Extremum-Seeking Guidance and Conic-Sector-Based Control of Aerospace Systems

by

Alex Walsh

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in The University of Michigan
2018

Doctoral Committee:

Assistant Professor James Richard Forbes, Chair
Professor Dennis Bernstein
Assistant Professor Dimitra Panagou
Professor Jing Sun

Alex Walsh
aexwalsh@umich.edu
ORCID iD: 0000-0002-3209-6479

To my family for their steadfast support and love.

Acknowledgements

First, I would like to thank my advisor, Prof. James R. Forbes. James showed incredible patience and dedication. He was ardent in his role as supervisor, teacher, and mentor. Despite all the challenges in the past four and a half years, James's support was unwavering. He provided a framework that allowed my ideas to flourish and my writing to substantially improve. He supported my studies that were directly and indirectly related to my dissertation, encouraged professional development and conference attendance, and facilitated my academic work through funding. He was above and beyond what I could have imagined from a Ph.D. supervisor.

I would like to thank my fellow Ph.D. students, particularly Ryan J. Caverly, Leila J. Bridgeman, David E. Zlotnik, and Frantisek M. Sobolic. Our discussions provided me with pleasure, understanding, and insight on a wide variety of subjects. I would like to thank Prof. James Cutler and the students Michigan Exploration Lab for providing me the opportunity to operate satellites, solder, design and implement guidance, estimation, and control systems, and participate in balloon launches. My Ph.D. experience would not be the same without this hands-on experience. Funding from NASA Armstrong Research Center and the National Science and Engineering Research Council of Canada was also critical to successful completion of my doctoral studies.

Finally, I must thank my friends and family. I met so many great people in Ann Arbor at the university, at the gym, and on my hockey team. All of you provided much needed balance. Trischa and Kayley kept me grounded if ever I ended up in the clouds, and my parents provided me with the foundation upon which everything that I have accomplished has been built.

Alex Walsh
December 2017

Preface

The elements of this dissertation that are original scholarship and distinct contributions to knowledge are as follows. Modified versions of most of this dissertation have appeared as archival journal or conference publications, which have been the subject of anonymous peer review. If the work has resulted in a publication, a reference is given. **The contributions for guidance of aerospace systems in Part II are as follows.**

- Chapter 4
  - A guidance and control algorithm formulated on $SO(3)$ that can account for attitude inclusion, and exclusion zones is introduced.
  - The guidance and control algorithm is implemented and tested on a spacecraft with three reaction wheels in simulation.
- Chapter 5
  - Constrained extremum-seeking guidance on $\mathbb{R}^3$ that uses a gain-projected Kalman filter is developed [1].
  - The constrained extremum-seeking guidance algorithm is demonstrated on a linear time-invariant plant, as well as the formation flight of two aircraft [1].
- Chapter 6
  - A relationship between the gradient from a Taylor series expansion of a function $SO(3) \to \mathbb{R}$ and the gradient that lies in the tangent space at a particular point of $SO(3)$ is derived [2].
  - An extremum-seeking guidance algorithm formulated on $SO(3)$ using a gain-projected Kalman filter that can account for attitude inclusion, and exclusion zones, and
  - an extremum-seeking guidance algorithm formulated on $SO(3)$ using an LMI-based Kalman filter that can take advantage of any type of LMI constraint on the control input and state are derived [2].

  - Extremum-seeking guidance for performance functions that map $SO(3) \to \mathbb{R}$ is demonstrated on a spacecraft maximizing signal received from a radiation source [2].

The contributions for control of nonlinear passive and conic systems in Part III are as follows.

- Chapter 8
  - A state-space structure with linear matrix inequality (LMI) conditions that show an affine system is very strictly passive (VSP) is derived [3].
  - The derived VSP controller is compared to an existing linear parameter varying (LPV) controller [3].
  - An affine VSP controller is used experimentally for the first time [3].
- Chapter 9
  - Sufficient LMI conditions are given for a polytopic system to be interior-conic. This is the analogous version to the Conic Sector Lemma for polytopic systems.
  - A modified version of the Conic Sector Lemma for polytopic systems is given that allows the upper conic bound to approach infinity.
  - Methods to determine conic bounds for polytopic systems are analyzed.
  - Two polytopic conic controller synthesis methods are given.
  - The conic controllers are tested on a heat exchanger and are compared to existing LPV controllers.

Except for one figure, all proofs, plots, illustrations, text, and numerical and experimental results in this dissertation were produced by Alex Walsh. James R. Forbes provided guidance and suggested edits throughout the dissertation. The LMI-based Kalman filter of Section 6.5.2 was derived based on work by Stephen A. Chee. Alex Walsh modified this Kalman filter for use with extremum-seeking guidance on $SO(3)$.

Table of Contents

List of Figures

Figure

List of Tables

Table

List of Appendices

Appendix

List of Abbreviations

ARE	algebraic Riccati equation
DCM	direction cosine matrix
EKF	extended Kalman filter
ISP	input strictly passive
KYP	Kalman-Yakubovich-Popov
LMI	linear matrix inequality
LPV	linear parameter varying
LTI	linear time invariant
LTV	linear time varying
MEKF	multiplicative extended Kalman filter
OSP	output strictly passive
PR	positive real
SDP	semidefinite program
SPR	strictly positive real
VSP	very strictly passive

List of Symbols

$\langle \cdot, \cdot \rangle$	a generic inner product
$\|\cdot\|$	a generic norm
$\mathcal{L}_2$	the Lebesgue space
$\mathcal{L}_{2e}$	the extended Lebesgue space
$\mathbb{R}$	the set of real numbers
$\mathbb{C}$	the set of complex numbers
$\mathbb{Z}$	the set of integers
$\mathbb{N}$	the set of natural numbers
$\mathbb{R}_{>0}$, $\mathbb{R}_{\geq 0}$, ...	real numbers in the interval $(0, \infty)$, $[0, \infty)$, ...
$S\mathbb{R}^n$	the set of symmetric matrices on $\mathbb{R}^{n \times n}$
$\mathbb{S}^n$	the sphere in $\mathbb{R}^{n+1}$
$SO(n)$	the n-dimensional special orthogonal group
$\overline{\mathcal{M}}$	the embedding space of the manifold $\mathcal{M}$
$\mathfrak{so}(n)$	the n-dimensional Lie algebra associated with $SO(n)$, the set of antisymmetric matrices on $\mathbb{R}^{n \times n}$
$T_{\mathbf{C}}SO(3)$	the tangent space to $SO(3)$ at $\mathbf{C} \in SO(3)$
$R_{\mathbf{C}}(\cdot)$	the retraction of an element in $T_{\mathbf{C}}SO(3)$ to $SO(3)$ at $\mathbf{C}$
$\boldsymbol{\nabla}(\cdot)$	the gradient
$\boldsymbol{\mathcal{P}}_s(\cdot)$	symmetric projection operator

$\boldsymbol{\mathcal{P}}_a(\cdot)$	antisymmetric projection operator
cone	interior conic sector
$\mathrm{tr}(\cdot)$	trace
$(\cdot)^{\mathsf{T}}$	transpose
$(\cdot)^{\times}$	cross operator
$(\cdot)^{\vee}$	uncross operator
$(\cdot)^{\perp}$	orthogonal complement
$\mathbf{0}$	zero matrix
$\mathbf{1}$	identity matrix
$\mathbf{1}_i$	the i-th column of the identity matrix
$\mathrm{diag}\,(\mathbf{M}_1, \ldots, \mathbf{M}_n)$	matrix with block diagonals $\mathbf{M}_1$, ..., $\mathbf{M}_n$, and zeros elsewhere
$\mathbf{M} > 0$	matrix $\mathbf{M}$ is positive definite. Positive semidefinite, negative definite, and negative semidefinite are denoted similarity.
$\star$	symmetric portion of a matrix
$\mathcal{F}_a$	the frame of reference a
$\underrightarrow{r}$	a physical vector
$\underrightarrow{r}^{y_i y_j}$	the position of point y_i relative to point y_j
$\underrightarrow{r}^{\bullet a}$	the time derivative of $\underrightarrow{r}$ with respect to $\mathcal{F}_a$
$\underrightarrow{r}^{y_i y_j \bullet a} = \underrightarrow{v}^{y_i y_j / a}$	velocity of point y_i relative to point y_j with respect to $\mathcal{F}_a$
$\underrightarrow{\boldsymbol{\mathcal{F}}}_a$	a vectrix, that is a matrix of unit length physical vectors that form a basis for $\mathcal{F}_a$, where $\underrightarrow{\boldsymbol{\mathcal{F}}}_a^{\mathsf{T}} = [\underrightarrow{a}^1 \;\; \underrightarrow{a}^2 \;\; \underrightarrow{a}^3]$
$\mathbf{r}_a$	the physical vector $\underrightarrow{r}$ resolved in $\mathcal{F}_a$
$\mathbf{C}_{ba}$	a DCM parameterizing the attitude of $\mathcal{F}_b$ relative to $\mathcal{F}_a$

$\mathbf{q}^{ba}$	$\mathbf{q}^{ba} \in \mathbb{R}^n$, a parameterization of the attitude of $\mathcal{F}_b$ relative to $\mathcal{F}_a$
$\underrightarrow{J}^{\mathcal{B}z}$	the second moment of mass of body $\mathcal{B}$ relative to point z
$\underrightarrow{\omega}^{ba}$	angular velocity of $\mathcal{F}_b$ relative to $\mathcal{F}_a$
$\mathcal{X} \to \mathcal{Y}$	a mapping from $\mathcal{X}$ to $\mathcal{Y}$
$\mathbf{x} \mapsto \mathbf{f}(\mathbf{x})$	indicates a mapping of an element $\mathbf{x} \in \mathcal{X}$, to a range, $\mathbf{f}(\mathbf{x}) \in \mathcal{Y}$
$\mathcal{R}(\cdot)$	range
$\mathcal{N}(\cdot)$	nullspace
$\mathbf{w}_k \sim \mathcal{N}(\bar{\mathbf{w}}_k, \mathbf{Q}_k)$	white noise, $\mathbf{w}_k$, with mean $\bar{\mathbf{w}} \in \mathbb{R}^n$, and covariance $\mathbf{Q}_k = \mathbf{Q}_k^\mathsf{T} \geq 0$
$\forall$	for all
$\lvert\cdot\rvert$	absolute value, or magnitude of a physical vector

Abstract

This dissertation studies guidance and control of aerospace systems. Guidance algorithms are used to determine desired trajectories of systems, and in particular, this dissertation examines constrained extremum-seeking guidance. This type of guidance is part of a class of algorithms that drives a system to the maximum or minimum of a performance function, where the exact relation between the function's input and output is unknown. This dissertation abstracts the problem of extremum-seeking to constrained matrix manifolds. Working with a constrained matrix manifold necessitates mathematics other than the familiar tools of linear systems. The performance function is optimized on the manifold by estimating a gradient using a Kalman filter, which can be modified to accommodate a wide variety of constraints and can filter measurement noise. A gradient-based optimization technique is then used to determine the extremum of the performance function. The developed algorithms are applied to aircraft and spacecraft.

Control algorithms determine which system inputs are required to drive the systems outputs to follow the trajectory given by guidance. Aerospace systems are typically nonlinear, which makes control more challenging. One approach to control nonlinear systems is linear parameter varying (LPV) control, where well-established linear control techniques are extended to nonlinear systems. Although LPV control techniques work quite well, they require an LPV model of a system. This model is often an approximation of the real nonlinear system to be controlled, and any stability and performance guarantees that are derived using the system approximation are usually void on the real system. A solution to this problem can be found using the Passivity Theorem and the Conic Sector Theorem, two input-output stability theories, to synthesize LPV controllers. These controllers guarantee closed-loop stability even in the presence of system approximation. Several control techniques are derived and implemented in simulation and experimentation, where it is shown that these new controllers are robust to plant uncertainty.

Part I

Introduction

Chapter 1

Introduction

1.1 Motivation and Objectives

Aerospace vehicles, such as aircraft and spacecraft, rely on estimation, guidance, and control strategies to realize mission objectives. Estimation is the process of filtering noise and biases corrupting sensor data in order to determine vehicle states, such as the position, velocity, attitude, and angular velocity of the vehicle, that may or may not be measured directly. Generating a path from the current estimated vehicle position to the desired vehicle position is guidance. Control is the application of inputs to a vehicle, such as forces and torques, such that the vehicle follows the generated path. This dissertation is divided into two parts, where each part focuses on guidance and control respectively.

1.1.1 Guidance

Control algorithms are typically designed to minimize the error between a measured output and a desired output. The desired output of the system can be determined in a variety of different ways, such as by a human operator, by a lookup table scheduled as a function of time, or by a guidance algorithm. Often guidance algorithms are optimal in some sense, where guidance commands are given by the solution of an optimization problem that takes a performance metric into account. The relation between a system's output and the performance metric may be known or unknown. An example of a known relationship is when the performance metric is to minimize the difference between a desired and estimated attitude. An example of an unknown relationship is the orientation of an antenna that maximizes signal strength of an unknown radiation source. In such a situation, an extremum-seeking guidance algorithm can be used to provide commands, such as a desired trajectory,

to a system in order to maximize or minimize the performance function. Examples of extremum seeking spans several industries, such as the automotive sector [6, 7, 8], the energy sector [9, 10], biomedical engineering [11], and aviation [12, 13].

This dissertation emphasizes attitude guidance. The nonlinear nature of the state space associated with spacecraft kinematics and dynamics compounds difficulties when designing a constrained attitude control algorithm. In particular, the most fundamental attitude parameterization, the direction cosine matrix (DCM), denoted as $\mathbf{C}$, must be orthonormal with a determinant equal to plus one. Matrices of this type constitute the special orthogonal group, denoted $SO(3)$. As such, the underlying state space is non-Euclidian. These nonlinearities pose a significant challenge to existing optimization techniques used to solve the associated constrained attitude control problem. Often, an attitude parameterization other than the DCM is used for attitude control, but for every problem circumvented by introducing a different attitude parameterization, another problem is introduced. For instance, the quaternion parameterization is nonunique, and Euler angles have a kinematic singularity [14]. DCMs possess no kinematic singularities, and are a global and unique attitude parameterization. As such, the first objective of Part II of this dissertation is to determine an attitude guidance algorithm using the DCM directly. The second objective is to develop an extremum-seeking guidance law on $SO(3)$ that accommodates attitude constraints. As a stepping stone to this objective, a secondary objective is a constrained extremum-seeking guidance law on $\mathbb{R}^3$.

1.1.2 Nonlinear Control

Input-output properties and associated stability theorems are indispensable tools to a control systems engineer. These tools have been widely used in robust, nonlinear, and optimal control. In particular, passivity and small gain properties, as well as their associated input-output stability theorems, are well-known and heavily used. The Passivity Theorem is commonly used for the robust control of mechanical and electrical systems, especially in the context of robotics [15]. Many robust control strategies, including $\mathcal{H}_\infty$ control, make use of the Small Gain Theorem, which has been used to robustly control aerospace [16], electric, and piezoelectric systems [17]. The Conic Sector Theorem has seen resurgence in recent years, and has been shown to be able to accommodate plants with passivity violations [18, 19]. The Extended Conic Sector Theorem [20] is a more general results that unifies many existing input-output theories [21].

Although passivity-based and conic-sector-based control have seen recent success,

linear parameter varying (LPV) control theory based on small gain remains a more widespread control paradigm. Nonlinear systems can be represented either approximately or exactly by LPV systems. When parameters are functions of exogenous signals only, the plant representation is referred to as an LPV system. When parameters are also functions of state or inputs, the plant representation referred to as quasi-LPV [22]. For brevity, these two cases are generally referred to as LPV. With origins in [23], the study of LPV systems has become popular because LPV controller synthesis is an extension of linear time-invariant (LTI) control methods. Several techniques to derive LPV controllers now exist [22, 24, 25, 26]. Closed-loop stability results involve the LPV controller in feedback with the LPV plant representation, and often, closed-loop stability of an LPV controller connected in feedback with the original nonlinear system cannot be claimed. However, in practice, the lack of stability guarantee of an LPV controller designed using an LPV model is mitigated because 1) the LPV plant representations can be precise enough for controller design, 2) robust control techniques are used to accommodate the differences between the LPV and original plant model, or 3) the LPV model may represent the nonlinear system exactly. Rigorous mathematical techniques to examine the effect of using LPV plant approximations are lacking, and experimentation can determine the practical usefulness of these approximations [22].

This dissertation aims to use input-output theory to overcome the challenges associated with using approximate LPV models to synthesize controllers for nonlinear systems. Specifically, once a passive plant has been identified, an LPV controller using the Passivity Theorem is sought. For systems that do not fall in the passivity framework, a tool to determine conic bounds for polytopic system is desired. In addition, control synthesis methods for polytopic conic systems must also be derived.

1.2 Outline and Contributions

General concepts are introduced in Chapter 2, including vector spaces, normed spaces, and optimization. These concepts are used throughout the dissertation. Part II focuses on the guidance of aerospace systems, and presents novel guidance methods for aircraft and spacecraft. Two chapters have application to spacecraft, and two chapters focus on extremum-seeking guidance. Specifically, Chapter 4 discusses optimal guidance and control of spacecraft, where algorithms determine both the trajectory and the control required to follow said trajectory. Chapters 5 and 6 cover extremum-seeking guidance, where the former focuses on objective functions that

map $\mathbb{R}^3 \to \mathbb{R}$, with application to formation flight of two aircraft. The latter chapter examines functions that map $SO(3) \to \mathbb{R}$, with guidance of spacecraft as an application example. The main contribution of this part of the dissertation is an extremum-seeking guidance algorithm on $SO(3)$. The development of an linear matrix inequality (LMI) constrained Kalman filter is also important as it allows estimation of gradients with a wide variety of constraints. LMIs are also numerially efficient to solve, which is vital for implementation purposes.

Part III is dedicated to LPV control using the Passivity and Conic Theorems. Within Part III, Chapter 7 gives an overview of gain, passivity, conic sectors, and LPV systems. Chapter 8 is devoted to LPV control in a passivity framework, where the work is derived form [3]. Chapter 9 focuses on control of interior-conic polytopic systems. The use of the Conic Sector Theorem in this chapter can allow for passivity violations and control of systems that do not fall under the passivity framework. The most significant contributions in this part are the passive and conic LPV controllers, as well as a method for determining conic bounds for a polytopic plant. Chapter 10 summarizes the importance of the findings in this dissertation and suggests new research directions. For a detailed list of contributions along with appearances in publications, see the Preface.

Chapter 2

Preliminaries

Two main themes of this dissertation are input-output theory and optimization. Input-output theory requires a review of signal properties, and is discussed in Section 2.1. Section 2.2 focuses on convex optimization and linear matrix inequalities.

2.1 Signals

Every system has inputs and outputs, and several properties must be considered in order to adequately study these systems.

2.1.1 Vector Spaces and Norms

Definition 2.1 (Vector Space [27])**.** Consider the space $\mathcal{V}$ and field $\mathbb{F}$. The operations of vector addition and scalar multiplication are defined as

1. for every pair $\mathbf{u},\ \mathbf{v} \in \mathcal{V}$, a unique element $\mathbf{z} = \mathbf{u} + \mathbf{v} \in \mathcal{V}$ is assigned and called their sum, and
2. for all $a \in \mathbb{F}$ and $\mathbf{u} \in \mathcal{V}$, there is a unique element $\mathbf{v} = a\mathbf{u} \in \mathcal{V}$ called their product.

The space $\mathcal{V}$ is a vector space if the following properties hold for all $\mathbf{u},\ \mathbf{v},\ \mathbf{w} \in \mathcal{V}$ and for all $a,\ b \in \mathbb{F}$, for addition

1. there exists an additive identity, denoted $\mathbf{0} \in \mathcal{V}$, such that $\mathbf{u} + \mathbf{0} = \mathbf{u}$,
2. there exists an inverse, denoted $-\mathbf{u} \in \mathcal{V}$, such that $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$,
3. the commutativity relationship $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$ holds,
4. the associativity property $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$ holds,

and for multiplication

1. there exists a multiplicative identity, denoted $1 \in \mathbb{F}$, such that $1\mathbf{u} = \mathbf{u}$,
2. the associativity property $a(b\mathbf{u}) = (ab)\mathbf{u}$ holds,
3. the vector distributivity property $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + b\mathbf{v}$ holds, and
4. the scalar distributivity property $(a + b)\mathbf{u} = a\mathbf{u} + b\mathbf{u}$.

Definition 2.2. A norm $\|\cdot\|$ on a vector space $\mathcal{V}$ is a function that maps $\mathcal{V} \to \mathbb{R}_{\geq 0}$, for all $\mathbf{u} \in \mathcal{V}$ that satisfies

1. $\|\mathbf{u}\| = 0$ if and only if $\mathbf{u} = \mathbf{0}$,
2. $\|a\mathbf{u}\| = |a| \|\mathbf{u}\|$ for all $a \in \mathbb{F}$, and
3. the triangle inequality $\|\mathbf{u} + \mathbf{v}\| \leq \|\mathbf{u}\| + \|\mathbf{v}\|$, for all $\mathbf{v} \in \mathcal{V}$.

2.1.2 Inner Product Spaces

Definition 2.3 (Inner Product)**.** An inner product $\langle \cdot, \cdot \rangle$ on a vector space $\mathcal{V}$ is a function mapping $\mathcal{V} \times \mathcal{V} \to \mathbb{F}$ such that

1. for all $\mathbf{u} \in \mathcal{V}$, $\langle \mathbf{u}, \mathbf{u} \rangle \geq 0$,
2. for all $\mathbf{u} \in \mathcal{V}$, $\langle \mathbf{u}, \mathbf{v} \rangle = 0$ if and only if $\mathbf{v} = \mathbf{0}$,
3. for all $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w} \in \mathcal{V}$ and a, $b \in \mathbb{F}$, $\langle \mathbf{u}, a\mathbf{v} + b\mathbf{w} \rangle = a \langle \mathbf{u}, \mathbf{v} \rangle + b \langle \mathbf{u}, \mathbf{w} \rangle$. This condition implies that the mapping $\mathbf{u} \mapsto \langle \mathbf{v}, \mathbf{u} \rangle$ is linear on $\mathcal{V}$.

