methods and Software*, vol. 11–12, pp. 625–653, 1999. Version 1.05 available from <http://fewcal.kub.nl/sturm>.

- [50] J. Mattingley and S. Boyd, “CVXGEN: A code generator for embedded convex optimization,” *Optimization and Engineering*, vol. 13, no. 1, pp. 1–27, 2012.