An inner product space, $\mathcal{X}$, is a vector space that is equipped with an inner product. Every real inner product space is also a normed space since an inner product $\langle \mathbf{u}, \mathbf{u} \rangle$ for $\mathbf{u} \in \mathcal{X}$ satisfies Definition 2.2. An important property of an inner product space is the Cauchy-Schwartz inequality. Given the inner product space $\mathcal{X}$, the Cauchy-Schwartz inequality states that given $\mathbf{u} \in \mathcal{X}$ and $\mathbf{v} \in \mathcal{X}$, then

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq \|\mathbf{u}\| \|\mathbf{u}\| .$$

Definition 2.4 (Lebesgue Space)**.** The Lebesgue space, $\mathcal{L}_2$, is an inner product space, and is given by all square integral functions defined by

$$\mathcal{L}_2 = \left\{ \mathbf{x} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n \;\middle|\; \int_0^\infty \mathbf{x}^\mathsf{T}(t)\mathbf{x}(t)\mathrm{d}t < \infty \right\} .$$

The extended Lebesgue space space, $\mathcal{L}_{2e}$, is given by

$$\mathcal{L}_{2e} = \left\{ \mathbf{x} : \mathbb{R}_{\geq 0} \to \mathbb{R}^n \;\middle|\; \int_0^T \mathbf{x}^\mathsf{T}(t)\mathbf{x}(t)\mathrm{d}t < \infty, \; \forall T \in \mathbb{R}_{>0} \right\} .$$

The inner product of two functions, $\mathbf{u} \in \mathcal{L}_{2e}$ and $\mathbf{v} \in \mathcal{L}_{2e}$, is given by

$$\langle \mathbf{u}, \mathbf{v} \rangle_{2T} = \int_0^T \mathbf{u}^\mathsf{T} \mathbf{v} \, \mathrm{d}t,$$

and the norm of a function $\mathbf{u} \in \mathcal{L}_{2e}$ is given by

$$\langle \mathbf{u}, \mathbf{u} \rangle_{2T} = \|\mathbf{u}\|_{2T}^2 .$$

Definition 2.5. A function $\mathbf{u} \in \mathcal{L}_\infty$ if $\|\mathbf{u}\|_\infty = \sup_{t \in \mathbb{R}_{>0}} [\max_{i=1 \cdots n} |u_i(t)|] < \infty$ [28].

2.2 Optimization and LMIs

Optimal control and guidance is a central theme to this dissertation. Optimization is used to find optimal trajectories, estimator gains, and control in Part II, and is used to synthesize controllers in Part III. In all cases, optimization problems are reformulated as semidefinite programs (SDPs), which are convex, have a linear objective, and linear matrix inequality (LMI) constraints. This section defines convexity, LMIs, and techniques to transform quadratic objectives to linear objectives.

2.2.1 Convex Sets

Definition 2.6 (Convexity [29])**.** A set, $\mathcal{S}$, in a real inner product space is convex if for all $\mathbf{x},\ \mathbf{y} \in \mathcal{S}$ and $\alpha \in [0, 1]$, $\alpha\mathbf{x} + (1 - \alpha)\mathbf{y} \in \mathcal{S}$. A function, $f : \mathcal{S} \to \mathbb{R}$, is strictly convex if for all $\mathbf{x},\ \mathbf{y} \in \mathcal{S}$, $\alpha \in (0, 1)$, and $\mathbf{x} \neq \mathbf{y}$, $f(\alpha\mathbf{x}+(1-\alpha)\mathbf{y}) < \alpha f(\mathbf{x})+(1-\alpha)f(\mathbf{y})$.

Theorem 2.7 (Weierstrass)**.** *If a function $f(x) : D \to \mathbb{R}$ is continuous and all $x \in D$, where D is compact, then an extremum exists on D [30, p. 39].*

Proposition 2.8. *Suppose that $f : \mathcal{S} \to \mathbb{R}$ is strictly convex and continuous. If $\mathcal{S} \subset \mathbb{R}^n$ is closed, bounded, and convex, then a unique minimizer of f exists in $\mathcal{S}$.*

Proof. Existence of a solution comes from Theorem 2.7 since $\mathcal{S}$ is closed and bounded, thus compact, and since f is continuous. If the minimizer were not unique, then for any two minimizers $\mathbf{x},\ \mathbf{y} \in \mathcal{S}$ such that $f(\mathbf{x}) = f(\mathbf{y})$, then $\mathbf{z} = \alpha\mathbf{x}+(1-\alpha)\mathbf{y} \in \mathcal{S}$ since $\mathcal{S}$ is convex. Since f is strictly convex, then

$$f(\mathbf{z}) = f(\alpha\mathbf{x} + (1 - \alpha)\mathbf{y}) < \alpha f(\mathbf{x}) + (1 - \alpha)f(\mathbf{y}) = f(\mathbf{x}),$$

which contradicts the assumption that $\mathbf{x}$ and $\mathbf{y}$ are minimizers. □

2.2.2 Linear Matrix Inequalities

Definition 2.9. (Linear Matrix Inequality [31]) An LMI, $\mathbf{F} : \mathbb{R}^m \to \mathbb{R}^{n\times n}$, in the variable $\mathbf{x} \in \mathbb{R}^m$, is given by

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + \sum_{i=1}^{m} x_i \mathbf{F}_i \leq 0, \tag{2.1}$$

where $\mathbf{x} = [x_1 \ \cdots \ x_m]^\mathsf{T}$, $\mathbf{F}_i \in \mathbb{R}^{n\times n}$, $i = 0, \ldots, m$.

In [31], the LMI is defined in terms of strict inequalities. Here, an LMI is defined in terms of nonstrict inequalities to ensure existence and uniqueness of solutions when minimizing an objective subject to an LMI constraint. For example, consider minimizing $f(x) = x^2$ subject to $x > 0$. The minimum is at $x = 0$, which does not satisfy the strict inequality. When strict inequalities are needed, $\mathbf{F}(\mathbf{x}) + \epsilon\mathbf{1} \leq 0$ is used for some $\epsilon > 0$ such that

$$\mathbf{F}(\mathbf{x}) < \mathbf{F}(\mathbf{x}) + \epsilon\mathbf{1} \leq 0.$$

In this dissertation, LMIs are not written in the form of (2.1), rather they are written as a function of a matrix variable. Consider finding a matrix $\mathbf{P} = \mathbf{P}^\mathsf{T} \in \mathbb{R}^{n\times n}$ that satisfies

$$\mathbf{P}\mathbf{A} + \mathbf{A}^\mathsf{T}\mathbf{P} + \mathbf{Q} \leq 0, \tag{2.2}$$

where $\mathbf{P} > 0$. However, the LMI in (2.2) is not in the form of (2.1). Consider when $n = 2$, and $\mathbf{x} = [p_1 \ \ p_2 \ \ p_3]^\mathsf{T}$, and

$$\mathbf{P} = \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} = p_1\mathbf{E}_1 + p_2\mathbf{E}_2 + p_3\mathbf{E}_3,$$

where

$$\mathbf{E}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{E}_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{E}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

The matrices $\mathbf{E}_i$ are linear independent and symmetric, and form a basis for $\mathbf{P}$. The LMI (2.2) can be rewritten as

$$p_1(\mathbf{E}_1\mathbf{A} + \mathbf{A}^\mathsf{T}\mathbf{E}_1) + p_2(\mathbf{E}_2\mathbf{A} + \mathbf{A}^\mathsf{T}\mathbf{E}_3) + p_3(\mathbf{E}_2\mathbf{A} + \mathbf{A}^\mathsf{T}\mathbf{E}_3) + \mathbf{Q} \leq 0. \tag{2.3}$$

By letting $\mathbf{F}_0 = \mathbf{Q}$ and $\mathbf{F}_i = \mathbf{E}_i\mathbf{A} + \mathbf{A}^\mathsf{T}\mathbf{E}_i$, yields

$$\mathbf{F}_0 + \sum_{i=1}^{3} p_i \mathbf{F}_i \leq 0,$$

which is in the form of (2.1).

Proposition 2.10. *An LMI $\mathbf{F} : \mathbb{R}^m \to \mathbb{R}^{n\times n}$, is convex.*

Proof. Consider $\mathbf{x},\ \mathbf{y} \in \mathbb{R}^m$ and $\alpha \in [0, 1]$. Using Definition 2.6 yields

$$\mathbf{F}(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) = \mathbf{F}_0 + \sum_{i=1}^{m}(\alpha x_i + (1-\alpha)y_i)\mathbf{F}_i = \alpha\mathbf{F}(\mathbf{x}) + (1-\alpha)\mathbf{F}(\mathbf{y}).$$

□

LMIs are useful because they are convex, as shown in Proposition 2.10. Thus, when LMIs are the constraints of an optimization or feasibility problem, then Proposition 2.8 can be used.

Often an objective of an optimization problem is quadratic. The Schur complement is a useful tool for transforming an LMI that is quadratic in a variable $\mathbf{S} \in \mathbb{R}^{n\times m}$ to an LMI that is linear in $\mathbf{S}$.

Lemma 2.11 (Schur complement [31])**.** *Consider the matrices $\mathbf{Q} = \mathbf{Q}^\mathsf{T} \in \mathbb{R}^{n\times n}$, $\mathbf{R} = \mathbf{R}^\mathsf{T} \in \mathbb{R}^{m\times m}$, and $\mathbf{S} \in \mathbb{R}^{m\times n}$. The following LMI is satisfied*

$$\begin{bmatrix} \mathbf{Q} & \mathbf{S}^\mathsf{T} \\ \mathbf{S} & \mathbf{R} \end{bmatrix} > 0,$$

if and only if

$$\mathbf{Q} > 0, \quad \mathbf{Q} - \mathbf{S}^\mathsf{T}\mathbf{R}^{-1}\mathbf{S} > 0,$$

or

$$\mathbf{R} > 0, \quad \mathbf{R} - \mathbf{S}\mathbf{Q}^{-1}\mathbf{S}^\mathsf{T} > 0.$$

Since LMIs are almost always symmetric, a shorthand is used for symmetric matrices. A "$\star$" indicates symmetry for off-diagonal terms in a matrix. For example,

$$\mathbf{A} = \mathbf{A}^\mathsf{T} = \begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{C}^\mathsf{T} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \star & \mathbf{D} \end{bmatrix}.$$

Part II

Guidance of Aerospace Systems

Chapter 3

Introduction to Guidance of Aerospace Systems

3.1 Introduction

Spacecraft mission requirements often necessitate reorientation of the spacecraft bus subject to various constraints. Attitude pointing exclusion zones are defined for attitude sensors and sensitive instruments, such as star trackers and telescopes. Attitude pointing inclusion zones are defined for mission hardware, such as a communications antenna pointing towards a ground station, or solar panels pointing towards the sun. Similar constraints are common for many spacecraft. A noteworthy example is the James Webb Space Telescope (JWST), which has an 85° attitude exclusion zone towards the Sun, and a 45° inclusion zone allowable for observation [32]. Observations of an area of the Galaxy require a control algorithm to maintain or reorient the spacecraft to different attitudes while simultaneously satisfying these constraints.

In the case of the JWST, the relationship between the attitude and the objective is known, which Chapter 4 examines. Conversely, consider a spacecraft with attitude constraints attempting to track a signal originating from an unknown location. The objective is to maximize the received signal, but the attitude that maximizes this objective is unknown. This is an extremum-seeking guidance problem that maximizes a function $SO(3) \to \mathbb{R}$, and is challenging due to attitude constraints and the nonconvex nature of $SO(3)$. Thus, as a first step, extremum-seeking guidance for a function $\mathbb{R}^3 \to \mathbb{R}$ is discussed in Chapter 5, and then extremum-seeking guidance for $SO(3) \to \mathbb{R}$ is discussed in Chapter 6.

Attitude must be described by a parameterization for both guidance and control. As addressed in Section 1.1.1, the DCM is an attitude parameterization that is both

global and unique. Other parameterizations, such as quaternions and Euler angles, are frequently used, but they each suffer from certain deficiencies. The yaw-pitch-roll Euler angle sequence is used in Chapter 5. Once notation is reviewed in Section 3.2, Section 3.3 reviews axis/angle, Euler angles, quaternions, the DCM, and their relationship with angular velocity are discussed in detail. The purpose of Section 3.3 is to further motivate the use of DCMs for attitude guidance found in Chapter 4 and Chapter 6.

3.2 Notation

A frame of reference $\mathcal{F}_a$ is defined by a set of three orthonormal dextral basis vectors, $\underrightarrow{a}^1$, $\underrightarrow{a}^2$, and $\underrightarrow{a}^3$. [14]. The matrix of vectors, known as the vectrix, is defined as

$$\underrightarrow{\boldsymbol{\mathcal{F}}}_a = \begin{bmatrix} \underrightarrow{a}^1 \\ \underrightarrow{a}^2 \\ \underrightarrow{a}^3 \end{bmatrix}.$$

The physical vector $\underrightarrow{r}$ resolved in $\mathcal{F}_a$ and $\mathcal{F}_b$ is denoted by

$$\underrightarrow{r} = \underrightarrow{\boldsymbol{\mathcal{F}}}_a^{\mathsf{T}} \mathbf{r}_a = \underrightarrow{\boldsymbol{\mathcal{F}}}_b^{\mathsf{T}} \mathbf{r}_b. \tag{3.1}$$

The symmetric projection operator $\boldsymbol{\mathcal{P}}_s : \mathbb{R}^{n\times n} \to S\mathbb{R}^n$ projects the matrix $\mathbf{U} \in \mathbb{R}^{n\times n}$ to the set of symmetric matrices,

$$S\mathbb{R}^n = \{\mathbf{U} \in \mathbb{R}^{n\times n} \mid \mathbf{U} = \mathbf{U}^{\mathsf{T}}\},$$

where $\boldsymbol{\mathcal{P}}_s(\mathbf{U}) = \frac{1}{2}(\mathbf{U} + \mathbf{U}^{\mathsf{T}})$. The antisymmetric projection operator $\boldsymbol{\mathcal{P}}_a : \mathbb{R}^{n\times n} \to \mathfrak{so}(n)$ projects a matrix $\mathbf{U} \in \mathbb{R}^{n\times n}$ to the set of antisymmetric matrices,

$$\mathfrak{so}(n) = \{\mathbf{U} \in \mathbb{R}^{n\times n} \mid \mathbf{U} = -\mathbf{U}^{\mathsf{T}}\},$$

where $\boldsymbol{\mathcal{P}}_a(\mathbf{U}) = \frac{1}{2}(\mathbf{U} - \mathbf{U}^{\mathsf{T}})$. The operator $(\cdot)^\times$ maps $\mathbb{R}^3 \to \mathfrak{so}(3)$. For example, for a column matrix $\mathbf{v} = [v_1 \;\; v_2 \;\; v_3]^{\mathsf{T}}$, $\mathbf{v}^\times$ is given by

$$\mathbf{v}^\times = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}.$$

The operator $(\cdot)^\vee$ maps $\mathfrak{so}(3) \to \mathbb{R}^3$, such that $(\mathbf{v}^\times)^\vee = \mathbf{v}$. For example,

$$\begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix}^\vee = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \mathbf{v}.$$

Column matrices with unit norms are elements of spheres, defined as $\mathbb{S}^{n-1} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x}^\mathsf{T}\mathbf{x} = 1\}$.

3.3 Attitude Parameterizations

A contribution of this dissertation is extreum-seeking guidance using the DCM as opposed to another attitude parameterization. Several parameterizations of attitude exist, but the DCM is a global representation, is unique, and possesses no kinematic singularities [33]. In this section, properties of the DCM, axis/angle, Euler angles, and quaternions are reviewed to demonstrate the advantages and disadvantages of each. In general, a parameterization of the attitude of $\mathcal{F}_b$ relative to $\mathcal{F}_a$ is denoted $\mathbf{q}^{ba}$.

3.3.1 The Direction Cosine Matrix

Applying the left dot product of $\underrightarrow{\boldsymbol{\mathcal{F}}}_b$ to (3.1) and rearranging yields

$$\mathbf{r}_b = \underrightarrow{\boldsymbol{\mathcal{F}}}_b \cdot \underrightarrow{\boldsymbol{\mathcal{F}}}_a^\mathsf{T} \mathbf{r}_a,$$

where

$$\mathbf{C}_{ba} = \underrightarrow{\boldsymbol{\mathcal{F}}}_b \cdot \underrightarrow{\boldsymbol{\mathcal{F}}}_a^\mathsf{T}.$$

The matrix $\mathbf{C}_{ba} \in SO(3)$ is a direction cosine matrix (DCM) relating the orientation of $\mathcal{F}_b$ relative to $\mathcal{F}_a$, where

$$SO(3) = \{\mathbf{C} \in \mathbb{R}^{3\times 3} \mid \mathbf{C}^\mathsf{T}\mathbf{C} = \mathbf{1},\ \det \mathbf{C} = +1\}.$$

The DCM is the most fundamental attitude parameterization, as it is both global and unique.

The angular velocity of $\mathcal{F}_a$ relative to $\mathcal{F}_b$ resolved in $\mathcal{F}_c$ is given by $\boldsymbol{\omega}_c^{ba}$. Angular velocity is related to the DCM via Poisson's equation, given by

$$\dot{\mathbf{C}}_{ba} = -\boldsymbol{\omega}_b^{ba^\times} \mathbf{C}_{ba}. \tag{3.2}$$

3.3.2 Axis/Angle and the Rotation Vector

Theorem 3.1 (Euler's Theorem [14]). *The most general motion of a rigid body with one point fixed is a rotation about an axis through that point.*

Consider two frames, $\mathcal{F}_a$ and $\mathcal{F}_b$. Theorem 3.1 implies that the attitude of $\mathcal{F}_b$ relative to $\mathcal{F}_a$ can be globally parameterized by an axis and an angle, such as the unit length physical vector $\underset{\rightarrow}{a}^{ba}$ and ϕ^{ba}, which is indeed the case. The axis $\underset{\rightarrow}{a}^{ba}$ is the same resolved in either $\mathcal{F}_a$ or $\mathcal{F}_b$, which implies $\mathbf{a}_b^{ba} = \mathbf{a}_a^{ba} = \mathbf{a}^{ba}$, where $\mathbf{a}^{ba} \in \mathbb{S}^2$. The column matrix $\mathbf{a}^{ba}$ is in fact an eigenvector of $\mathbf{C}_{ba}$, where

$$\mathbf{C}_{ba}\mathbf{a}^{ba} = \mathbf{a}^{ba}.$$

The DCM $\mathbf{C}_{ba}$ is given by

$$\mathbf{C}_{ba}(\mathbf{q}^{ba}) = \cos(\phi^{ba})\mathbf{1} + (1 - \cos(\phi^{ba}))\mathbf{a}^{ba}\mathbf{a}^{ba^\mathsf{T}} - \sin(\phi^{ba})\mathbf{a}^{ba^\times}.$$

The axis/angle $\mathbf{q}^{ba} = [\mathbf{a}^{ba} \;\; \phi^{ba}]^\mathsf{T}$ is not unique for a given DCM for any $\mathbf{a}^{ba} \in \mathbb{S}^2$, since

$$\mathbf{C}_{ba}([\mathbf{a}^{ba^\mathsf{T}} \;\; \pi]^\mathsf{T}) = \mathbf{C}_{ba}([\mathbf{a}^{ba^\mathsf{T}} \;\; -\pi]^\mathsf{T}).$$

The *rotation vector*, denoted by $\underset{\rightarrow}{\phi}^{ba}$, is defined as a function of the axis/angle parameterization, where

$$\underset{\rightarrow}{\phi}^{ba} = \phi^{ba}\underset{\rightarrow}{a}^{ba}, \quad \text{and} \quad \boldsymbol{\phi}^{ba} = \phi^{ba}\mathbf{a}^{ba}.$$

The DCM is related to $\underset{\rightarrow}{\phi}^{ba}$ via the matrix exponential. Specifically, $\mathbf{C}_{ba} = e^{-\boldsymbol{\phi}^{ba^\times}}$ where $e^{-\boldsymbol{\phi}^{ba^\times}}$ is the matrix exponential given by

$$e^{-\boldsymbol{\phi}^{ba^\times}} = \cos(\phi^{ba})\mathbf{1} + (1 - \cos(\phi^{ba}))\left(\frac{\boldsymbol{\phi}^{ba}}{\phi^{ba}}\right)\left(\frac{\boldsymbol{\phi}^{ba}}{\phi^{ba}}\right)^\mathsf{T} - \sin(\phi^{ba})\left(\frac{\boldsymbol{\phi}^{ba}}{\phi^{ba}}\right)^\times, \tag{3.3}$$

where $\phi^{ba} = (\boldsymbol{\phi}^{ba^\mathsf{T}}\boldsymbol{\phi}^{ba})^{\frac{1}{2}}$ [34, pp. 42–45]. The matrix exponential maps $\mathfrak{so}(3) \to SO(3)$, and is used extensively in Chapter 6.

Computation of $\boldsymbol{\phi}^{ba}$ is given by $\boldsymbol{\phi}^{ba} = -(\ln(\mathbf{C}_{ba}))^\vee$. For a small rotation, that is when $\phi^{ba} \ll 1$, $\mathbf{C}_{ba}$ can be approximated as

$$\mathbf{C}_{ba} \approx \mathbf{1} - \boldsymbol{\phi}^{ba^\times}, \tag{3.4}$$

which can be derived by using small angle approximations in (3.3).

Angular velocity and the axis/angle parameterization are related by

$$\boldsymbol{\omega}_b^{ba} = \mathbf{S}_b^{ba}(\mathbf{q}^{ba})\dot{\mathbf{q}}^{ba}, \qquad \mathbf{S}_b^{ba}(\mathbf{q}^{ba}) = \begin{bmatrix} \sin(\phi^{ba})\mathbf{1} - (1-\cos(\phi^{ba}))\mathbf{a}^{ba\times} & \mathbf{a}^{ba} \end{bmatrix}, \tag{3.5}$$

$$\dot{\mathbf{q}}^{ba} = \boldsymbol{\Gamma}_b^{ba}(\mathbf{q}^{ba})\boldsymbol{\omega}_b^{ba}, \qquad \boldsymbol{\Gamma}_b^{ba}(\mathbf{q}^{ba}) = \begin{bmatrix} \frac{1}{2}(\mathbf{a}^{ba\times} - \cot(\frac{\phi^{ba}}{2})\mathbf{a}^{ba\times}\mathbf{a}^{ba\times}) \\ \mathbf{a}^{ba^{\mathsf{T}}} \end{bmatrix}. \tag{3.6}$$

A noticeable deficiency in (3.5) and (3.6) is that when $\phi = 2\pi k$, $k \in \mathbb{Z}$, the axis $\mathbf{a}^{ba}$ is undefined, and thus $\mathbf{S}_b^{ba}(\mathbf{q}^{ba})$ is undefined. In addition, the limit of $\cot(\frac{\phi^{ba}}{2})$ as ϕ approaches $2\pi k$ is infinite, and thus the magnitude of $\dot{\mathbf{a}}^{ba}$ also approaches infinity. This is known as a kinematic singularity. Kinematic singularities pose problems during simulation, since the attitude parameterization rates tend to infinity when ϕ approaches $2\pi k$, such as during set point regulation driving $\mathbf{C}_{ba}$ to $\mathbf{1}$.

3.3.3 Euler Angles

A DCM can be parametrized using three principal rotations. A principal rotation about the i^{th} basis vector of $\mathcal{F}_b$ or $\mathcal{F}_a$ by an angle α is denoted by $\mathbf{C}_{ba} = \mathbf{C}_i(\alpha)$. For example, the rotation sequence given by a $3-2-1$ sequence from $\mathcal{F}_a$ to $\mathcal{F}_b$, described by

$$\mathcal{F}_a \xrightarrow[3]{\psi} \xrightarrow[2]{\theta} \xrightarrow[1]{\phi} \mathcal{F}_b, \tag{3.7}$$

is related to the DCM $\mathbf{C}_{ba}$ by

$$\mathbf{C}_{ba} = \mathbf{C}_1(\phi)\mathbf{C}_2(\theta)\mathbf{C}_3(\psi),$$

where $\mathbf{q}^{ba} = [\phi \;\; \theta \;\; \psi]^{\mathsf{T}}$ are the Euler angles. The specific Euler angle sequence in (3.7) is known as a yaw, pitch, and roll. Other Euler angle combinations are also used in various applications, including $3-1-3$, $1-2-3$, etc.

Although Euler angles are used extensively for aircraft dynamics, and are used in Chapter 5, all Euler angles suffer from a condition where there can be a loss of degrees of freedom, and the sequence coalesces into one degree of freedom. For example, consider the $3-1-3$ Euler angle sequence given by

$$\mathbf{C}_{ba}(\mathbf{q}^{ba}) = \mathbf{C}_3(\alpha)\mathbf{C}_1(\beta)\mathbf{C}_3(\gamma). \tag{3.8}$$

When $\beta = 0$, (3.8) becomes

$$\mathbf{C}_{ba}(\mathbf{q}^{ba}) = \mathbf{C}_3(\alpha + \gamma),$$

and until $\beta \neq 0$, rotation about only the $\underset{\rightarrow}{a}^3$ and $\underset{\rightarrow}{b}^3$ axis is possible.

Euler angles are a global attitude representation, however, they are not unique. For any angle in a sequence, such as in (3.8), $\mathbf{C}_{ba}$ is the same for any $\alpha \pm 2\pi k$, $\beta \pm 2\pi k$, and $\gamma \pm 2\pi k,\ k \in \mathbb{Z}$.

The relationship between Euler angles and angular velocity depends on the specific Euler angle sequence in used. A $3-2-1$ Euler angle sequence has the relation

$$\boldsymbol{\omega}_b^{ba} = \mathbf{S}_b^{ba}(\mathbf{q}^{ba})\dot{\mathbf{q}}^{ba}, \qquad \mathbf{S}_b^{ba}(\mathbf{q}^{ba}) = \begin{bmatrix} \mathbf{1}_1 & \mathbf{C}_1(\phi)\mathbf{1}_2 & \mathbf{C}_1(\phi)\mathbf{C}_2(\theta)\mathbf{1}_3 \end{bmatrix}, \tag{3.9}$$

$$\dot{\mathbf{q}}^{ba} = \boldsymbol{\Gamma}_b^{ba}(\mathbf{q}^{ba})\boldsymbol{\omega}_b^{ba}, \qquad \boldsymbol{\Gamma}_b^{ba}(\mathbf{q}^{ba}) = \mathbf{S}_b^{ba^{-1}}(\mathbf{q}^{ba}). \tag{3.10}$$

The matrices $\mathbf{S}_b^{ba}(\mathbf{q}^{ba})$ and $\boldsymbol{\Gamma}_b^{ba}(\mathbf{q}^{ba})$ posses a kinematic singularity, which is given when

$$\det(\mathbf{S}_b^{ba}(\mathbf{q}^{ba})) = \cos(\theta) = 0,$$

that is when $\theta = \frac{\pi}{2} + \pi k,\ k \in \mathbb{Z}$.

3.3.4 Quaternions

Quaternions, also known as Euler parameters, are a widely used parameterization for attitude control. They are related to the axis/angle parameterization via

$$\mathbf{q}^{ba} = \begin{bmatrix} \boldsymbol{\epsilon} \\ \eta \end{bmatrix} = \begin{bmatrix} \mathbf{a}\sin\left(\frac{\phi}{2}\right) \\ \cos\left(\frac{\phi}{2}\right) \end{bmatrix},$$

where $\mathbf{a}$ is the axis of rotation, ϕ is the angle, and $\mathbf{q}^{ba} \in \mathbb{S}^3$ is the quaternion. Some superscripts ^{ba} have been dropped for brevity. Quaternions are advantageous because the DCM is related to the quaternion algebraically, that is without any trigonometric functions. This relationship is given by

$$\mathbf{C}_{ba}(\mathbf{q}^{ba}) = (1 - 2\boldsymbol{\epsilon}^\mathsf{T}\boldsymbol{\epsilon})\mathbf{1} + 2\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\mathsf{T} - 2\eta\boldsymbol{\epsilon}^\times.$$

Quaternions are a universal cover for $SO(3)$, that is they are a global attitude representation. However, quaternions are not unique since $\mathbf{C}_{ba}(\mathbf{q}^{ba}) = \mathbf{C}_{ba}(-\mathbf{q}^{ba})$.

Table 3.1: Summary of attitude parameterizations, including number of parameters and number of constraints.

Parameterization	Global	Unique	Kin. Singularity	no. param.	no. constr.
DCM	yes	yes	no	9	6
Axis/Angle	yes	no	yes	4	1
Quaternions	yes	no	no	4	1
Euler Angles	yes	no	yes	3	0

The relationship between quaternions and angular velocity is given by

$$\boldsymbol{\omega}_b^{ba} = \mathbf{S}_b^{ba}(\mathbf{q}^{ba})\dot{\mathbf{q}}^{ba}, \qquad \mathbf{S}_b^{ba}(\mathbf{q}^{ba}) = \begin{bmatrix} 2(\eta\mathbf{1} - \boldsymbol{\epsilon}^\times) & -2\boldsymbol{\epsilon} \end{bmatrix}, \tag{3.11}$$

$$\dot{\mathbf{q}}^{ba} = \boldsymbol{\Gamma}_b^{ba}(\mathbf{q}^{ba})\boldsymbol{\omega}_b^{ba}, \qquad \boldsymbol{\Gamma}_b^{ba}(\mathbf{q}^{ba}) = \frac{1}{2}\begin{bmatrix} \eta\mathbf{1} + 2\boldsymbol{\epsilon}^\times \\ -\boldsymbol{\epsilon}^\mathsf{T} \end{bmatrix}. \tag{3.12}$$

Quaternions do not suffer from a kinematic singularity, but since they double cover $SO(3)$, other control problems may arise such as unwinding.

3.3.5 Summary of Parameterizations

The advantages and disadvantages of each attitude parameterization are found in Table 3.1. Each parameterization except for the DCM has a clear disadvantage in terms of not being global, not being unique, or possessing a kinematic singularity. However, the DCM has more parameters and constraints than other parameterizations, and thus the quaternion has been the main focus of attitude control for the past four decades. In the recent literature there has been a push to evolve control algorithms from using an attitude parameterization, such as the quaternion, to using the DCM directly. For example, adaptive attitude control on $SO(3)$ is developed in [35], coordinated attitude control is studied in [36], passivity-based attitude control without and with actuator saturation is investigated in [37, 38], and global exponential attitude control is considered in [39]. Estimation using complementary filters on $SO(3)$ is studied in [40], and [41, 42, 43] develop various $SO(3)$-constrained attitude estimation methods. Other contributions to optimal control on $SO(3)$ include optimization without attitude constraints [44], the projection operator approach [45], and work with reference governors and model predictive control [46]. Unlike [46], dynamics are not linearized in this dissertation, and kinematics are not linearized at a point where the associated DCM is the identity. Finally, in [2], extremum-seeking

on $SO(3)$ with attitude inclusion and exclusion zones is developed using a Kalman filter.

Chapter 4

Guidance and Control on $SO(3)$

4.1 Introduction

Guidance is a critical component for spacecraft attitude control. Despite the fact a quaternion parameterization of attitude is nonunique, several different methods of path planning and optimal attitude control using quaternions exist. In the path planning approach, the optimization method considers the entire attitude trajectory during the optimization. Discretized quaternion attitude path planning examples include [47, 48], and [49], where the latter two papers build on the A^* method [50]. The sequential optimization method, presented in [51, 52], determines an optimal attitude by approximating the optimal control problem as a semidefinite program (SDP) at each time step. Note that by considering each time step individually, it is implied the entire attitude trajectory is not considered during the optimization. SDPs are particularly attractive since they are solved numerically efficiently using interior-point methods, and provided the problem is feasible, solutions are globally optimal [31, p. 1]. Due to the advantages of SDPs, the approach of this chapter is to reformulate the optimal control problem as an SDP. Solving nonconvex or nonlinear optimization problems by sequential linearization and SDPs is formalized in [53].

In the aforementioned optimal attitude control papers, the DCM is not used directly for control purposes. As discussed in Section 3.3, DCMs are global and unique, and there has been a recent push to extend and develop control and estimation algorithms for $SO(3)$. Therefore, as other papers develop $SO(3)$-based controllers and filters, the focus of this chapter is a constrained $SO(3)$-based optimal attitude controller that employs sequential linearization.

This chapter's main contribution is developing an optimal attitude control technique that evolves on constrained subsets of $SO(3)$. In particular, this chapter ex-

pands on [51] by considering attitude inclusion and exclusion zones, formulates the control problem as an SDP, and uses the DCM directly. This chapter also considers a spacecraft with reaction wheels as opposed to a rigid body with body torques applied to it for control purposes. The control method is then demonstrated in closed-loop on a spacecraft equipped with three reaction wheels. The controller solves for control torques subject to control magnitude and angular velocity constraints. The chapter begins with preliminaries in Section 4.2, the description of the control technique in Section 4.3, the numerical example in Section 4.4, and ends with closing remarks in Section 4.5.

4.2 Spacecraft Dynamics

Consider an inertial frame $\mathcal{F}_a$, and a spacecraft with body frame $\mathcal{F}_b$. The desired spacecraft attitude is denoted by frame $\mathcal{F}_d$. The DCM $\mathbf{C}_{ba}$ parameterizes the attitude of $\mathcal{F}_b$ relative to $\mathcal{F}_a$, and is assumed to be known deterministically. In practice, $\mathbf{C}_{ba}$ can be estimated using a suite of sensors such as a magnetometer, a sun sensor, a horizon sensor, and others [54]. The spacecraft moment of inertia relative to its centre of mass, c, resolved in $\mathcal{F}_b$ is $\mathbf{J}_b^{\mathcal{B}c} \in \mathbb{R}^{3\times3}$, where $\mathcal{B}$ denotes the spacecraft body. The spacecraft is equipped with three orthonormal reaction wheels, each with a spin-axis moment of inertia of I_s, and rotation speed $\dot{\boldsymbol{\gamma}} \in \mathbb{R}^3$. The control inputs are torques that drive the wheels, and are given by $\mathbf{u} \in \mathbb{R}^3$. The spacecraft's equations of motion are [35, 55, Appendix B.1]

$$\mathbf{J}_b^{\mathcal{B}c}\dot{\boldsymbol{\omega}}_b^{ba} + \dot{\boldsymbol{\omega}}_b^{ba\times}(\mathbf{J}_b^{\mathcal{B}c}\dot{\boldsymbol{\omega}}_b^{ba} + I_\mathrm{s}\dot{\boldsymbol{\gamma}}) + I_\mathrm{s}\ddot{\boldsymbol{\gamma}} = \mathbf{0}, \tag{4.1}$$

$$I_\mathrm{s}\ddot{\boldsymbol{\gamma}} = -\mathbf{u}, \tag{4.2}$$

$$\dot{\mathbf{C}}_{ba} = -\boldsymbol{\omega}_b^{ba\times}\mathbf{C}_{ba}. \tag{4.3}$$

Let $\mathcal{F}_{b_k}$ be $\mathcal{F}_b$ at time t_k. The discrete-time dynamics and kinematics between t_k and $t_{k+1}$are

$$\boldsymbol{\omega}_{b_{k+1}}^{b_{k+1}a} = \boldsymbol{\omega}_{b_k}^{b_k a} + T\mathbf{J}_{b_k}^{\mathcal{B}c^{-1}}\left(\mathbf{J}_{b_k}^{\mathcal{B}c}\boldsymbol{\omega}_{b_k}^{b_k a} + I_\mathrm{s}\dot{\boldsymbol{\gamma}}_k\right)^\times \boldsymbol{\omega}_{b_k}^{b_k a} + T\mathbf{J}_{b_k}^{\mathcal{B}c^{-1}}\mathbf{u}_k, \tag{4.4}$$

where

$$I_\mathrm{s}\ddot{\boldsymbol{\gamma}}_k = -\mathbf{u}_k, \tag{4.5}$$

$$\dot{\boldsymbol{\gamma}}_{k+1} = \dot{\boldsymbol{\gamma}}_k + T\ddot{\boldsymbol{\gamma}}_k. \tag{4.6}$$

The discrete-time attitude kinematics between t_{k+1} and t_{k+2} are

$$\mathbf{C}_{b_{k+2}a} = \exp(-T\boldsymbol{\omega}_{b_{k+1}}^{b_{k+1}a^\times})\mathbf{C}_{b_{k+1}a}. \tag{4.7}$$

The kinematics (4.7) are determined for t_{k+2}, since $\boldsymbol{\omega}_{b_{k+1}}^{b_{k+1}a}$ is a function of $\mathbf{u}_k$. The DCM $\mathbf{C}_{b_{k+1}a}$ is a function of $\boldsymbol{\omega}_{b_k}^{b_k a}$, which is a function of a past control input, $\mathbf{u}_{k-1}$, and cannot be changed.

When $k = 0$, the initial conditions are $\boldsymbol{\omega}_{b_0}^{b_0 a}$ and $\mathbf{C}_{b_0 a}$. The DCM $\mathbf{C}_{b_1 a}$ is given by Poisson's equation, $\mathbf{C}_{b_1 a} = \exp(-T\boldsymbol{\omega}_{b_0}^{b_0 a^\times})\mathbf{C}_{b_0 a}$. The DCM $\mathbf{C}_{b_1 a}$ is only dependent on the initial conditions, and not on the control input $\mathbf{u}_0$. The DCM $\mathbf{C}_{b_2 a}$ is determined via (4.7) once $\mathbf{u}_0$ and $\boldsymbol{\omega}_{b_1}^{b_1 a}$ are ascertained in the resulting optimal control problem.

4.3 Constrained Attitude Control on $SO(3)$

This section discusses the constrained attitude control problem on $SO(3)$ for the spacecraft from Section 4.2. The algorithm provides a desired attitude and angular velocity, which is a guidance step. However, the algorithm also provides a control input for the spacecraft. This chapter begins by setting up the optimal control problem and then discussing how it can be solved. The final algorithm is summarized by Algorithm 4.1.

4.3.1 Optimal Attitude Control Problem

The proposed control law determines a control input $\mathbf{u}_k$ that minimizes the objective function

$$J(\mathbf{u}_k, \mathbf{C}_{d_{k+2}a}) = -\mathrm{tr}(\mathbf{C}_{d_{k+2}a}\mathbf{B}_{k+2}^\mathsf{T}), \tag{4.8}$$

where $\mathbf{B}_{k+2} \in \mathbb{R}^{3\times 3}$, and $\mathbf{C}_{d_{k+2}a} \in SO(3)$ is the desired attitude of the spacecraft at time t_{k+2}. The control $\mathbf{u}_k$ and $\mathbf{C}_{d_{k+2}a}$ are related via a modification of (4.4) and (4.7). While solving the optimization problem, the control effort $\mathbf{u}_k$ yields the desired angular velocity, $\boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}a}$, given by

$$\boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}a} = \boldsymbol{\omega}_{b_k}^{b_k a} + T\mathbf{J}_{b_k}^{\mathcal{B}c^{-1}}\left(\mathbf{J}_{b_k}^{\mathcal{B}c}\boldsymbol{\omega}_{b_k}^{b_k a} + I_\mathrm{s}\dot{\boldsymbol{\gamma}}_k\right)^\times \boldsymbol{\omega}_{b_k}^{b_k a} + T\mathbf{J}_{b_k}^{\mathcal{B}_c^{-1}}\mathbf{u}_k. \tag{4.9}$$

The desired attitude is given by

$$\mathbf{C}_{d_{k+2}a} = \exp(-T\boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}a^\times})\mathbf{C}_{b_{k+1}a}. \tag{4.10}$$

Equation (4.8) is a general objective function that can be used for many purposes, including the minimization of the angle between two vectors, where one is resolved in the desired frame, $\mathbf{x}_{d_{k+2}}$, and the other is resolved in the inertial frame, $\mathbf{y}_a$. One example of the former is the direction of the solar cells of a spacecraft and an example of the latter is the position vector of the spacecraft relative to the sun. For example, let $J(\mathbf{u}_k, \mathbf{C}_{d_{k+2}a})$ be the objective function

$$J(\mathbf{u}_k, \mathbf{C}_{d_{k+2}a}) = -\mathbf{x}_{d_{k+2}}^{\mathsf{T}} \mathbf{C}_{d_{k+2}a} \mathbf{y}_a, \tag{4.11}$$

which can be rewritten as

$$\begin{aligned} J(\mathbf{u}_k, \mathbf{C}_{d_{k+2}a}) &= -\mathrm{tr}(\mathbf{x}_{d_{k+2}}^{\mathsf{T}} \mathbf{C}_{d_{k+2}a} \mathbf{y}_a) \\ &= -\mathrm{tr}(\mathbf{C}_{d_{k+2}a} \mathbf{y}_a \mathbf{x}_{d_{k+2}}^{\mathsf{T}}) \\ &= -\mathrm{tr}(\mathbf{C}_{d_{k+2}a} \mathbf{B}_{k+2}^{\mathsf{T}}), \end{aligned}$$

where $\mathbf{B}_{k+2}^{\mathsf{T}} = \mathbf{y}_a \mathbf{x}_{d_{k+2}}^{\mathsf{T}}$. In another example, it may be desirable to track N stars using a sensor whose lens is in direction $\mathbf{x}_{d_{k+2}}$, and the stars are in directions $\mathbf{y}_a^i$. The matrix $\mathbf{B}_{k+2}$ is thus given by

$$\mathbf{B}_{k+2}^{\mathsf{T}} = \sum_{i=1}^{N} w_i \mathbf{y}_a^i \mathbf{x}_{d_{k+2}}^{\mathsf{T}},$$

where each w_i are weighting scalars that allow the emphasis to be placed in the observation of specific stars.

In this chapter, the matrix $\mathbf{B}_{k+2}$ is chosen to be $\mathbf{B}_{k+2} = \mathbf{C}_{t_{k+2}a}$, where $\mathbf{C}_{t_{k+2}a}$ parameterizes the attitude of a target frame $\mathcal{F}_{t_{k+2}}$ relative to the frame $\mathcal{F}_a$. The DCM $\mathbf{C}_{t_{k+2}a}$ is a function of time, where

$$\mathbf{C}_{t_{k+2}a} = \exp(-T\boldsymbol{\omega}_{t_{k+1}}^{t_{k+1}a^\times}) \mathbf{C}_{t_{k+1}a}.$$

The objective function (4.8) is augmented with angular velocity error

$$J_1(\mathbf{u}_k, \mathbf{C}_{d_{k+2}a}, \boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}a}) = -w_{\mathrm{p}} \mathrm{tr}(\mathbf{C}_{d_{k+2}a} \mathbf{C}_{t_{k+2}a}^{\mathsf{T}}) + w_{\mathrm{d}} \boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}t^{\mathsf{T}}} \boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}t}, \tag{4.12}$$

where $\boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}t} = \boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}a} - \mathbf{C}_{d_{k+1}a} \mathbf{C}_{t_{k+1}a}^{\mathsf{T}} \boldsymbol{\omega}_{t_{k+1}}^{t_{k+1}a}$, and where $w_{\mathrm{p}} > 0$ and $w_{\mathrm{d}} > 0$ are weighting scalars. This objective is inspired by PD control and is the weighted sum of the attitude error and the angular velocity error. The resulting controller takes the form shown by the block diagram in Fig. 4.1.

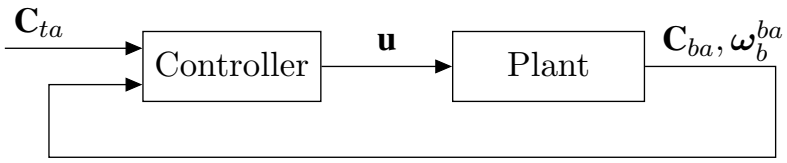


Figure 4.1: Attitude controller block diagram.

4.3.2 Constraints

The control algorithm must account for various constraints. The magnitude of the angular velocity is constrained by

$$\sqrt{\boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}a^{\mathsf{T}}} \boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}a}} \leq \omega_{\max}, \tag{4.13}$$

and the control effort constraint is given by

$$-u_{k,\max} \leq u_{k,i} \leq u_{k,\max}, \quad i = 1, \ldots, n_{\mathrm{w}}. \tag{4.14}$$

The attitude constraints are given by

$$\boldsymbol{\xi}_{d_{k+2}}^{i^{\mathsf{T}}} \mathbf{C}_{d_{k+2}a} \boldsymbol{\zeta}_a^i \geq \cos(\alpha_i), \ i = 1, \ldots, n_{\mathrm{c}}, \tag{4.15}$$

where the maximum angle between the vectors $\underrightarrow{\xi}^{\,i}$ and $\underrightarrow{\zeta}$ is α_i. The constraint (4.15) is a cone that can represent a pointing inclusion or exclusion zone [51, 2].

4.3.3 Solving the Optimal Control Problem

Problem 4.1. The optimal control problem to determine inputs $\mathbf{u}$ is given by minimizing (4.12) subject to (4.5), (4.6), (4.9), (4.10), (4.13)-(4.15). ■

Problem 4.1 is not an SDP due to i) a quadratic term in the objective function (4.12), ii) the exponential in (4.10), and iii) the norm constraint in (4.13). In addition, iv) the angular velocity and v) attitude constraints, (4.13) and (4.15) respectively, may be infeasible due to initial conditions or uncertainties in the system. As such, these constraints must be softened. These five modifications to Problem 4.1 must be addressed, as discussed next. The SDP version of Problem 4.1 is given by Problem 4.2 at the end of this section.

i) Quadratic term in (4.12): The optimization problem given by minimizing $\mathcal{J} = w_{\mathrm{d}} \boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}t^{\mathsf{T}}} \boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}t}$ is equivalent to minimize $\mathcal{J}' = w_{\mathrm{d}} z_\omega$, where z_ω must satisfy the LMI

given by [31]

$$\begin{bmatrix} z_\omega & \boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}t^\mathsf{T}} \\ \boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}t} & \mathbf{1} \end{bmatrix} \geq 0. \tag{4.16}$$

The objective $\mathcal{J}'$ is linear with an LMI constraint (4.16).

ii) Exponential in (4.10): The technique used to solve the optimal attitude control problem is sequential linearization, and as such, (4.10) is linearized using (3.4), which yields

$$\mathbf{C}_{d_{k+2}a} = (\mathbf{1} - T\boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}a^\times})\mathbf{C}_{b_{k+1}a}. \tag{4.17}$$

Placing a norm constraint on $\boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}a}$ with a small enough time step T ensures the linearization is valid. In fact, choosing an appropriate $\omega_{\max}$ and then enforcing (4.13) is equivalent to choosing a trust region [53].

iii) Norm constraint in (4.13): The norm constraint in (4.13) is simplified by using the Schur complement [31]. The angular velocity must satisfy the LMI given by

$$\begin{bmatrix} \omega_{\max}^2 & \boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}a^\mathsf{T}} \\ \boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}a} & \mathbf{1} \end{bmatrix} \geq 0. \tag{4.18}$$

iv) Angular velocity (4.13): Given that $\mathbf{u}_k$ is constrained by (4.14), there may not be a feasible control to satisfy (4.18). As such, the slack variable $b_\omega \geq 0$ with weight s_ω is added to the objective function, and (4.18) is modified such that

$$\begin{bmatrix} \omega_{\max}^2 & \boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}a^\mathsf{T}} \\ \boldsymbol{\omega}_{d_{k+1}}^{d_{k+1}a} & \mathbf{1} \end{bmatrix} \geq -b_\omega. \tag{4.19}$$

The slack variable softens the constraint to allow for control that cannot keep or bring the angular velocity below its maximum. Once the slack variable is minimized to zero, the constraint (4.19) is equivalent to (4.18). The slack variable prioritizes control constraints, and allows (4.18) to be enforced gradually.

v) Attitude constraints (4.15): Due to disturbances or initial conditions, the spacecraft can violate attitude constraints and it may not be feasible to satisfy these constraints at the next time step. As such, the constraints given by (4.15) are softened by introduce the slack variable $b_{\mathrm{a},i} \geq 0$ and weight s_i, resulting in the constraint

$$\boldsymbol{\xi}_{d_{k+2}}^{i^\mathsf{T}} \mathbf{C}_{d_{k+2}a} \boldsymbol{\zeta}_a^i \geq \cos(\alpha_i) - b_{\mathrm{a},i}, \;\; i = 1, \ldots, n_\mathrm{c}. \tag{4.20}$$

It is now possible to present Problem 4.2, the SDP version of Problem 4.1. Problem 4.1 is approximately equivalent to Problem 4.2 due to the approximation in (4.17).

Algorithm 4.1 Guidance and Control formulated on $SO(3)$

1: **for** $k = k_{\text{initial}}$ to k_{final} **do**
2: Estimate $\boldsymbol{\omega}_{b_k}^{b_k a}$, $\dot{\boldsymbol{\gamma}}_k$, and $\mathbf{C}_{b_k a}$.
3: Determine $\mathbf{C}_{b_{k+1} a}$ using $\boldsymbol{\omega}_{b_k}^{b_k a}$ and $\mathbf{C}_{b_k a}$ via Poisson's equation.
4: Solve Problem 4.2 to determine $\mathbf{u}_k$, $\mathbf{C}_{d_{k+2} a}$, and $\boldsymbol{\omega}_{d_{k+1}}^{d_{k+1} a}$.
5: Use $\mathbf{u}_k$ to control the spacecraft for time t_k to t_{k+1}.
6: **end for**

However, due to the angular velocity constraint and small time step, Problem 4.1 and Problem 4.2 are identical from a practical point of view.

Problem 4.2. Consider the objective function

$$J_2(\mathbf{u}_k, \mathbf{C}_{d_{k+2} a}, \boldsymbol{\omega}_{d_{k+1}}^{d_{k+1} a}, z_\omega, \mathbf{b}) = -w_{\text{p}} \text{tr}(\mathbf{C}_{d_{k+2} a} \mathbf{C}_{t_{k+2} a}^{\mathsf{T}}) + w_{\text{d}} z_\omega + \mathbf{s}^{\mathsf{T}} \mathbf{b}, \tag{4.21}$$

where $\mathbf{b} = [b_\omega \;\; b_{\text{a},1} \;\; \cdots \;\; b_{\text{a},n_{\text{c}}}]^{\mathsf{T}} \geq \mathbf{0}$ and $\mathbf{s} = [s_\omega \;\; s_1 \;\; \cdots \;\; s_{n_{\text{c}}}]^{\mathsf{T}}$. The objective is to minimize (4.21), subject to the constraints given by (4.5), (4.6), (4.9), (4.14), (4.17), (4.19), and (4.20). ■

As previously discussed, the role of the slack variables, which manifest themselves in the objective function in the term $\mathbf{s}^{\mathsf{T}}\mathbf{b}$, is to allow the problem to be feasible in case of uncertainty, and to prioritize the control constraints. The control algorithm has direct control over control torques, but not direct control over attitude and angular velocity due to uncertainty and dynamics. Therefore, while control torque constraints can always be satisfied, uncertainty in the physical system or initial conditions can result in attitude and angular velocity constraints to be violated. In effect, the term $\mathbf{s}^{\mathsf{T}}\mathbf{b}$ allows for constraints to be violated. However, $\mathbf{s}^{\mathsf{T}}\mathbf{b}$ first dominates the objective function, and drives $\mathbf{b}$ to a small value. Once $\mathbf{b}$ is arbitrarily small, constraints are satisfied, and the other terms in the objective function of (4.21) are minimized.

4.4 Numerical Example

In this section, a spacecraft endowed with three reaction wheels is controlled using the optimal control algorithm developed in Section 4.3. The moment of inertia of the spacecraft bus is $\mathbf{J}_b^{\mathcal{B}c} = \text{diag}\,(4, 4, 1)\ \text{kg} \cdot \text{m}^2$, each reaction wheel spin axis has moment of inertia $I_{\text{s}} = 0.04\ \text{kg} \cdot \text{m}^2$. The control law runs at 1 Hz, and thus $T = 1$ s. The dynamics of the spacecraft are simulated with a fourth-order Runge-Kutta numerical integration method. An extra step is taken to ensure that $\mathbf{C}_{b_k a}$

does not drift numerically from $SO(3)$. The scalar and matrix weights are given by $w_{\mathrm{p}} = 0.6, \quad w_{\mathrm{d}} = 1, \quad w_{\mathrm{d}} = 10^3 \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}^{\mathsf{T}}$. The target attitude is given by $\mathbf{C}_{t_k a} = \mathbf{C}_3\left(\frac{2\pi T t_k}{10}\right)$, until $k = 200$, and then the target attitude is held constant and equal to $\mathbf{C}_{t_k a} = \mathbf{1}$. This trajectory leads the spacecraft to spin around, and then stop and match its attitude with $\mathcal{F}_a$.

The attitude constraints are given by

$$\mathbf{x}_b^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \ \mathbf{y}_a^1 = -\begin{bmatrix} 0.8944 \\ 0 \\ -0.4472 \end{bmatrix}, \ \alpha_1 = 180^\circ - 20^\circ, \tag{4.22}$$

$$\mathbf{x}_b^2 = \begin{bmatrix} 0 \\ 0.0995 \\ 0.9950 \end{bmatrix}, \ \mathbf{y}_a^2 = -\begin{bmatrix} 0.5774 \\ 0.5774 \\ 0.5774 \end{bmatrix}, \ \alpha_3 = 180^\circ - 30^\circ, \tag{4.23}$$

$$\mathbf{x}_b^3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \ \mathbf{y}_a^3 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ \alpha_2 = 40^\circ, \tag{4.24}$$

where (6.65) and (6.66) are the exclusion zones, and (4.24) is the inclusion zone. The SDP given by Problem 4.2 is solved using MOSEK [56] through the YALMIP interface [57].

The simulation results of the closed-loop system of the controller and spacecraft simulated with attitude constraints are shown in Fig. 4.2 and Fig. 4.3. As shown in Fig. 4.2a, the initial reaction of the system is to satisfy the attitude constraints. In this case, the slack variables dominate the objective function given in (4.21) so that the error associated with being outside or inside an inclusion or exclusion zone is minimized quickly. The constraints are satisfied when the lines of the constraints are greater than zero.

Fig. 4.3a shows the angular velocity and angular velocity constraints. The initial angular velocity is at a maximum when the controller drives the system to an attitude that satisfies the constraints. The angular velocity is then nonzero as the system tracks $\mathbf{C}_{ta}$. Variations in the angular velocity occur as the spacecraft encounters different constraints on its trajectory. When $\mathbf{C}_{ta} = \mathbf{1}$ after 200 s, the angular velocity of the system tends to zero.

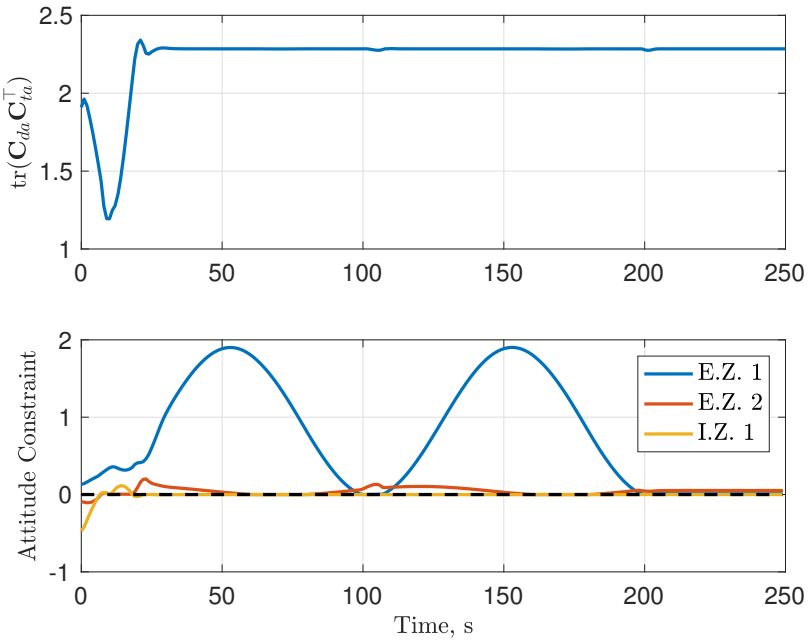
Fig. 4.3b shows the reaction wheel torques on the spacecraft. The torques give an

angular acceleration to create the angular velocity required to satisfy the constraints, and then to drive the angular velocity back to zero. The torque limits are set to 0.05 N·m.

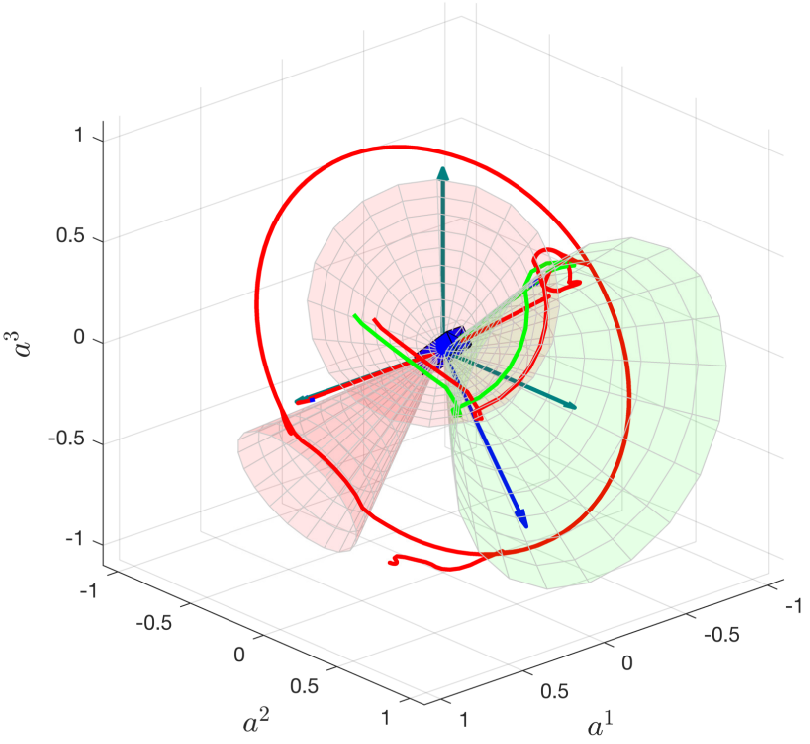
Fig. 4.2b shows a the spacecraft at the end of the simulation. The tips of the vectors $\underrightarrow{x}^i$ that define the attitude constraints draw a locus of points to visualize the trajectory of the spacecraft. The red lines are the exclusion zones and these lines avoid the red cones, which are the exclusion zones. The green line is associated with the inclusion zone. Its initial path is perpendicular to the green cone, as it takes the shortest route possible to satisfy the inclusion zone constraint.

4.5 Closing Remarks

The main contribution of this chapter is to solve an optimal attitude control problem for a spacecraft subject to attitude constraints, while treating the attitude constraints in $SO(3)$. At each time step, a semidefinite program is solved to determine the reaction wheel control torques. Similar to other solutions that use sequential linearization [51, 53], this controller does not consider the entire attitude trajectory during the optimization. Given the three attitude constraints, the angular velocity, and control constraints in the numerical example considered, the controller is able to provide effective attitude tracking.

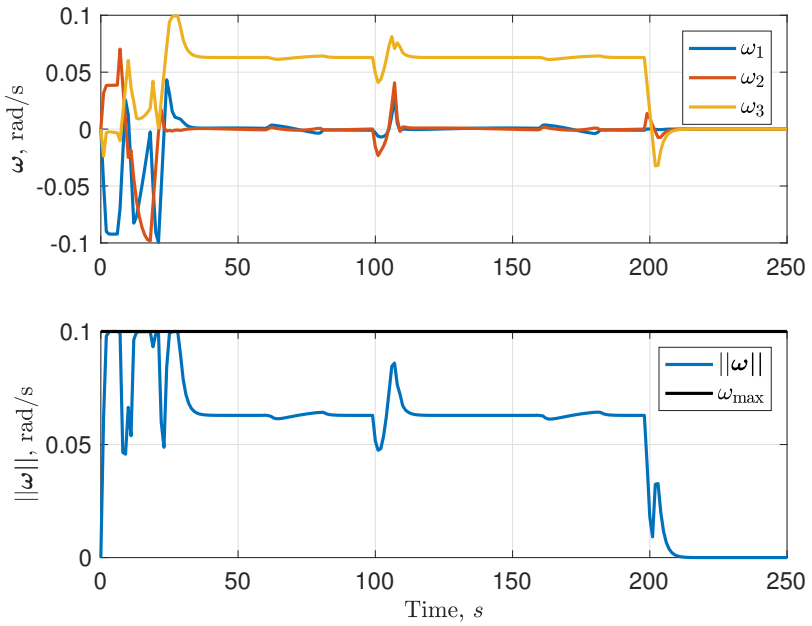


(a) Objective function and exclusion zone (E.Z.) and inclusion zones (I.Z.) for optimal control problem. Constraints are satisfied when lines are above black dotted line (zero).

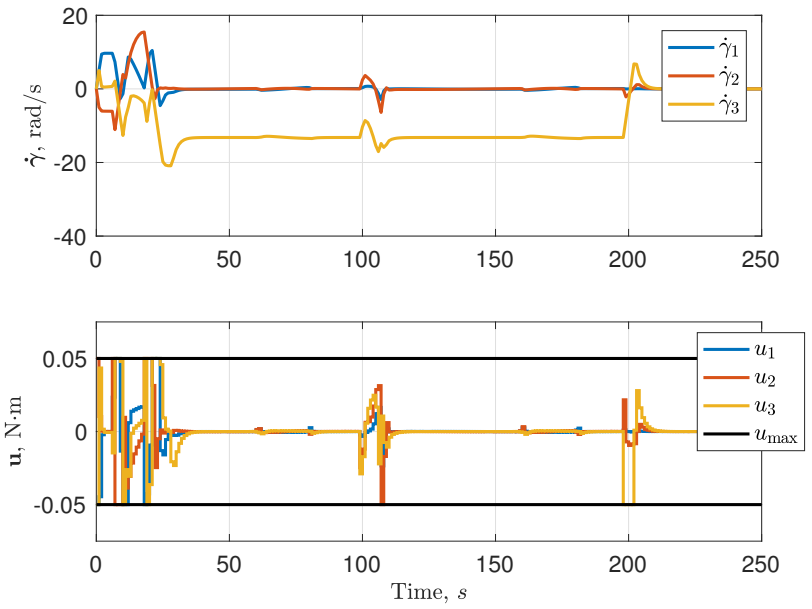


(b) Three-dimensional plot of spacecraft attitude. Exclusion zones are the red cones the inclusion zone is the green cone.

Figure 4.2: Spacecraft objective and trajectory for attitude tracking.



(a) Spacecraft angular velocity and angular velocity constraint.



(b) Reaction wheel speeds and torques.

Figure 4.3: Spacecraft states and control for attitude tracking.

Chapter 5

Extremum-Seeking Guidance on $\mathbb{R}^3$

5.1 Introduction

Extremum-seeking uses a pre-determined optimization method to determine the extremum of the objective function. Newton's method requires knowledge of the Hessian and gradient of the objective function. In one formulation of extremum-seeking guidance, a Kalman filter estimates the gradient and Hessian of the objective function, which are then used to determine the objective function's extremum [13]. However, in [13], the extremum-seeking is unconstrained, which may not be suitable in some situations. Current constrained extremum-seeking guidance techniques use adaptive methods to estimate the gradient and the Hessian of the objective function. Systems with constrained inputs, such as controller saturation are studied in [58]. Constrained extremum-seeking guidance that assumes the objective function and constraints are available for measurement are studied in [59], and state-constrained nonlinear systems using a barrier function and an adaptive method are studied in [60]. Constrained extremum-seeking in one dimension with projection is discussed in [61]. One application example includes adaptive dynamic inversion extremum-seeking guidance for constrained robotics systems [62].

The main contribution of this chapter is a linearly constrained extremum-seeking guidance algorithm enabled by a constrained Kalman filter, such as the Kalman filters described by [63, 64, 65]. In this chapter, the gain-projected Kalman filter is chosen as it is easily adapted to a projected gradient optimization method. The algorithm is demonstrated on a linear time-invariant (LTI) example, and then applied to the formation flight of two aircraft. In formation flight, the trailing aircraft reduces fuel burn by flying in the upwash of a leading aircraft wake. Figure 5.1 shows two aircraft in formation flight and the wingtip vortex of the leading aircraft. The problem of

Figure 5.1: Formation flight of two aircraft [4]. The leading aircraft's wingtip vortex is observable due to the smoke. The trailing aircraft is flying in the wake of the leading aircraft.

formation flight using an extremum-seeking guidance law has been previously studied [66, 13], and is thus a suitable benchmark problem to demonstrate the constrained extremum-seeking guidance method proposed in this chapter. The model of the aircraft is given by [67]. Only the trailing aircraft is simulated, and the aerodynamic effects of the leading aircraft is incorporated into the model using data from [68, 69].

The above contributions are shown by transitioning from the known unconstrained Kalman filter extremum-seeking method in Section 5.2, to the new constrained Kalman filter extremum-seeking formulation in Section 5.3 by considering a constrained optimization method. The extremum-seeking algorithm with the constrained Kalman filter is demonstrated on an LTI system in Section 5.4, and the formation flight example is in Section 5.5. We then consider real-world implementation challenges and closing remarks in Section 6.7.

5.2 Unconstrained Extremum-Seeking Guidance

The extremum-seeking guidance law is designed to determine the output of a system that minimizes or maximizes the objective function $J(\mathbf{z})$. Consider the system given by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \tag{5.1}$$

$$\mathbf{z} = \mathbf{h}(\mathbf{x}), \tag{5.2}$$

$$\mathbf{y} = \mathbf{c}(\mathbf{x}). \tag{5.3}$$

The measurements $\mathbf{y}$ are used by a feedback controller to drive the performance variable $\mathbf{z}$ to an extremum of $J(\mathbf{z})$, denoted as $\mathbf{z}^\star$.

In some applications, it is not possible to evaluate the objective function directly. In fact, in many situations, the specific form of the objective function, as well as the value of $\mathbf{z}^\star$, is completely unknown. For example, in the context of two aircraft in formation flight, it is generally unknown what position the trailing aircraft must be relative to the leading aircraft to minimize fuel consumption. In other words, the exact relationship between the position $\mathbf{z}$ and the fuel burn or drag $J(\mathbf{z})$ is unknown. To determine $\mathbf{z}^\star$, the desired output at time t_k, $\mathbf{z}_k^{\mathrm{d}} = \mathbf{z}^{\mathrm{d}}(t_k)$, is computed using

$$\mathbf{z}_k^{\mathrm{d}} = \mathbf{z}_k + \kappa_k \mathbf{d}_k, \tag{5.4}$$

where $\kappa_k > 0$ is the step size and where $\mathbf{d}_k$ is the descent direction. In the gradient descent method, $\mathbf{d}_k = -\mathbf{b}_k$, where $\mathbf{b}_k$ is the gradient. Newton's method uses a descent direction of $-\mathbf{M}_k^{-1}\mathbf{b}_k$, where $\mathbf{M}_k$ is the Hessian. The inner-loop controller then minimizes the difference between $\mathbf{z}_k$ and $\mathbf{z}_k^{\mathrm{d}}$. Thus as $k \to \infty$, the output $\mathbf{z}_k$ approaches $\mathbf{z}_k^\star$.

This chapter focuses on variations of gradient-based optimization methods. Since the specific form of the objective function is unknown, the gradient must be estimated. One method to estimate the gradient is to use a linear time-varying (LTV) Kalman filter, such as the one proposed in [13]. The Taylor expansion of $J(\mathbf{z})$ about $\mathbf{z}_k = \mathbf{z}(t_k)$ is

$$J(\mathbf{z}) \approx J(\mathbf{z}_k) + \mathbf{b}_k^{\mathsf{T}}(\mathbf{z} - \mathbf{z}_k), \tag{5.5}$$

where $\mathbf{b}_k$ is the gradient at $\mathbf{z}_k$, and higher-order terms have been neglected. In extremum seeking, $\mathbf{b}_k$ is *unknown*, and the measurement of the performance function $J(\mathbf{z}_{k-1})$ and $J(\mathbf{z}_z)$ is *known*. Thus, instead of using the Taylor series to approximate

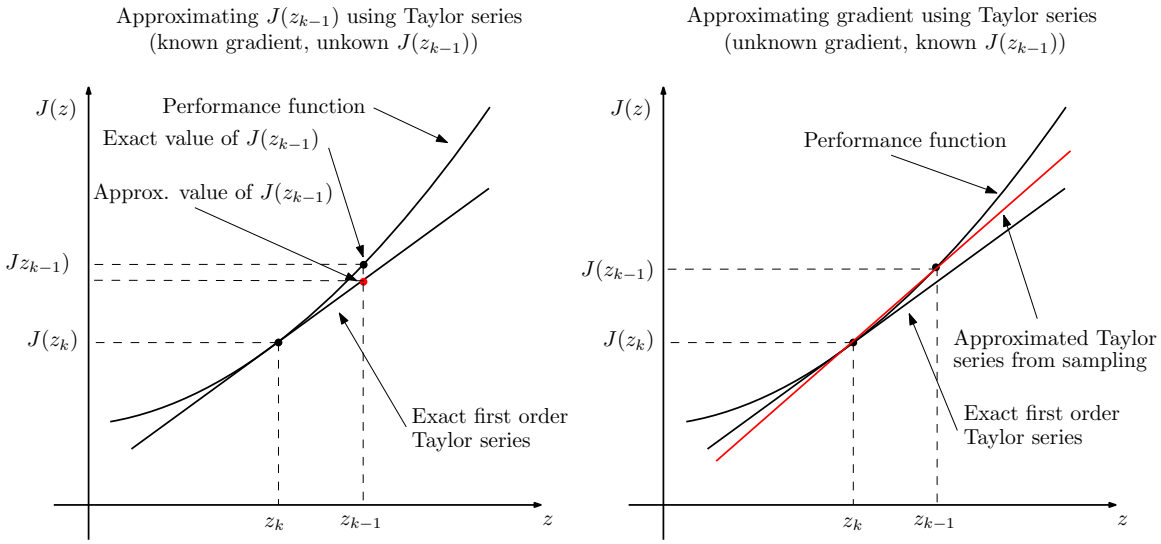


Figure 5.2: Sampling the performance function $J(z)$, where $J : \mathbb{R} \to \mathbb{R}$. On the left, the Taylor series at z_k can be used to approximate $J(z_{k-1})$. On the right, multiple measurements of z and J can be used to approximate the Taylor series at z_k.

J in the neighbourhood of $\mathbf{z}_k$, the Taylor series is used to approximate $\mathbf{b}_k$ in the the neighbourhood of $\mathbf{z}_k$, using measurements of J at $\mathbf{z}_{k-\ell}$, for $\ell = 0, \ldots, N$. The linear case is detailed in [13], but for clarity, an example for a function $J : \mathbb{R} \to \mathbb{R}$ is shown in Fig. 5.2. The exact first order Taylor series cannot be obtained due to the sampling, but if measurements are made close enough together, the difference between the approximated Taylor series and the actual Taylor series is negligible.

Evaluating (5.5) at $\mathbf{z}_{k-\ell} = \mathbf{z}(t_{k-\ell})$, $\ell = 1, \ldots, N$, and then rearranging yields,

$$\begin{bmatrix} \Delta J_k \\ \vdots \\ \Delta J_{k-N+1} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_k^\mathsf{T} \Delta \mathbf{z}_k \\ \vdots \\ \mathbf{b}_k^\mathsf{T} \Delta \mathbf{z}_{k-N+1} \end{bmatrix}, \tag{5.6}$$

where $\Delta J_{k-\ell+1} = J(\mathbf{z}_k) - J(\mathbf{z}_{k-\ell})$ and $\Delta \mathbf{z}_{k-\ell+1} = \mathbf{z}_k - \mathbf{z}_{k-\ell}$ for $\ell = 1, \ldots, N$. Equation (5.6) can be rewritten as

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{b}_k + \mathbf{v}_k, \tag{5.7}$$

where $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$ is measurement noise, and where

$$\mathbf{y}_k = \begin{bmatrix} \Delta J_k & \cdots & \Delta J_{k-N+1} \end{bmatrix}^\mathsf{T}, \tag{5.8}$$

$$\mathbf{H}_k = \begin{bmatrix} \Delta \mathbf{z}_k & \cdots & \Delta \mathbf{z}_{k-N+1} \end{bmatrix}^\mathsf{T}. \tag{5.9}$$

Algorithm 5.2 Unconstrained Extremum-Seeking Guidance $\mathbb{R}^3 \to \mathbb{R}$

for $k = k_{\text{initial}}$ to k_{final} **do**
  Measure $\mathbf{z}_{k-\ell}$ for $\ell = 0, \ldots, N$.
  Measure $J(\mathbf{z}_{k-\ell})$ for $\ell = 0, \ldots, N$.
  Construct $\mathbf{y}_k$ and $\mathbf{H}_k$ using (5.8) and (5.9).
  Predict gradient using $\hat{\mathbf{b}}_k^- = \hat{\mathbf{b}}_{k-1}$.
  Predict covariance using $\mathbf{P}_k^- = \mathbf{P}_{k-1} + \mathbf{Q}_{k-1}$.
  Determine gain $\mathbf{K}_k$ using (5.13).
  Correct gradient to obtain $\hat{\mathbf{b}}_k$ using (5.11).
  Correct covariance to obtain $\mathbf{P}_k$ using (5.12).
  Determine $\mathbf{z}_k^{\text{d}}$ using $\hat{\mathbf{b}}_k$ via (5.4), where $\kappa_k > 0$ and $\mathbf{d}_k = -\hat{\mathbf{b}}_k$.
end for

The notation $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$, $\mathbf{v}_k \in \mathbb{R}^n$, $\mathbf{R}_k \in \mathbb{R}^{n \times n}$, indicates that $\mathbf{v}_k$ is Gaussian noise with a mean of $\mathbf{0}$, and covariance $\mathbf{R}_k$. For the Kalman filter formulation, (5.7) serves as the measurement model, and

$$\mathbf{b}_k = \mathbf{b}_{k-1} + \mathbf{w}_{k-1} \tag{5.10}$$

serves as the process model, where $\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{k-1})$ is process noise. The Kalman filter estimates $\mathbf{b}_k$ using a prediction step

$$\begin{aligned}
\hat{\mathbf{b}}_k^- &= \hat{\mathbf{b}}_{k-1}, \\
\hat{\mathbf{y}}_k^- &= \mathbf{H}_k \hat{\mathbf{b}}_k^-, \\
\mathbf{P}_k^- &= \mathbf{P}_{k-1} + \mathbf{Q}_{k-1},
\end{aligned}$$

and a correction step

$$\hat{\mathbf{b}}_k = \hat{\mathbf{b}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-), \tag{5.11}$$

$$\mathbf{P}_k = (\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)^\mathsf{T} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^\mathsf{T}, \tag{5.12}$$

where $\hat{\mathbf{b}}_k$ is the estimated gradient, and $\mathbf{P}_k$ is the error covariance. The Kalman gain $\mathbf{K}_k$ is given by

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^\mathsf{T}\mathbf{V}_k^{-1}, \tag{5.13}$$

where $\mathbf{V}_k$ is the measurement error covariance given by $\mathbf{V}_k = \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^\mathsf{T} + \mathbf{R}_k$. Unconstrained extremum-seeking guidance is summarized in Algorithm 5.2.

5.3 Constrained Extremum-Seeking Guidance

Linearly constrained extremum-seeking guidance is accomplished using a constrained Kalman filter. The novelty of this chapter is the use of a constrained Kalman filter for the constrained optimization problem. The gain-projected Kalman filter from [65] is modified to constrain the gradient estimate in the same way the projected gradient optimization method constrains the gradient for constrained optimization in [30]. Providing the Kalman filter with knowledge of the constraints on the gradient can allow the filter to exploit this additional information and provide an improved gradient estimate [65].

Constrained extremum-seeking determines $\mathbf{z}^\star$ that solves

$$\min J(\mathbf{z}), \text{ s.t.} \tag{5.14}$$

$$\mathbf{a}_j^\mathsf{T}\mathbf{z} \geq \zeta_j, \ j = 1, \ldots, n_\mathrm{c}, \tag{5.15}$$

where n_c is the number of constraints. Traditional projected-gradient methods assume that $\mathbf{z}_k$ is feasible. However, we do not make this assumption and instead project $\mathbf{z}_k$ onto the feasible set to obtain $\mathbf{z}_k^\mathrm{feas}$ by solving

$$\min \frac{1}{2}(\mathbf{z}_k^\mathrm{feas} - \mathbf{z}_k)^\mathsf{T}(\mathbf{z}_k^\mathrm{feas} - \mathbf{z}_k), \text{ s.t.} \tag{5.16}$$

$$\mathbf{a}_j^\mathsf{T}\mathbf{z}_k^\mathrm{feas} \geq \zeta_j, \ j = 1, \ldots, n_\mathrm{c}. \tag{5.17}$$

The problem defined by (5.16)-(5.17) is solved by constructing the matrices $\mathbf{A}_k^\mathrm{v}$ and $\boldsymbol{\zeta}_k^\mathrm{v}$, which are the violated constraints, where

$$\mathbf{A}_k^\mathrm{v} = \begin{bmatrix} \mathbf{a}_1^\mathrm{v} & \cdots & \mathbf{a}_{n_\mathrm{v}}^\mathrm{v} \end{bmatrix}^\mathsf{T}, \quad \boldsymbol{\zeta}_k^\mathrm{v} = \begin{bmatrix} \zeta_1^\mathrm{v} & \cdots & \zeta_{n_\mathrm{v}}^\mathrm{v} \end{bmatrix}^\mathsf{T}, \tag{5.18}$$

and n_v is the number of violated constraints. The $\mathbf{a}_i^\mathrm{v}$'s and the ζ_i^v's are the violated $\mathbf{a}_j$'s and ζ_j's. The feasibility problem can thus be written as

$$\min \frac{1}{2}(\mathbf{z}_k^\mathrm{feas} - \mathbf{z}_k)^\mathsf{T}(\mathbf{z}_k^\mathrm{feas} - \mathbf{z}_k), \text{ s.t.}$$

$$\mathbf{A}_k^\mathrm{v}\mathbf{z}_k^\mathrm{feas} = \boldsymbol{\zeta}_k^\mathrm{v},$$

with the solution

$$\mathbf{z}_k^\mathrm{feas} = \mathbf{z}_k - \mathbf{A}_k^{\mathrm{v}^\mathsf{T}}(\mathbf{A}_k^\mathrm{v}\mathbf{A}_k^{\mathrm{v}^\mathsf{T}})^{-1}(\mathbf{A}_k^\mathrm{v}\mathbf{z}_k - \boldsymbol{\zeta}_k^\mathrm{v}). \tag{5.19}$$

Similar to (5.4), $\mathbf{z}_k^\mathrm{d}$ is determined by using $\mathbf{z}_k^\mathrm{feas}$ and is given by $\mathbf{z}_k^\mathrm{d} = \mathbf{z}_k^\mathrm{feas} + \kappa_k \mathbf{d}_k$,

where $\mathbf{d}_k$ must still be determined. In the projected gradient method, $\mathbf{d}_k$ is chosen to minimize the Euclidian distance between $\mathbf{d}_k$ and $-\hat{\mathbf{b}}_k$ while satisfying (5.15). In the constrained Kalman filter formulation, $\mathbf{d}_k = -\tilde{\mathbf{b}}_k$, where $\tilde{\mathbf{b}}_k$ is the constrained gradient estimate. The constrained gradient is given by minimizing the Euclidian distance between the estimated gradient $\hat{\mathbf{b}}$ and $\tilde{\mathbf{b}}_k$, while satisfying (5.15). This problem is given by

$$\begin{aligned} \min \frac{1}{2}(\hat{\mathbf{b}}_k - \tilde{\mathbf{b}}_k)^\mathsf{T}(\hat{\mathbf{b}}_k - \tilde{\mathbf{b}}_k), \quad \text{s.t.,} \\ \mathbf{z}_k^\mathrm{d} = \mathbf{z}_k^\mathrm{feas} - \kappa_k \tilde{\mathbf{b}}_k, \\ \mathbf{A}_k^\mathrm{active}\mathbf{z}_k^\mathrm{d} = \boldsymbol{\zeta}_k^\mathrm{active}, \end{aligned}$$

and $\tilde{\mathbf{b}}_k$ is given by

$$\tilde{\mathbf{b}}_k = (\mathbf{1} - \mathbf{A}_k^{\mathrm{active}^\mathsf{T}}(\mathbf{A}_k^\mathrm{active}\mathbf{A}_k^{\mathrm{active}^\mathsf{T}})^{-1}\mathbf{A}_k^\mathrm{active})\hat{\mathbf{b}}_k. \tag{5.20}$$

The matrices $\mathbf{A}_k^\mathrm{active}$ and $\mathbf{b}_k^\mathrm{active}$ represent the active constraints, and are formed in a similar manner to that of the feasibility problem. This modification to the Kalman filter does not bias the filter [70]. To finish constraining the Kalman filter, the prediction step of the Kalman filter must also use $\tilde{\mathbf{b}}_{k-1}$, where

$$\hat{\mathbf{b}}_k^- = \tilde{\mathbf{b}}_{k-1}. \tag{5.21}$$

If $\hat{\mathbf{b}}_k^- = \hat{\mathbf{b}}_{k-1}$ were used instead of (5.21), the result would be an unconstrained Kalman filter with constrained optimization. This discrepancy is the key difference between the constrained and unconstrained Kalman filter formulations for constrained extremum-seeking.

Using the projected gradient method, there is a maximum value for κ_k. For each inactive constraint,

$$\kappa_{k,i}^\mathrm{max} = -(\mathbf{a}_i^\mathsf{T}\mathbf{z}^\mathrm{feas} - b_i)\frac{1}{\mathbf{a}_i^\mathsf{T}\mathbf{d}_k}, \tag{5.22}$$

where each i is an inactive constraint. If $\kappa_{k,i}^\mathrm{max} < 0$, it is ignored. The maximum κ_k is then determined by using the minimum $\kappa_{k,i}^\mathrm{max}$. Constrained extremum-seeking guidance with a gain-projected Kalman filter is summarized in Algorithm 5.3. In Algorithm 5.3, persistent excitation is added to $\mathbf{z}_k^\mathrm{d}$ to ensure observability of $\mathbf{b}_k$, as explained in Section 5.4.

Algorithm 5.3 Constrained Extremum-Seeking Guidance $\mathbb{R}^3 \to \mathbb{R}$ with Gain-Projected Kalman Filter

for $k = k_{\text{initial}}$ to k_{final} **do**
  Measure $\mathbf{z}_{k-\ell}$ for $\ell = 0, \ldots, N$.
  Measure $J(\mathbf{z}_{k-\ell})$ for $\ell = 0, \ldots, N$.
  Construct $\mathbf{y}_k$ and $\mathbf{H}_k$ using (5.8) and (5.9).
  Predict gradient using $\hat{\mathbf{b}}_k^- = \tilde{\mathbf{b}}_{k-1}$.
  Predict covariance using $\mathbf{P}_k^- = \mathbf{P}_{k-1} + \mathbf{Q}_{k-1}$.
  Determine gain $\mathbf{K}_k$ using (5.13).
  Construct violated constraints $\mathbf{A}_k^{\text{v}}$ and $\boldsymbol{\zeta}_k^{\text{v}}$ using (5.18).
  Determine $\mathbf{z}_k^{\text{feas}}$ using (5.19).
  Determine set of active constraints $\mathbf{A}_k^{\text{active}}$.
  Correct gradient to obtain $\hat{\mathbf{b}}_k$ using (5.11).
  Constrain $\hat{\mathbf{b}}_k$ using (5.20) to obtain $\tilde{\mathbf{b}}_k$.
  Correct covariance to obtain $\mathbf{P}_k$ using (5.12).
  Note maximum step size κ_k from (5.22).
  Determine $\mathbf{z}_k^{\text{d}}$ using $\mathbf{z}_k^{\text{d}} = \mathbf{z}_k^{\text{feas}} - \kappa_k \tilde{\mathbf{b}}_k + \mathbf{w}_k^{\text{pe}}$.
end for

5.4 Linear Example

Constrained extremum-seeking is illustrated on the LTI plant given by the state-space model

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad \mathbf{z} = \mathbf{C}_1\mathbf{x}, \tag{5.23}$$

where

$$\mathbf{A} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -2 & -3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -2 & -3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \tag{5.24}$$

$$\mathbf{C}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{5.25}$$

The objective is to drive $\mathbf{z}$ to the extremum of

$$J(\mathbf{z}) = (z_1 - 0.3)^2 + 4(z_2 + 0.2)^2 + 10(z_3 - 0.1)^2. \tag{5.26}$$

The desired output $\mathbf{z}_k^{\mathrm{d}} = \mathbf{z}_k^{\mathrm{d}}(t_k)$ is given by the output of the extremum-seeking guidance algorithm from Section 5.3, which estimates the gradient of $J(\mathbf{z})$. Ten ($N = 10$) measurements are chosen and thus

$$\mathbf{H}_k = \begin{bmatrix} \Delta\mathbf{z}_k & \dots & \Delta\mathbf{z}_{k-9} \end{bmatrix}^{\mathsf{T}}. \tag{5.27}$$

Persistent excitation is added to the desired states

$$\mathbf{z}_k^{\mathrm{d}} = \mathbf{z}_k - \epsilon\hat{\mathbf{b}}_k + \mathbf{w}_k^{\mathrm{pe}}, \tag{5.28}$$

where $\mathbf{w}_k^{\mathrm{pe}}$ is zero-mean Gaussian noise with covariance $0.05\mathbf{1}$. The persistent excitation is to ensure that the estimates of the gradient converges to its desired values.

The system (5.23) is augmented with the integrator $\dot{\mathbf{e}} = \mathbf{z} - \mathbf{z}^{\mathrm{d}}$. The augmented system is

$$\begin{aligned} \dot{\mathbf{x}}^{\mathrm{a}} &= \mathbf{A}^{\mathrm{a}}\mathbf{x}^{\mathrm{a}} + \mathbf{B}_1^{\mathrm{a}}\mathbf{z}^{\mathrm{d}} + \mathbf{B}_2^{\mathrm{a}}\mathbf{u}, \\ \mathbf{z} &= \mathbf{C}_1^{\mathrm{a}}\mathbf{x}^{\mathrm{a}}, \\ \mathbf{y} &= \mathbf{C}_2^{\mathrm{a}}\mathbf{x}^{\mathrm{a}}, \end{aligned}$$

where $\mathbf{y}$ is the measurement, and where

$$\mathbf{x}^{\mathrm{a}} = \begin{bmatrix} \mathbf{x} \\ \mathbf{e} \end{bmatrix}, \quad \mathbf{A}^{\mathrm{a}} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C}_1 & \mathbf{0} \end{bmatrix}, \quad \mathbf{B}_1^{\mathrm{a}} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{1} \end{bmatrix}.$$
$$\mathbf{B}_2^{\mathrm{a}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix}, \ \mathbf{C}_1^{\mathrm{a}} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} \end{bmatrix}, \ \mathbf{C}_2^{\mathrm{a}} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}.$$

The triple $(\mathbf{A}^{\mathrm{a}}, \mathbf{B}_2^{\mathrm{a}}, \mathbf{C}_2^{\mathrm{a}})$ is observable and controllable. The control is given by the LQG controller

$$\dot{\mathbf{x}}_{\mathrm{c}} = \mathbf{A}_{\mathrm{c}}\mathbf{x}_{\mathrm{c}} + \mathbf{B}_{\mathrm{c}}\mathbf{y}, \quad \mathbf{u} = \mathbf{C}_{\mathrm{c}}\mathbf{x}_{\mathrm{c}},$$

where

$$\begin{aligned} \mathbf{A}_{\mathrm{c}} &= \mathbf{A}^{\mathrm{a}} - \mathbf{B}_2^{\mathrm{a}}\mathbf{C}_{\mathrm{c}} - \mathbf{B}_{\mathrm{c}}\mathbf{C}_2^{\mathrm{a}}, & \mathbf{B}_{\mathrm{c}} &= \mathbf{\Pi}\mathbf{C}_2^{\mathrm{a}^{\mathsf{T}}}\mathbf{R}_2^{-1}, \\ \mathbf{C}_{\mathrm{c}} &= \mathbf{R}_1^{-1}\mathbf{B}_2^{\mathrm{a}^{\mathsf{T}}}\mathbf{P}. \end{aligned}$$

The matrices $\mathbf{P}$ and $\mathbf{\Pi}$ are given by the algebraic Riccati equations

$$\mathbf{A}^{\mathrm{a}^\mathsf{T}}\mathbf{P} + \mathbf{P}\mathbf{A}^{\mathrm{a}} - \mathbf{P}\mathbf{B}_2^{\mathrm{a}}\mathbf{R}_1^{-1}\mathbf{B}_2^{\mathrm{a}^\mathsf{T}}\mathbf{P} + \mathbf{Q}_1 = \mathbf{0},$$
$$\mathbf{\Pi}\mathbf{A}^{\mathrm{a}^\mathsf{T}} + \mathbf{A}^{\mathrm{a}}\mathbf{\Pi} - \mathbf{\Pi}\mathbf{C}_2^{\mathrm{a}^\mathsf{T}}\mathbf{R}_2^{-1}\mathbf{C}_2^{\mathrm{a}}\mathbf{\Pi} + \mathbf{Q}_2 = \mathbf{0}.$$

The extremum of (5.26) is $\mathbf{z}^\star = \begin{bmatrix} 0.3 & -0.2 & 0.1 \end{bmatrix}^\mathsf{T}$. The performance variables z_i reach the extremal values $z_i^\star$, where $i = 1, 2, 3$. The problem is constrained such that $-0.2 \leq z_i \leq 0.2$ for $i = 1, 2, 3$. These constraints are chosen so that one element of the minimizer lies within the constrained set, one element of the minimizer lies on the constrained set, and one element of the minimizer lies outside the constrained set. The results of the simulation are shown in Fig. 5.3. As expected, each output z_i converges to either to $z_i^\star$ or to the constraint. Notice how z_1 does not hit the unconstrained $z_1^\star$ because it lies outside the constraint set, and as such, hits the constraint boundary. The performance variable z_2 is still able to reach $z_2^\star$ because $z_2^\star$ lies exactly on the constraint boundary. The performance variable z_3 is also able to reach $z_3^\star$ because $z_3^\star$ lies within the constraint. Notice also that z_1 and z_2 violate the constraints just slightly owing to the presence of noise that persistently excites the system.

5.5 Application to Formation Flight of Two Aircraft

The following notation is used in this section. For a column matrix $\mathbf{e}(t) \in \mathbb{R}^n$, let the function $\mathbf{\Gamma}(\mathbf{e}) : \mathbb{R}^n \to \mathbb{R}^n$ be defined as

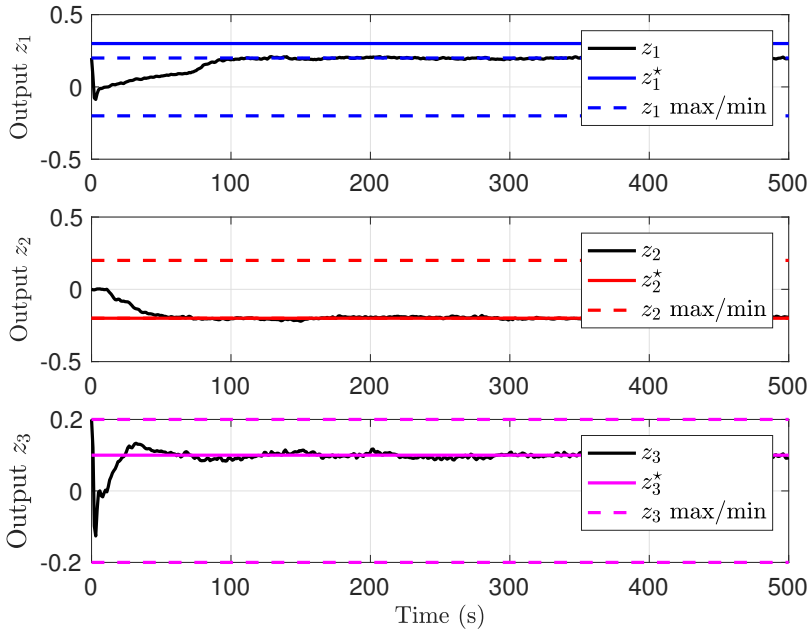
$$\mathbf{\Gamma}(\mathbf{e}) = \mathbf{K}_{\mathrm{p}}\mathbf{e} + \mathbf{K}_{\mathrm{d}}\dot{\mathbf{e}} + \mathbf{K}_{\mathrm{i}}\int_0^t \mathbf{e}\ \mathrm{d}t, \tag{5.29}$$

where $\mathbf{K}_{\mathrm{p}} = \mathbf{K}_{\mathrm{p}}^\mathsf{T} > 0$, $\mathbf{K}_{\mathrm{d}} = \mathbf{K}_{\mathrm{d}}^\mathsf{T} \geq 0$, and $\mathbf{K}_{\mathrm{i}} = \mathbf{K}_{\mathrm{i}}^\mathsf{T} \geq 0$. The function $\mathbf{\Gamma}(\cdot)$ is denoted Γ when $\Gamma : \mathbb{R} \to \mathbb{R}$.

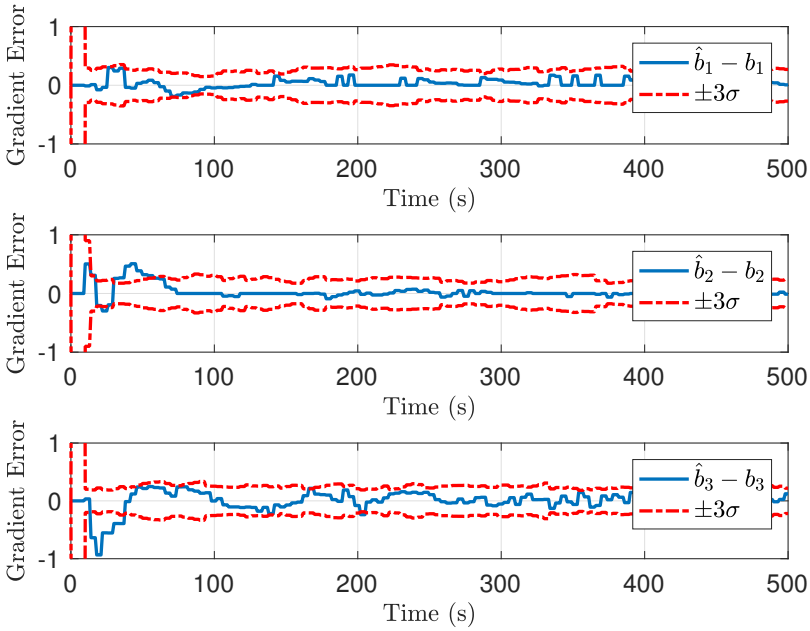
5.5.1 Aircraft Kinematics

Let $\mathcal{F}_a$ be the inertial frame, $\mathcal{F}_v$ is the wind frame, and $\mathcal{F}_b$ is the body frame of the aircraft. The wind frame is defined such that $\underrightarrow{v}^1$ is aligned with the velocity vector. A $3-2-1$ Euler angle sequence

$$\mathcal{F}_a \xrightarrow[3]{\psi} \xrightarrow[2]{\theta} \xrightarrow[1]{\phi} \mathcal{F}_b \tag{5.30}$$



(a) Output of system versus time with bounds of ± 0.2.



(b) Gradient estimates with $\pm 3\sigma$ bounds.

Figure 5.3: 3D extremum-seeking guidance simulation with gradient descent. Objective function is a smooth function, noise has been added, and constraints are to keep each state $|\cdot| \leq 0.2$.

describes the attitude of the aircraft's body frame $\mathcal{F}_b$ relative to the inertial frame $\mathcal{F}_a$, where ψ is the yaw angle, θ is the pitch angle, and ϕ is the roll angle. In addition,

the relations

$$\mathcal{F}_a \xrightarrow[3]{\chi} \xrightarrow[2]{\gamma} \xrightarrow[1]{\mu} \mathcal{F}_v \xrightarrow[3]{-\beta} \xrightarrow[2]{\alpha} \mathcal{F}_b, \tag{5.31}$$

describe the aircraft's attitude with an intermediate relative wind frame $\mathcal{F}_v$, where χ is the heading, γ is the flight path angle, μ is the bank angle, β is the sideslip angle, and α is the angle of attack. These kinematics assume zero wind. The negative sign associated with the sideslip is included to satisfy the convention adopted by most of the aircraft literature.

The relations defined by (5.30)-(5.31) allow us to define

$$\mathbf{q}^{va} = \begin{bmatrix} \mu & \gamma & \chi \end{bmatrix}^{\mathsf{T}}, \qquad \mathbf{q}^{bv} = \begin{bmatrix} \alpha & \beta \end{bmatrix}^{\mathsf{T}},$$
$$\mathbf{q}^{ba} = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^{\mathsf{T}},$$

with associated parameterized DCMs and angular velocities given by

$$\mathbf{C}_{va} = \mathbf{C}_1(\mu)\mathbf{C}_2(\gamma)\mathbf{C}_3(\chi), \qquad \boldsymbol{\omega}_v^{va} = \mathbf{S}_v^{va}(\mathbf{q}^{va})\dot{\mathbf{q}}^{va},$$
$$\mathbf{C}_{bv} = \mathbf{C}_2(\alpha)\mathbf{C}_3(-\beta), \qquad \boldsymbol{\omega}_b^{bv} = \mathbf{S}_b^{bv}(\mathbf{q}^{bv})\dot{\mathbf{q}}^{bv},$$
$$\mathbf{C}_{ba} = \mathbf{C}_1(\phi)\mathbf{C}_2(\theta)\mathbf{C}_3(\psi), \qquad \boldsymbol{\omega}_b^{ba} = \mathbf{S}_b^{ba}(\mathbf{q}^{ba})\dot{\mathbf{q}}^{ba}.$$

The mapping matrices between angle rates and angular velocities are [14]

$$\mathbf{S}_v^{va}(\mathbf{q}^{va}) = \begin{bmatrix} \mathbf{1}_2 & \mathbf{C}_2(\gamma)\mathbf{1}_3 \end{bmatrix}, \tag{5.32}$$
$$\mathbf{S}_b^{bv}(\mathbf{q}^{bv}) = \begin{bmatrix} \mathbf{1}_3 & -\mathbf{C}_2(\alpha)\mathbf{1}_2 & \mathbf{C}_2(\alpha)\mathbf{C}_3(-\beta)\mathbf{1}_1 \end{bmatrix}, \tag{5.33}$$
$$\mathbf{S}_b^{ba}(\mathbf{q}^{ba}) = \begin{bmatrix} \mathbf{1}_1 & \mathbf{C}_1(\phi)\mathbf{1}_2 & \mathbf{C}_1(\phi)\mathbf{C}_2(\theta)\mathbf{1}_3 \end{bmatrix}. \tag{5.34}$$

5.5.2 Flight Control Law

The position of the aircraft resolved in $\mathcal{F}_a$ is $\mathbf{r}_a^{cw}$, where c is the centre of mass of the aircraft, and where w is an unforced particle that can be though of as the origin of the inertial frame. The leading aircraft's centre of mass is at point ℓ, and the position of the trailing aircraft relative to the leading aircraft is given by $\mathbf{r}_a^{c\ell}$. The velocity of the aircraft with respect to $\mathcal{F}_a$ is given by $\mathbf{v}_a^{cw/a}$, and the aircraft's dynamics are given

by

$$\begin{aligned}\dot{\mathbf{v}}_a^{cw/a} &= \mathbf{f}^r(\mathbf{r}_a^{cw}, \mathbf{v}_a^{cw/a}, \mathbf{q}^{ba}, \boldsymbol{\omega}_b^{ba}, \mathbf{r}_a^{c\ell}) \\ &\quad + \mathbf{G}^r(\mathbf{r}_a^{cw}, \mathbf{v}_a^{cw/a}, \mathbf{q}^{ba}, \boldsymbol{\omega}_b^{ba})\mathbf{u}, \qquad (5.35)\\ \dot{\boldsymbol{\omega}}_b^{ba} &= \mathbf{f}^\omega(\mathbf{q}^{ba}, \boldsymbol{\omega}_b^{ba}, \mathbf{r}_a^{c\ell}) + \mathbf{G}^\omega(\mathbf{q}^{ba}, \boldsymbol{\omega}_b^{ba}, \mathbf{r}_a^{c\ell})\mathbf{u}^\delta, \qquad (5.36)\end{aligned}$$

where $\mathbf{u} = [u^{\mathrm{t}} \;\; \mathbf{u}^{\delta^\mathsf{T}}]^\mathsf{T}$ are the control inputs, u^{t} is the thrust produced by the engines, and $\mathbf{u}^\delta$ are the control surface deflections. Equation (5.35) can be used to derive an airspeed equation

$$\begin{aligned}\dot{v}_a^{cw/a} = f^v(\mathbf{r}_a^{cw}, \mathbf{v}_a^{cw/a}, \mathbf{q}^{ba}, \boldsymbol{\omega}_b^{ba}, \mathbf{r}_a^{c\ell}, \mathbf{u}^\delta) \\ + G^v(\mathbf{r}, \dot{\mathbf{r}}, \mathbf{q}^{ba}, \boldsymbol{\omega}_b^{ba})u^{\mathrm{t}},\end{aligned} \qquad (5.37)$$

where $v_a^{dw/a} = \left\| \mathbf{v}_a^{cw/a} \right\|$, and $\|\cdot\|$ is the Euclidian norm.

The flight control laws are based on nonlinear dynamic inversion, inspired by [71]. This control law is designed to track a desired position $\mathbf{r}_a^{dw}$, which is the output of the extremum-seeking guidance law.

First, a control law is given by $\mathbf{v}_a^{dw/a} = \boldsymbol{\Gamma}(\mathbf{r}_a^{dw} - \mathbf{r}_a^{cw})$. To track $\mathbf{v}_a^{dw/a}$, the desired flight path and heading angles are required. Similar to the relation between $\mathcal{F}_a$ and $\mathcal{F}_b$ in (5.30), the desired frame $\mathcal{F}_d$ is given by

$$\mathcal{F}_a \xrightarrow[3]{\chi_{\mathrm{d}}} \xrightarrow[2]{\gamma_{\mathrm{d}}} \xrightarrow[1]{\mu_{\mathrm{d}}} \mathcal{F}_{v_{\mathrm{d}}} \xrightarrow[3]{-\beta_{\mathrm{d}}} \xrightarrow[2]{\alpha_{\mathrm{d}}} \mathcal{F}_d, \qquad (5.38)$$

where the angles with the subscript "d" signifies "desired". Using (5.38), and the desired velocity, the desired flight path and heading angles are given by

$$\gamma_{\mathrm{d}} = \sin^{-1}\left(\frac{-\mathbf{1}_3^\mathsf{T}\mathbf{v}_a^{dw/a}}{v^{dw/a}}\right), \quad \chi_{\mathrm{d}} = \sin^{-1}\left(\frac{\mathbf{1}_2^\mathsf{T}\mathbf{v}_a^{dw/a}}{\mathbf{1}_1^\mathsf{T}\mathbf{v}_a^{dw/a}}\right).$$

The desired flight path and heading angle rates are

$$\dot{\gamma}_{\mathrm{d}} = \Gamma(\gamma_{\mathrm{d}} - \gamma), \quad \dot{\chi}_{\mathrm{d}} = \Gamma(\chi_{\mathrm{d}} - \chi).$$

The desired bank angle is approximated as $\mu_{\mathrm{d}} = \tan^{-1}\left(\dot{\chi}_{\mathrm{d}} v^{dw/a}/g\cos(\theta)\right)$, and the desired sideslip angle is, $\beta_{\mathrm{d}} = 0$, which should result in coordinated flight. The desired angle of attack is given by $\alpha_{\mathrm{d}} = \Gamma(\gamma_{\mathrm{d}} - \gamma)$, and the desired angle of bank, sideslip,

angle of attack rates are given by

$$\dot{\alpha}_{\mathrm{d}} = \Gamma(\alpha_{\mathrm{d}} - \alpha), \quad \dot{\mu}_{\mathrm{d}} = \Gamma(\mu_{\mathrm{d}} - \mu), \quad \dot{\beta}_{\mathrm{d}} = \Gamma(\beta_{\mathrm{d}} - \beta).$$

Similar to the relation between $\mathcal{F}_a$ and $\mathcal{F}_b$ in (5.31), the desired frame $\mathcal{F}_d$ is given by

$$\mathcal{F}_a \xrightarrow[3]{\chi_{\mathrm{d}}} \xrightarrow[2]{\gamma_{\mathrm{d}}} \xrightarrow[1]{\mu_{\mathrm{d}}} \mathcal{F}_{v_{\mathrm{d}}} \xrightarrow[3]{-\beta_{\mathrm{d}}} \xrightarrow[2]{\alpha_{\mathrm{d}}} \mathcal{F}_d, \tag{5.39}$$

Using (5.39), the desired angular velocity is $\boldsymbol{\omega}_b^{da} = \mathbf{C}_{bd}(\mathbf{S}_d^{dv_{\mathrm{d}}}\dot{\mathbf{q}}^{dv_{\mathrm{d}}} + \mathbf{C}_{dv_{\mathrm{d}}}\mathbf{S}_{v_{\mathrm{d}}}^{v_{\mathrm{d}}a}\dot{\mathbf{q}}^{v_{\mathrm{d}}a})$, where

$$\mathbf{q}^{dv_{\mathrm{d}}} = \begin{bmatrix} \alpha_{\mathrm{d}} & \beta_{\mathrm{d}} \end{bmatrix}^{\mathsf{T}}, \quad \mathbf{q}^{v_{\mathrm{d}}a} = \begin{bmatrix} \mu_{\mathrm{d}} & \gamma_{\mathrm{d}} & \chi_{\mathrm{d}} \end{bmatrix}^{\mathsf{T}}.$$

The matrices $\mathbf{S}_d^{dv_{\mathrm{d}}}$ and $\mathbf{S}_{v_{\mathrm{d}}}^{v_{\mathrm{d}}a}$ can be determined in a similar manner as (5.32)-(5.33). The desired angular acceleration is given by $\dot{\boldsymbol{\omega}}_b^{da} = \boldsymbol{\Gamma}(\boldsymbol{\omega}_b^{da} - \boldsymbol{\omega}_b^{ba})$, and control surface deflections are

$$\mathbf{u}^{\delta} = \mathbf{G}^{\omega^{-1}}(\dot{\boldsymbol{\omega}}_b^{da} - \mathbf{f}^{\omega}). \tag{5.40}$$

Lastly, using (5.37), the thrust is given by

$$u^{\mathrm{t}} = G^{v^{-1}}(\dot{v}^{dw/a} - f^{v}), \tag{5.41}$$

where desired speed is $v^{dw/a} = ||\mathbf{v}_a^{dw/a}||$, and the desired acceleration is $\dot{v}^{dw/a} = \Gamma(v^{dw/a} - v^{cw/a})$. The control inputs to the aircraft are the combination of (5.40) and (5.41).

5.5.3 Simulation Results

The constrained Kalman filter-based extremum-seeking guidance runs at 0.2 Hz, with 10 measurement collected at 1 Hz. The autopilot runs at 10 Hz, and the aircraft is simulated in continuous time. The leading aircraft is flying at a speed of Mach 0.56 at an altitude of 25,000 ft, which corresponds to the condition 1 flight conditions from [68, 69]. The desired position relative to the leading aircraft is given by $\mathbf{r}_a^{d\ell}$. When simulated with the extremum-seeking guidance law, the performance variables are $\mathbf{z} = \mathbf{r}_a^{c\ell}$, $\mathbf{z}^{\mathrm{d}} = \mathbf{r}_a^{d\ell}$. The constraint $[-51 \;\; 0 \;\; -11]^{\mathsf{T}}$ m $\leq \mathbf{r}_a^{d\ell} \leq [-28 \;\; 10 \;\; 11]^{\mathsf{T}}$ m defines a zone where empirical drag reduction data is available for simulation, where formation flight has the greatest effect on drag reduction, and which maintains a distance from the leading aircraft. The drag reduction as a function of $\mathbf{r}_a^{c\ell}$ is shown in Fig. 5.4a, where it can be seen that a 20% drag reduction is possible. Change in drag reduction

is much less pronounced in the $r_{a,1}^{c\ell}$ direction, and thus variation in z_1^{d} has less of an effect. For these simulations, the measured performance $J(\mathbf{z})$ is the change in drag shown in Fig. 5.4a, and the relationship between $\mathbf{z}$ and $J(\mathbf{z})$ is much more complex here than in (5.26).

No external wind gusts are introduced into the simulation, and the autopilot of the second aircraft has no knowledge of the effect of the wake generated by the leading aircraft. The simulation results of the extremum-seeking algorithm, in tandem with the autopilot and the aircraft, are shown in Fig. 5.4 and Fig. 5.5. Figure 5.4b shows that $\mathbf{z}^{\mathrm{d}}$ mostly lies within the constraints as sinusoidal excitation has been added to $\mathbf{z}^{\mathrm{d}}$ to improve the gradient estimates. As expected, $\mathbf{z}^{\mathrm{d}}$ approaches $\mathbf{z}^{\star}$.

The gradient error estimates are shown in Fig. 5.5a. This plot shows the difference between $\tilde{\mathbf{b}}$ and the actual gradient $\mathbf{b}$ projected onto the constrained set, denoted as $\mathbf{b}^{\mathrm{c}}$. This is done so that when the extremum-seeking guidance law is at an active constraint, the expected error will be zero. The gradient estimates remain within the $\pm 3\sigma$ bounds, which indicates that the Kalman filter remains consistent. As the system stabilizes, the $\pm 3\sigma$ bounds increase since the measurements provided to the Kalman filter are closer together, and $\Delta\mathbf{z}_{k-\ell+1}$, $\ell = 1, \ldots, N$ becomes closer to zero.

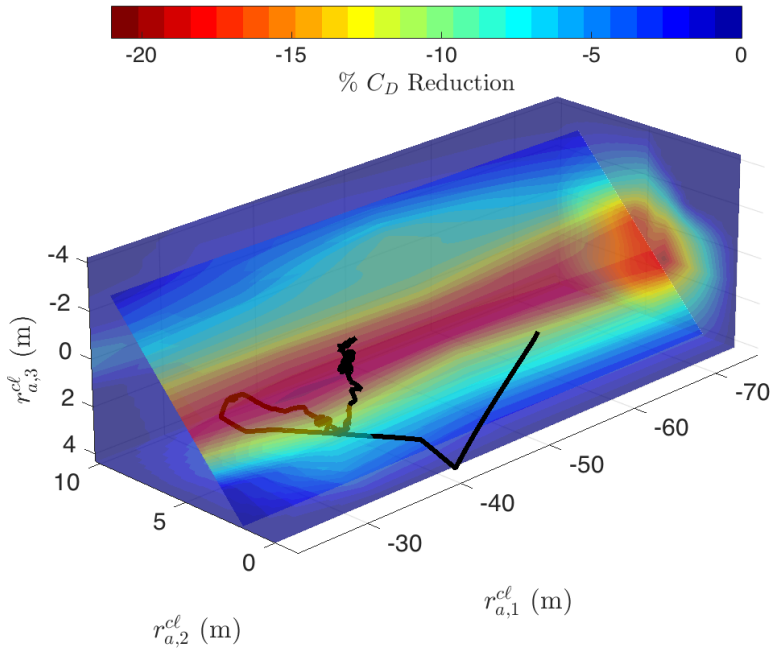
The error states of the aircraft are shown in Fig. 5.5b. These states are important because the extremum-seeking guidance law uses the current output $\mathbf{z}_k$ to obtain $\mathbf{z}_k^{\mathrm{d}}$. If the autopilot is unable to drive $\mathbf{z}_k$ to $\mathbf{z}_k^{\mathrm{d}}$, then $\mathbf{z}_k^{\mathrm{d}}$ will be unable to converge to $\mathbf{z}^{\star}$. As an example, z_3^{d} remains at the upper limit of the constrained set for most of the simulation since the autopilot slowly corrects for the error in the $r_{a,3}^{c\ell}$ direction. The large amplitude oscillations in the plot are due to the autopilot. In addition, these oscillations appear more severe than they are because of the scale of the plot. The scale is chosen to highlight the steady state of the aircraft.

5.6 Closing Remarks

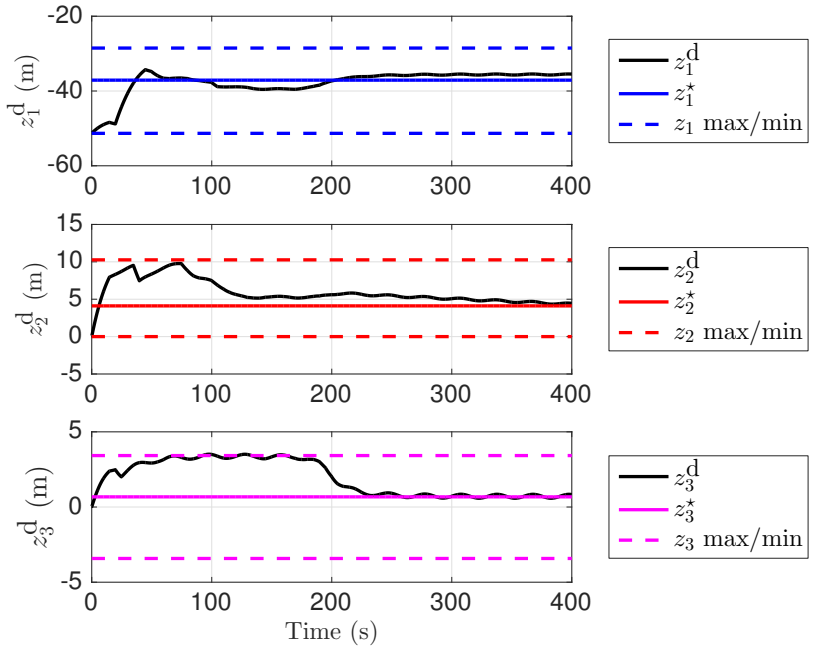
This chapter presents an extremum-seeking controller using a constrained Kalman filter for extremum-seeking guidance on $\mathbb{R}^3$. A linear plant is presented and the developed technique is applied to a formation flight problem. Although no disturbances were present in the formation flight simulation, this is not a large concern because these disturbances would affect the autopilot rather than the extremum-seeking guidance algorithm. That being said, the extremum-seeking guidance method directly used the percent change of the drag coefficient. In practice, other metrics such as fuel flow or rolling moment could be used. However, since changes in these measurements

take time, the extremum-seeking controller should be run at a much lower frequency.

In previous extremum-seeking work, the Hessian was also estimated for use with a Newton's method approach to optimization. In this work, estimating the Hessian was a challenging task that did not yield promising results. Exact reasons why the Kalman filter had trouble estimating the Hessian were not pursued. In the future, determining these reasons and experimenting with other types of filters, including sigma point filters, can provide methods to estimate the Hessian associated with the performance function.

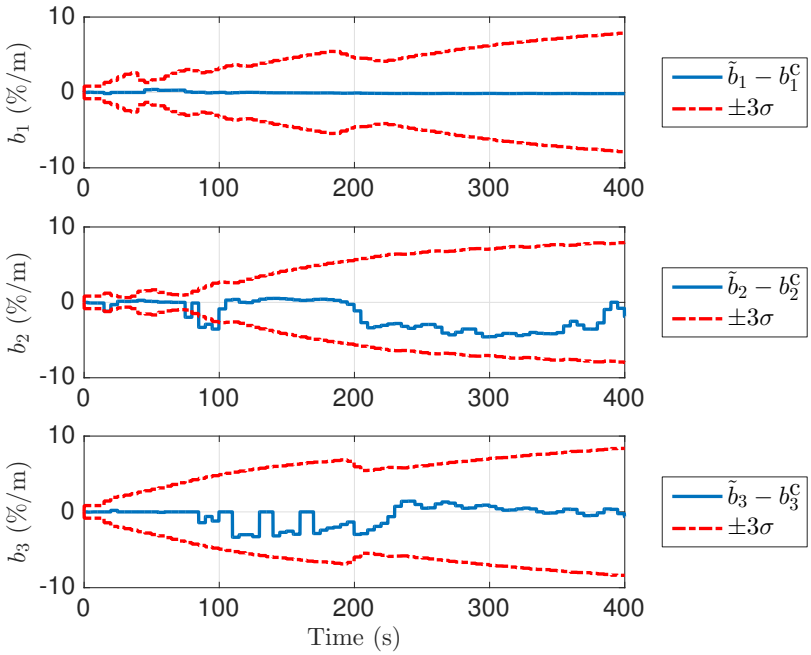


(a) Percent change of C_D plotted as a function of $\mathbf{r}_a^{c\ell}$. The solid black line represents $\mathbf{z}^{\mathrm{d}}$ for the duration of the simulation.

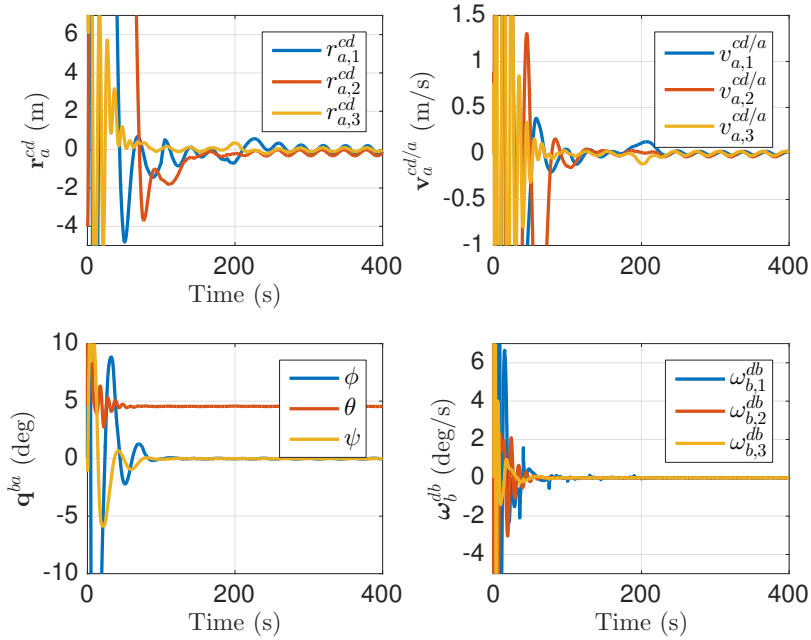


(b) Desired position $\mathbf{z}^{\mathrm{d}} = \mathbf{r}_a^{d\ell}$ and position constraints.

Figure 5.4: Constrained extremum-seeking using constrained Kalman filter applied to formation flight control.



(a) Gradient estimates with $\pm 3\sigma$ bounds.



(b) Velocity, position, and angular velocity errors. Attitude parameterized using (5.30) also shown.

Figure 5.5: Estimated gradient and state error of constrained extremum-seeking using constrained Kalman filter applied to formation flight control.

Chapter 6

Extremum-Seeking Guidance on $SO(3)$

6.1 Introduction

Consider extremum-seeking algorithms that attempt to maximize an objective function $J : D \to \mathbb{R}$, where $J = J(\mathbf{z})$, and where $\mathbf{z} \in D$ is the performance variable of the plant. There are three main approaches to extremize the objective function J on the the restricted domain $D \subset \mathbb{R}^{n \times m}$, $n, m \in \mathbb{N}$, depending on the nature of the constraints. These methods include a Lagrangian approach [72], a barrier function approach [60, 58, 59], and an optimization on manifolds approach [73, 74]. Each method has its own strengths depending on the nature of the given constraints. In this chapter, extremum seeking on the manifold $SO(3)$ and on a constrained subset of $SO(3)$ is discussed. In particular, extremum seeking on $SO(3)$ is enabled by merging a Lagrangian-based approach and a manifold-based gradient optimization algorithm.

Gradient-based extremum-seeking guidance methods require estimation of the gradient, a process that usually falls in two categories: a parameter estimation approach [8, 75, 61], and a Kalman filter approach [13]. A Kalman filter is advantageous since it is able to filter measurement noise while providing the gradient estimate, and is therefore the approach used in this chapter. For constrained subsets of $SO(3)$, a constrained gradient optimization method is used, which can be enabled by a constrained filtering method. In this chapter, attitude inclusion and exclusion zones are written as linear inequality constraints, which are a function of the gradient of the performance function. The linear attitude constraints are constructed at each time step, and a constraint to ensure the linearization remains valid can then be used. The idea of sequential linearization and then ensuring small step sizes are small is simi-

lar to that of [53]. A constrained gradient is then estimated using a gain-projected Kalman filter [70], which requires linear constraints. This filter projects the posteriori gradient estimate onto a constrained set so that the desired trajectory does not violate the inclusion or exclusion zones. Further discussion of filtering with equality an inequality state constraints can be found in [65].

In general, a Kalman filter determines the best state estimate in a minimum mean square error sense. This is an optimization problem that can be formulated as a semidefinite programming (SDP) problem using linear matrix inequalities (LMIs), which can be solved efficiently using interior-point methods [31]. Using an LMI approach to Kalman filtering, state constraints or set-membership can easily be introduced. An LMI-based approach is used for set-membership filtering for equality, inequality, and linearized nonlinear constraints in [76, 77]. Unlike the gain-projected Kalman filter, the LMI-based filter modifies the Kalman gain so that the state estimate is constrained during the correction step, and thus no extra constraining step is needed to ensure the state estimates satisfy the constraints. Using [76, 77] as inspiration, a novel LMI-based Kalman filter is introduced to estimate the constrained gradient for the extremum-seeking guidance problem. The filter has a different structure, different assumptions on the system's characteristics, and a different derivation than in [76], and can handle both the linear inequality and norm constraints placed on the gradient estimates. These attributes make the LMI-based filter well suited for estimating the constrained gradient in extremum-seeking applications.

There are several contributions in this chapter that together realize extremum seeking on $SO(3)$ and on a constrained subset of $SO(3)$. First, the relationship between the gradient of a function on $SO(3)$ and a Taylor series expansion of the same function is elucidated. The extremum-seeking algorithm hinges on this relationship. Second, inclusion and exclusion zones on $SO(3)$ are reformulated as linear inequality constraints that are easily embedded into a constrained optimization problem. To be clear, it is the manner the constraints are reformulated into a form suitable for constrained optimization that is the contribution, not the fashion that the constraints are initially posed. The initial statement of the attitude constraints is identical to other work, such as in [51, 78]. Third, the gradient of an unknown constrained performance function is estimated using a gain-projected Kalman filter and a novel LMI-based Kalman filter. The estimated gradient is used in a gradient ascent optimization algorithm. Fourth, the proposed extremum-seeking technique is demonstrated on a spacecraft attitude guidance problem. Specifically, the proposed extremum-seeking guidance law is used to determine a desired attitude that maximizes an objective

function, and a feedback control law minimizes the error between the actual attitude and the desired attitude of the spacecraft. For purposes of this chapter, the spacecraft's attitude is assumed to be known exactly, but similar control techniques can be applied if the attitude were estimated from sun sensors, star trackers, magnetometers, etc.

The first contribution is developed in Section 6.2 once pertinent notation and concepts from differential geometry are reviewed. Section 6.3 gives an overview of gradient and projected gradient ascent optimization on $SO(3)$. Unconstrained and constrained extremum-seeking on $SO(3)$ using gradient and projected gradient ascent are shown in Section 6.4 and Section 6.5, respectively. Application of the developed extremum-seeking guidance algorithms to a spacecraft equipped with a patch antenna and three reaction wheels is in Section 6.6, and closing remarks are in Section 6.7.

6.2 Mathematical Preliminaries

This section briefly introduces necessary notation, and then reviews important concepts from differential geometry. The tangent space of $SO(3)$ and the gradient of a function $f : SO(3) \to \mathbb{R}$ is explained, as well as the relationship between a first order Taylor series expansion of f and the gradient of f that lies in the tangent space of $SO(3)$. The explanation of the gradient is necessary because a gradient ascent optimization method is used for the extremum-seeking guidance algorithm.

6.2.1 Useful Identities

Proposition 6.1. *Let* $\mathbf{M} \in \mathbb{R}^{n\times n}$*, and* $\mathbf{\Omega} \in \mathfrak{so}(n)$*, then*

$$\operatorname{tr}(\mathbf{M}\mathbf{\Omega}) = \operatorname{tr}(\boldsymbol{\mathcal{P}}_a(\mathbf{M})\mathbf{\Omega}). \tag{6.1}$$

Proof. Expand the right side using the definition of $\boldsymbol{\mathcal{P}}_a(\mathbf{M})$ to obtain the left side. □

Proposition 6.2. *Let* $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$*, then*

$$\operatorname{tr}(-\mathbf{u}^\times \mathbf{v}^\times) = 2\mathbf{u}^\mathsf{T}\mathbf{v}. \tag{6.2}$$

Proof. Use the identity defined by Eq. (2.56b) in [34, p. 29] in (6.2). □

Proposition 6.3 (Chain Rule)**.** *Given a function* $f(\mathbf{X}) : \mathbb{R}^{n\times n} \to \mathbb{R}$ *and a function*

$\mathbf{X}(y) : \mathbb{R} \to \mathbb{R}^{n\times n}$, *the derivative of* $f(\mathbf{X}(y))$ *with respect to* y *is given by*

$$\frac{df}{dy} = \operatorname{tr}\left[\left(\frac{df(\mathbf{X})}{d\mathbf{X}}\right)\left(\frac{d\mathbf{X}(y)}{dy}\right)\right]. \tag{6.3}$$

Proof. See Appendix B.2. □

In Section 6.3, continuous functions of the form $f : SO(3) \to \mathbb{R}$ are considered. The manifold $SO(3)$ is compact and, as such, the function f admits an extremum according to Theorem 2.7.

6.2.2 Manifolds and Tangent Spaces

The manifold $SO(3)$ is an embedded submanifold of $\mathbb{R}^{3\times3}$ [79, pp. 41-42]. The embedding space of $SO(3)$, denoted as $\overline{SO(3)}$, is the manifold $\mathbb{R}^{3\times3}$. The tangent space to $SO(3)$ at $\mathbf{C} \in SO(3)$ is [79, p. 42]

$$T_{\mathbf{C}}SO(3) = \{\boldsymbol{\Omega}\mathbf{C} : \boldsymbol{\Omega} \in \mathfrak{so}(3)\}. \tag{6.4}$$

A manifold whose tangent spaces are equipped with a smoothly varying inner product is called a Riemannian manifold. The manifold $\mathbb{R}^{3\times3}$ has the inner product given by

$$\langle \mathbf{A}, \mathbf{B}\rangle = \operatorname{tr}(\mathbf{A}^{\mathsf{T}}\mathbf{B}), \tag{6.5}$$

where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{3\times3}$. The submanifolds and tangent manifolds of $\mathbb{R}^{3\times3}$ inherit this inner product. Thus, $SO(3)$ becomes a Riemmannian manifold when $T_{\mathbf{C}}SO(3)$ is equipped with the inner product defined by (6.5). The inner product is important since it is used to define the gradient of a function.

The orthogonal complement of $T_{\mathbf{C}}SO(3)$ is denoted as $(T_{\mathbf{C}}SO(3))^{\perp}$. An element of $\mathbf{Y} \in (T_{\mathbf{C}}SO(3))^{\perp}$ is defined such that for all $\mathbf{X} \in T_{\mathbf{C}}SO(3)$, $\langle \mathbf{X}, \mathbf{Y}\rangle = 0$. The space $(T_{\mathbf{C}}SO(3))^{\perp}$ is given by [79, p. 48]

$$(T_{\mathbf{C}}SO(3))^{\perp} = \{\mathbf{S}\mathbf{C} : \mathbf{S} \in S\mathbb{R}^3\}. \tag{6.6}$$

Proposition 6.4. *Let* $T_{\mathbf{C}}\overline{SO(3)}$ *be the tangent space of* $\overline{SO(3)}$ *at* $\mathbf{C}$*. An element* $\mathbf{Z} \in T_{\mathbf{C}}\overline{SO(3)}$ *can be decomposed into the sum of an element in* $T_{\mathbf{C}}SO(3)$ *and in* $(T_{\mathbf{C}}SO(3))^{\perp}$ *by*

$$\mathbf{Z} = \boldsymbol{\mathcal{P}}_{\mathbf{C}}(\mathbf{Z}) + \boldsymbol{\mathcal{P}}_{\mathbf{C}}^{\perp}(\mathbf{Z}), \tag{6.7}$$

where $\boldsymbol{\mathcal{P}}_{\mathbf{C}}$ *and* $\boldsymbol{\mathcal{P}}_{\mathbf{C}}^{\perp}$ *are projection operators that define the mapping* $\boldsymbol{\mathcal{P}}_{\mathbf{C}} : T_{\mathbf{C}}\overline{SO(3)} \to$

$T_{\mathbf{C}}SO(3)$ *and* $\boldsymbol{\mathcal{P}}_{\mathbf{C}}^{\perp} : T_{\mathbf{C}}\overline{SO(3)} \to (T_{\mathbf{C}}SO(3))^{\perp}$. *These projections are given by [79, p. 48]*

$$\boldsymbol{\mathcal{P}}_{\mathbf{C}}(\mathbf{Z}) = \boldsymbol{\mathcal{P}}_a(\mathbf{Z}\mathbf{C}^{\mathsf{T}})\mathbf{C}, \tag{6.8}$$

$$\boldsymbol{\mathcal{P}}_{\mathbf{C}}^{\perp}(\mathbf{Z}) = \boldsymbol{\mathcal{P}}_s(\mathbf{Z}\mathbf{C}^{\mathsf{T}})\mathbf{C}. \tag{6.9}$$

Proof. Comparing (6.8)-(6.9) to (6.4) and (6.6) shows that $\boldsymbol{\mathcal{P}}_{\mathbf{C}}(\mathbf{Z}) \in T_{\mathbf{C}}SO(3)$ and $\boldsymbol{\mathcal{P}}_{\mathbf{C}}^{\perp}(\mathbf{Z}) \in (T_{\mathbf{C}}SO(3))^{\perp}$. Evaluating (6.7) using (6.8) and (6.9) completes the proof. □

6.2.3 Gradients on $SO(3)$ from Definition

Let $f : SO(3) \to \mathbb{R}$ be a continuous function, where $SO(3)$ is a Riemmanian manifold equipped with the inner product given by (6.5). The gradient of f at $\mathbf{C}$, denoted by $\boldsymbol{\nabla} f(\mathbf{C})$, is defined as the unique element of $T_{\mathbf{C}}SO(3)$ that for all $\boldsymbol{\Xi} \in T_{\mathbf{C}}SO(3)$, satisfies

$$\langle \boldsymbol{\nabla} f(\mathbf{C}), \boldsymbol{\Xi} \rangle = \left. \frac{\mathrm{d}(f(\boldsymbol{\Gamma}(\epsilon)))}{\mathrm{d}\epsilon} \right|_{\epsilon=0}, \tag{6.10}$$

where $\boldsymbol{\Gamma}(\epsilon) \in SO(3)$, $\boldsymbol{\Gamma}(0) = \mathbf{C}$, and where $\boldsymbol{\Gamma}'(0) = [\mathrm{d}\boldsymbol{\Gamma}(\epsilon)/\mathrm{d}\epsilon]_{\epsilon=0} = \boldsymbol{\Xi}$ [79, p. 46]. Note that the derivative of $e^{t\mathbf{A}}$, $\mathbf{A} \in \mathbb{R}^{n\times n}$, with respect to t is $\mathrm{d}e^{t\mathbf{A}}/\mathrm{d}t = \mathbf{A}e^{t\mathbf{A}} = e^{t\mathbf{A}}\mathbf{A}$. As such, an example of a function $\boldsymbol{\Gamma}(\epsilon)$ that satisfies the criteria for use in (6.10) is

$$\boldsymbol{\Gamma}(\epsilon) = e^{-\epsilon\mathbf{g}^{\times}}\mathbf{C}, \tag{6.11}$$

$$\boldsymbol{\Gamma}'(0) = -\mathbf{g}^{\times}\mathbf{C}, \tag{6.12}$$

where $\mathbf{g}$ is any element of $\mathbb{R}^3$.

Example 6.5. Consider the function $f(\mathbf{C}) = \mathrm{tr}(\mathbf{B}^{\mathsf{T}}\mathbf{C})$, that maps $SO(3)$ to $\mathbb{R}$, where $\mathbf{B} \in \mathbb{R}^{3\times 3}$. Evaluating (6.10) using (6.11)-(6.12) yields

$$\left\langle \boldsymbol{\nabla} f(\mathbf{C}), -\mathbf{g}^{\times}\mathbf{C} \right\rangle = \frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathrm{tr}\left(\mathbf{B}^{\mathsf{T}}e^{-\epsilon\mathbf{g}^{\times}}\mathbf{C}\right)\Big|_{\epsilon=0}$$

$$\mathrm{tr}\left[-(\boldsymbol{\nabla} f(\mathbf{C}))^{\mathsf{T}}\mathbf{g}^{\times}\mathbf{C}\right] = \mathrm{tr}(-\mathbf{B}^{\mathsf{T}}\mathbf{g}^{\times}\mathbf{C}).$$

Using the identity (6.1) yields

$$\mathrm{tr}\left[-\mathbf{C}(\boldsymbol{\nabla} f(\mathbf{C}))^{\mathsf{T}}\mathbf{g}^{\times}\right] = \mathrm{tr}(-\mathbf{g}^{\times}\boldsymbol{\mathcal{P}}_a(\mathbf{C}\mathbf{B}^{\mathsf{T}})).$$

Since $\boldsymbol{\nabla} f(\mathbf{C}) = \boldsymbol{\Omega}\mathbf{C}$ for some $\boldsymbol{\Omega} \in \mathfrak{so}(3)$, $\mathbf{C}(\boldsymbol{\nabla} f(\mathbf{C}))^{\mathsf{T}} = -\mathbf{C}\mathbf{C}^{\mathsf{T}}\boldsymbol{\Omega} = -\boldsymbol{\Omega}$ is antisymmetric. Thus using identity (6.2) yields

$$\begin{aligned} 2\mathbf{g}^{\mathsf{T}} \left[\mathbf{C}(\boldsymbol{\nabla} f(\mathbf{C}))^{\mathsf{T}}\right]^{\vee} &= 2\mathbf{g}^{\mathsf{T}} \boldsymbol{\mathcal{P}}_a(\mathbf{C}\mathbf{B}^{\mathsf{T}})^{\vee} \\ \mathbf{g}^{\mathsf{T}} \left[\mathbf{C}(\boldsymbol{\nabla} f(\mathbf{C}))^{\mathsf{T}} - \boldsymbol{\mathcal{P}}_a(\mathbf{C}\mathbf{B}^{\mathsf{T}})\right]^{\vee} &= \mathbf{0}. \end{aligned} \tag{6.13}$$

Since (6.13) must hold for all $\mathbf{g}$, the gradient is given by

$$\boldsymbol{\nabla} f(\mathbf{C}) = -\boldsymbol{\mathcal{P}}_a(\mathbf{C}\mathbf{B}^{\mathsf{T}})\mathbf{C}. \tag{6.14}$$

■

6.2.4 Gradients on $SO(3)$ Using Projection

A different method to derive a gradient on $SO(3)$ can be found using the projections discussed in Section 6.2.2. Let $\overline{f}$ be defined on $\mathbb{R}^{3\times 3} \to \mathbb{R}$, and let f be the restriction of $\overline{f}$ defined on $SO(3) \to \mathbb{R}$. The gradient of f at $\mathbf{C}$ is denoted as $\boldsymbol{\nabla} f(\mathbf{C})$, and the gradient of $\overline{f}$ at $\mathbf{C}$ is denoted as $\boldsymbol{\nabla}\overline{f}(\mathbf{C})$. The gradient $\boldsymbol{\nabla}\overline{f}(\mathbf{C})$ is given by [80, p. 641]

$$\boldsymbol{\nabla}\overline{f}(\mathbf{C}) = \left(\frac{\mathrm{d}\overline{f}}{\mathrm{d}\mathbf{C}}\right)^{\mathsf{T}}. \tag{6.15}$$

The gradient $\boldsymbol{\nabla} f(\mathbf{C})$ can be expressed in terms of $\boldsymbol{\nabla}\overline{f}(\mathbf{C})$ using (6.8), by projecting $\boldsymbol{\nabla}\overline{f}(\mathbf{C})$ onto $T_{\mathbf{C}}SO(3)$, that is [79, p. 48]

$$\boldsymbol{\nabla} f(\mathbf{C}) = \boldsymbol{\mathcal{P}}_{\mathbf{C}}(\boldsymbol{\nabla}\overline{f}(\mathbf{C})). \tag{6.16}$$

Example 6.6. Let $\overline{f}(\mathbf{C}) = \mathrm{tr}(\mathbf{B}^{\mathsf{T}}\mathbf{C})$, $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{3\times 3}$, and let f be the restriction of $\overline{f}$ such that $\mathbf{C} \in SO(3)$. The gradient $\boldsymbol{\nabla}\overline{f}(\mathbf{C})$ is given by [81, p. 691]

$$\boldsymbol{\nabla}\overline{f}(\mathbf{C}) = \left(\frac{\mathrm{d}}{\mathrm{d}\mathbf{C}}\mathrm{tr}(\mathbf{B}^{\mathsf{T}}\mathbf{C})\right)^{\mathsf{T}} = \mathbf{B}.$$

Using (6.8) and (6.16), the gradient $\boldsymbol{\nabla} f(\mathbf{C})$ is given by

$$\boldsymbol{\nabla} f(\mathbf{C}) = \boldsymbol{\mathcal{P}}_a(\mathbf{B}\mathbf{C}^{\mathsf{T}})\mathbf{C} = -\boldsymbol{\mathcal{P}}_a(\mathbf{C}\mathbf{B}^{\mathsf{T}})\mathbf{C}, \tag{6.17}$$

which is the same result as in (6.14). ■

6.2.5 Gradient from Taylor Series

Although the previous two sections have provided an analytic method to determine the gradient of a function on $SO(3)$, a numerical method is needed in this chapter for implementation purposes. In this section, we show the relationship between a Taylor series expansion of $f(\mathbf{C}) : SO(3) \to \mathbb{R}$ and the gradient $\boldsymbol{\nabla} f(\mathbf{C})$. This relationship is critical for the extremum-seeking guidance algorithm presented in Section 6.4.

Let $\mathbf{C} = e^{-\boldsymbol{\phi}^\times}\widehat{\mathbf{C}}$ be a parameterization of $\mathbf{C}$, where $\widehat{\mathbf{C}} \in SO(3)$ is constant, and where $\boldsymbol{\phi} = [\phi_1 \quad \phi_2 \quad \phi_3]^\mathsf{T} \in \mathbb{R}^3$ represents a rotation from the nominal $\widehat{\mathbf{C}}$. The first order Taylor series expansion of a function $f(\mathbf{C}(\boldsymbol{\phi}))$ at $\widehat{\mathbf{C}}$, that is around the point $\boldsymbol{\phi} = \mathbf{0} + \delta\boldsymbol{\phi}$, is given by

$$f(\mathbf{C}(\boldsymbol{\phi})) \approx f(\mathbf{C}(\mathbf{0})) + \mathbf{b}^\mathsf{T}\delta\boldsymbol{\phi}, \tag{6.18}$$

where $\mathbf{b} = [b_1 \quad b_2 \quad b_3]^\mathsf{T}$ is the gradient of (6.18), and where

$$\mathbf{b}^\mathsf{T} = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\phi}} f(\mathbf{C}(\boldsymbol{\phi}))\bigg|_{\boldsymbol{\phi}=\mathbf{0}}. \tag{6.19}$$

Since $\mathbf{b} \in \mathbb{R}^3$ is the gradient of $f(\mathbf{C}(\boldsymbol{\phi})) : \mathbb{R}^3 \to \mathbb{R}$ at $\boldsymbol{\phi} = \mathbf{0}$, and $\boldsymbol{\nabla} f(\widehat{\mathbf{C}}) \in T_{\widehat{\mathbf{C}}}SO(3)$ is the gradient of $f(\mathbf{C}) : SO(3) \to \mathbb{R}$ at $\widehat{\mathbf{C}}$, the matrices $\mathbf{b}$ and $\boldsymbol{\nabla} f(\widehat{\mathbf{C}})$ are different. However, for brevity, we refer to both $\mathbf{b}$ and $\boldsymbol{\nabla} f(\widehat{\mathbf{C}})$ as the gradient. This practice is acceptable in this chapter since $\mathbf{b}$ can uniquely identify $\boldsymbol{\nabla} f(\widehat{\mathbf{C}})$ and vice versa, via the relation given in Proposition 6.7.

Proposition 6.7. *The gradient* $\mathbf{b}$ *and the gradient* $\boldsymbol{\nabla} f(\widehat{\mathbf{C}})$ *are related via*

$$\boldsymbol{\nabla} f(\widehat{\mathbf{C}}) = -\frac{1}{2}\mathbf{b}^\times\widehat{\mathbf{C}}. \tag{6.20}$$

Proof. Since the parameterization of $\mathbf{C}$ results in $\mathbf{C} \in SO(3)$ for all $\boldsymbol{\phi} \in \mathbb{R}^3$, we are able to relax f to $\overline{f}$. Using (6.15), (6.19), and the chain rule given by (6.3), each

element of $\mathbf{b}$ is given by

$$\begin{aligned} b_i = \frac{\mathrm{d}}{\mathrm{d}\phi_i} f(\mathbf{C}(\boldsymbol{\phi}))\bigg|_{\boldsymbol{\phi}=\mathbf{0}} &= \operatorname{tr}\left[\left(\frac{\mathrm{d}\overline{f}(\mathbf{C})}{\mathrm{d}\mathbf{C}}\right)\left(\frac{\mathrm{d}\mathbf{C}}{\mathrm{d}\phi_i}\right)\right]_{\boldsymbol{\phi}=\mathbf{0}} \\ &= \operatorname{tr}\left[(\boldsymbol{\nabla}\overline{f}(\mathbf{C}))^{\mathsf{T}}\left(\frac{\mathrm{d}}{\mathrm{d}\phi_i}\exp\left(-\sum_{j=1}^{3}\phi_j\mathbf{1}_j^{\times}\right)\widehat{\mathbf{C}}\right)\right]_{\boldsymbol{\phi}=\mathbf{0}} \\ &= \operatorname{tr}\left[(\boldsymbol{\nabla}\overline{f}(\mathbf{C}))^{\mathsf{T}}\left(-\mathbf{1}_i^{\times}\exp\left(-\sum_{j=1}^{3}\phi_j\mathbf{1}_j^{\times}\right)\widehat{\mathbf{C}}\right)\right]_{\boldsymbol{\phi}=\mathbf{0}} \\ &= \operatorname{tr}\left[-(\boldsymbol{\nabla}\overline{f}(\widehat{\mathbf{C}}))^{\mathsf{T}}\mathbf{1}_i^{\times}\widehat{\mathbf{C}}\right] \\ &= \operatorname{tr}\left[-\widehat{\mathbf{C}}(\boldsymbol{\nabla}\overline{f}(\widehat{\mathbf{C}}))^{\mathsf{T}}\mathbf{1}_i^{\times}\right], \end{aligned} \tag{6.21}$$

where $\boldsymbol{\nabla}\overline{f}(\widehat{\mathbf{C}})$ is the gradient of $\overline{f}$ at $\widehat{\mathbf{C}}$. Using (6.1) and (6.2), (6.21) simplifies to

$$b_i = 2\left[\boldsymbol{\mathcal{P}}_a\left(\widehat{\mathbf{C}}(\boldsymbol{\nabla}\overline{f}(\widehat{\mathbf{C}}))^{\mathsf{T}}\right)^{\vee}\right]^{\mathsf{T}}\mathbf{1}_i,$$

and thus

$$\begin{aligned} \mathbf{b} &= 2\boldsymbol{\mathcal{P}}_a\left(\widehat{\mathbf{C}}(\boldsymbol{\nabla}\overline{f}(\widehat{\mathbf{C}}))^{\mathsf{T}}\right)^{\vee}, \\ -\frac{1}{2}\mathbf{b}^{\times} &= \boldsymbol{\mathcal{P}}_a\left(\boldsymbol{\nabla}\overline{f}(\widehat{\mathbf{C}})\,\widehat{\mathbf{C}}^{\mathsf{T}}\right). \end{aligned} \tag{6.22}$$

Right multiplying (6.22) by $\widehat{\mathbf{C}}$ and substituting in (6.16) yields (6.20). □

6.3 Maximization of a Function $J : SO(3) \to \mathbb{R}$

In this section, we review the gradient ascent and constrained gradient ascent methods to solve the problem

$$\max_{\mathbf{C}_{ba}\in SO(3)} J = J(\mathbf{C}_{ba}), \tag{6.23}$$

$$\text{s.t. } \mathbf{x}_b^{i\mathsf{T}}\mathbf{C}_{ba}\mathbf{y}_a^i \geq \cos(\alpha_i),\ i = 1,\ldots,n_{\mathrm{c}}, \tag{6.24}$$

where $J : SO(3) \to \mathbb{R}$ is continuous, $\alpha_i \in \mathbb{R}$, and n_{c} is the number of constraints. Since the manifold $SO(3)$ is compact [79, p. 42], and since J is continuous, J has an extremum in $SO(3)$ by Theorem 2.7. Equation (6.24) can be used to define both inclusion zones and exclusion zones, where $\mathbf{x}_b^i$ and $\mathbf{C}_{ba}\mathbf{y}_a^i$ must have a separation angle

no larger than α_i. To conserve the "$\geq$" sign, the exclusion zone between $\mathbf{x}_b^i$ and $\mathbf{C}_{ba}\mathbf{y}_a^i$ with angle α_i is written as

$$\mathbf{x}_b^{i\mathsf{T}}\mathbf{C}_{ba}(-\mathbf{y}_a^i) \geq \cos(\pi - \alpha_i). \tag{6.25}$$

In Section 6.3.1, the algorithm to solve the unconstrained problem (6.23) is considered. In Section 6.3.2, the algorithm to solve the constrained problem defined by (6.23) and (6.24) is shown.

6.3.1 Gradient Ascent

A retraction on a manifold is a mapping from the tangent space to the manifold, that is $T_{\mathbf{C}}SO(3) \to SO(3)$. The retraction of $\boldsymbol{\nabla} f(\mathbf{C}) = \boldsymbol{\Omega}\mathbf{C}$ is given by a Riemannian exponential mapping on $SO(3)$, and defined as $R_{\mathbf{C}}(\boldsymbol{\Omega}\mathbf{C}) = e^{\boldsymbol{\Omega}}\mathbf{C}$ [79, p. 59].

Example 6.8. Consider the gradient from Example 6.5 and 6.6. The retraction of $\boldsymbol{\nabla} f(\mathbf{C}) = -\boldsymbol{\mathcal{P}}_a(\mathbf{C}\mathbf{B}^{\mathsf{T}})\mathbf{C}$ is given by

$$R_{\mathbf{C}}(\boldsymbol{\nabla} f(\mathbf{C})) = \exp\left[-\boldsymbol{\mathcal{P}}_a(\mathbf{C}\mathbf{B}^{\mathsf{T}})\right]\mathbf{C}, \tag{6.26}$$

and (3.3) can be used to evaluate the exponential in (6.26), with $\boldsymbol{\phi} = \boldsymbol{\mathcal{P}}_a(\mathbf{C}\mathbf{B}^{\mathsf{T}})^{\vee}$. ■

Let the optimizer of $J = J(\mathbf{C}_{ba})$ be $\mathbf{C}_{b^\star a}$. The frame $\mathcal{F}_{b_k}$ is a frame that is defined at the k^{th} step of the gradient ascent algorithm. The gradient ascent algorithm determines successive $\mathbf{C}_{b_{k+1}a}$ using the gradient of J at $\mathbf{C}_{b_k a}$, given by $\boldsymbol{\nabla} J(\mathbf{C}_{b_k a})$. The matrix $\mathbf{C}_{b_{k+1}a}$ is given by

$$\mathbf{C}_{b_{k+1}a} = R_{\mathbf{C}_{b_k a}}(2\kappa_k \boldsymbol{\nabla} J(\mathbf{C}_{b_k a})), \tag{6.27}$$

where $2\kappa_k > 0$ is the step size [79, p. 62]. When the gradient $\boldsymbol{\nabla} J(\mathbf{C}_{b_k a})$ is determined from a Taylor series such as in (6.18), the update law is given by

$$\mathbf{C}_{b_{k+1}a} = e^{-\mathbf{d}_k^{\times}}\mathbf{C}_{b_k a}, \tag{6.28}$$

where $\mathbf{d}_k = \kappa_k \mathbf{b}_k$. Equation (6.28) is obtained by substituting (6.20) into (6.27), which is the critical step that enables the extremum-seeking algorithm. This gradient ascent algorithm is shown in Fig. 6.1. The step in the tangent space $T_{\mathbf{C}_{b_k a}}SO(3)$ is $-\kappa_k \mathbf{b}_k^{\times}\mathbf{C}_{b_k a}$, depicted by the blue arrow. The retraction from $T_{\mathbf{C}_{b_k a}}SO(3)$ to $SO(3)$ represented by the red arrow.

If the objective function is known and can be evaluated, κ_k can be chosen via

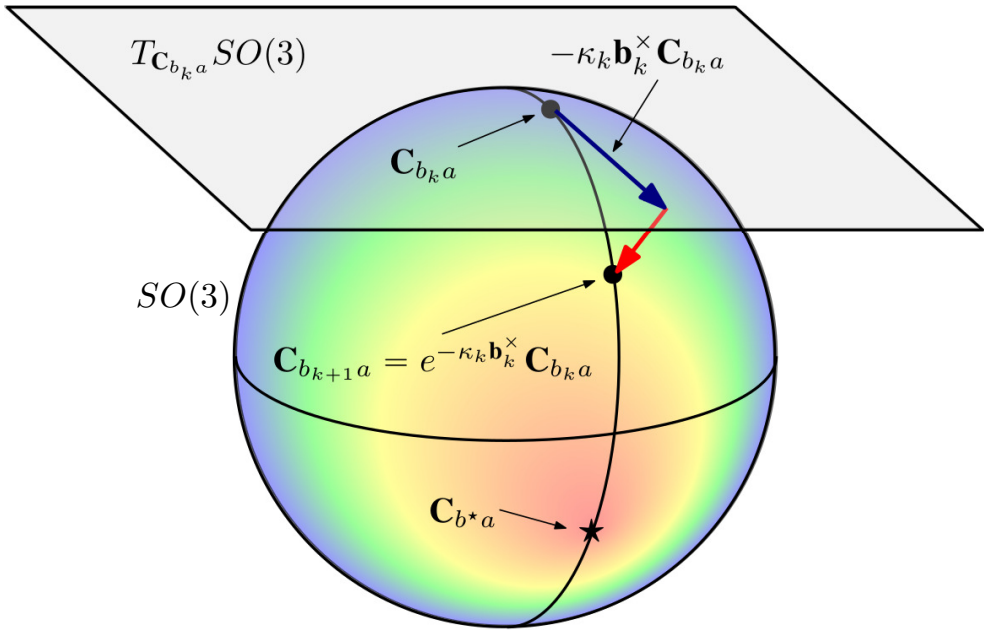


Figure 6.1: Gradient ascent for $J = J(\mathbf{C}_{ba})$, with maximum $\mathbf{C}_{b^\star a}$. The sphere is a visualization of $SO(3)$, and the plane is a visualization of the tangent space $T_{\mathbf{C}_{b_k a}} SO(3)$. The blue arrow is a step in the tangent space, and the red arrow represents the retraction from $T_{\mathbf{C}_{b_k a}} SO(3)$ to $SO(3)$. The shading represents values of J, with the maximum value at the red shading, and the minimum values at the blue shading.

an exact line search to ensure that there is a sufficient decay in J. However, the exact line search can be computationally expensive, and thus the line search can be approximated via Armijo's rule to provide a maximum step size [82, p. 33]. Wolfe's conditions also provide a provision for sufficient decrease by providing a lower bound on the step size [82, p. 33]. In extremum-seeking, the objective function cannot be evaluated and neither of these conditions can be used. Instead, the step size κ_k can be chosen to satisfy

$$d_{\min} \leq \kappa_k \left\| \mathbf{b}_k \right\|_2 \leq d_{\max}, \tag{6.29}$$

where $d_{\min}$ is a minimum step size and $d_{\max}$ is a maximum step size. These conditions do not directly ensure a sufficient decrease, but allow for quicker convergence of J when the curvature of J becomes small.

6.3.2 Constrained-Gradient Ascent

Consider the optimization problem given by (6.23)-(6.24). This problem is solved by incorporating constraints to the gradient ascent method presented in Section 6.3.1.

The updated DCM $\mathbf{C}_{b_{k+1}a}$ must satisfy (6.24), and thus (6.24) is rewritten as

$$\mathbf{x}_{b_{k+1}}^{i\mathsf{T}}\mathbf{C}_{b_{k+1}a}\mathbf{y}_a^i \geq \cos(\alpha_i). \tag{6.30}$$

The constraint defined by (6.30) can be transformed to a linear inequality constraint if the step $\mathbf{d}_k$ is small. One method to obtain a small $\mathbf{d}_k$ is by considering the norm constraint

$$\mathbf{d}_k^{\mathsf{T}}\mathbf{d}_k \leq d_{\max}^2. \tag{6.31}$$

With a small step $\mathbf{d}_k$, the small angle approximation (3.4) can be used, and (6.28) becomes

$$\mathbf{C}_{b_{k+1}a} = (\mathbf{1} - \mathbf{d}_k^{\times})\mathbf{C}_{b_k a}. \tag{6.32}$$

Substituting (6.32) into (6.30) yields

$$\mathbf{x}_{b_{k+1}}^{i\mathsf{T}}(\mathbf{1} - \mathbf{d}_k^{\times})\mathbf{C}_{b_k a}\mathbf{y}_a^i \geq \cos(\alpha_i). \tag{6.33}$$

Simplifying and rearranging (6.33) yields

$$\mathbf{A}_{\text{con},k}\mathbf{d}_k \geq \boldsymbol{\xi}_{\text{con},k}, \tag{6.34}$$

where

$$\mathbf{A}_{\text{con},k} = \begin{bmatrix} \mathbf{x}_{b_{k+1}}^{1\mathsf{T}}(\mathbf{C}_{b_k a}\mathbf{y}_a^1)^{\times} \\ \vdots \\ \mathbf{x}_{b_{k+1}}^{n_c\mathsf{T}}(\mathbf{C}_{b_k a}\mathbf{y}_a^{n_c})^{\times} \end{bmatrix}, \quad \boldsymbol{\xi}_{\text{con},k} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_{n_c} \end{bmatrix},$$

and where $\beta_i = \cos(\alpha_i) - \mathbf{x}_{b_{k+1}}^{i\mathsf{T}}\mathbf{C}_{b_k a}\mathbf{y}_a^i$. The linearization of the constraint is the second contribution. The linearization is possible since we are looking for $\mathbf{C}_{b_{k+1}a}$ in the neighbourhood of $\mathbf{C}_{b_k a}$ rather than looking for any $\mathbf{C}_{b_{k+1}a}$ that satisfies (6.30). Other methods to determine attitude based on inclusion and exclusion zones can be found in [51], but this method uses quaternions and quadratic constraints. In [78], admissible sets are used, and this method does not fit within this framework. Equation (6.34) is a linear constraint, which works well given the nature of the application, which is a projected gradient optimization.

The DCM $\mathbf{C}_{b_{k+1}a}$ is still given by (6.28), but $\mathbf{d}_k$ is chosen as the minimizer of

$$\min \frac{1}{2}(\mathbf{d}_k - \kappa_k\mathbf{b}_k)^{\mathsf{T}}(\mathbf{d}_k - \kappa_k\mathbf{b}_k), \tag{6.35}$$

such that (6.31) and (6.34) are satisfied. Ignoring (6.31) and choosing a small κ_k,

the solution to (6.35) subject to (6.34) is given by constructing the matrices of active constraints $\mathbf{A}_{\text{con},k}^{\text{active}}$ and $\boldsymbol{\xi}_{\text{con},k}^{\text{active}}$ to yield

$$\mathbf{A}_{\text{con},k}^{\text{active}}\mathbf{d}_k = \boldsymbol{\xi}_{\text{con},k}^{\text{active}}, \tag{6.36}$$

where the minimizing $\mathbf{d}_k$ is given by

$$\mathbf{d}_k = \kappa\mathbf{b}_k - \mathbf{A}_{\text{con},k}^{\text{active}^{\mathsf{T}}}(\mathbf{A}_{\text{con},k}^{\text{active}}\mathbf{A}_{\text{con},k}^{\text{active}^{\mathsf{T}}})^{-1}(\mathbf{A}_{\text{con},k}^{\text{active}}\kappa\mathbf{b}_k - \boldsymbol{\xi}_{\text{con},k}^{\text{active}}). \tag{6.37}$$

Equation (6.37) projects $\mathbf{d}_k$ onto the constrained set (6.34).

With a small step size κ_k, ignoring (6.31) is acceptable if it is either assumed or known that $\kappa\mathbf{b}_k$ is always small. When estimating the constrained gradient in Section 6.5, the gain-projected Kalman filter of Section 6.5.1 requires linear constraints, and norm constraints must be reformulated or ignored. However, the Kalman filter in Section 6.5.2 that uses LMI's to determine the Kalman gain is capable of using (6.31) directly in the filter formulation.

6.4 Unconstrained Guidance on $SO(3)$ with an Unknown Objective Function

Consider the system shown in Fig. 6.2. The attitude of $\mathcal{F}_b$ relative to $\mathcal{F}_a$ is parameterized by $\mathbf{C}_{ba}$. The function $J : SO(3) \to \mathbb{R}$ is a performance function. The outputs of the plant are $\mathbf{C}_{ba}$ and the measurement of $J = J(\mathbf{C}_{ba})$, described in Section 6.4.1. The extremum-seeking guidance law determines a desired frame $\mathcal{F}_d$ and the DCM $\mathbf{C}_{da}$. The attitude error $\mathbf{C}_{bd} = \mathbf{C}_{ba}\mathbf{C}_{da}^{\mathsf{T}}$ is used for feedback in the controller to drive $\mathbf{C}_{ba}$ to $\mathbf{C}_{da}$. Ideally, the desired attitude $\mathbf{C}_{da}$ converges to $\mathbf{C}_{b^\star a}$, where $\mathbf{C}_{b^\star a}$ is an unknown extremum of $J(\mathbf{C}_{ba})$. Therefore, since $\mathbf{C}_{ba}$ converges to $\mathbf{C}_{da}$ through feedback control, $\mathbf{C}_{ba}$ converges to $\mathbf{C}_{b^\star a}$. When the mapping of $SO(3) \to \mathbb{R}$ of J is unknown, the gradient must be estimated to use the gradient ascent algorithm from Section 6.3.1.

Using the gradient ascent optimization method and (6.28), the desired attitude is given by

$$\mathbf{C}_{d_k a} = e^{-\mathbf{d}_k^\times}\mathbf{C}_{b_k a}, \tag{6.38}$$

where $\mathbf{d}_k$ is the step. In gradient ascent, the update (6.28) gives the next attitude in the optimization, $\mathbf{C}_{b_{k+1}a}$. However, in extremum-seeking guidance, the update (6.38) yields the desired attitdue $\mathbf{C}_{d_k a}$ at time t_k. The purpose of this section is to ex-

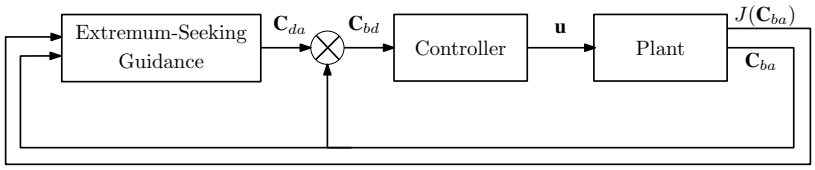


Figure 6.2: Extremum-seeking guidance block diagram. The attitude error is given by $\mathbf{C}_{bd} = \mathbf{C}_{ba}\mathbf{C}_{da}^{\mathsf{T}}$.

plain how to determine $\mathbf{d}_k$ for unconstrained extremum-seeking on $SO(3)$. Since $\mathbf{d}_k$ is determined from the gradient, the gradient must be estimated. The gradient is estimated using measurements described in Section 6.4.1 and the filter described in Section 6.4.2.

6.4.1 Performance Measurements

The estimation of the gradient of J is enabled by taking a first order Taylor series expansion of J using (6.18), at $\mathbf{C}_{ba}(t_k) = \mathbf{C}_{b_k a}$. The parameterization $\mathbf{C}_{ba} = \exp(-\boldsymbol{\phi}^{bb_k^\times})\mathbf{C}_{b_k a}$ in the Taylor series expansion yields

$$J(\mathbf{C}_{ba}) \approx J(\mathbf{C}_{b_k a}) + \mathbf{b}_k^{\mathsf{T}} \boldsymbol{\phi}^{bb_k}, \tag{6.39}$$

where $\mathbf{b}_k$ is the gradient of J at $\mathbf{C}_{b_k a}$. The unit of $\mathbf{b}_k$ is typically the unit of J per radian. When using a first order Taylor series expansion, it is implied that the Taylor series is a valid approximation of J in the neighbourhood of $\mathbf{C}_{b_k a}$. Therefore, if we knew the exact value of $\mathbf{b}_k$, $J(\mathbf{C}_{b_{k-1} a})$ could be approximated by evaluating (6.39) at $\mathbf{C}_{ba} = \mathbf{C}_{b_{k-1} a}$ to obtain

$$J(\mathbf{C}_{b_{k-1} a}) \approx J(\mathbf{C}_{b_k a}) + \mathbf{b}_k^{\mathsf{T}} \boldsymbol{\phi}^{b_{k-1} b_k}.$$

As in Section 5.2, the Taylor series is used to approximate $\mathbf{b}_k$ in the the neighbourhood of $\mathbf{C}_{b_k a}$. Equation (6.39) evaluated at $\mathbf{C}_{ba} = \mathbf{C}_{b_{k-\ell} a}$, $\ell = 1, \ldots, N$ and rearranged results in

$$\begin{bmatrix} \Delta J_k \\ \vdots \\ \Delta J_{k-N+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\phi}^{b_k b_{k-1}^{\mathsf{T}}} \\ \vdots \\ \boldsymbol{\phi}^{b_k b_{k-\ell}^{\mathsf{T}}} \end{bmatrix} \mathbf{b}_k, \tag{6.40}$$

where

$$\Delta J_{k-\ell+1} = J(\mathbf{C}_{b_k a}) - J(\mathbf{C}_{b_{k-\ell} a}),$$
$$\boldsymbol{\phi}^{b_k b_{k-\ell}} = -(\ln(\mathbf{C}_{b_k a}\mathbf{C}_{b_{k-\ell} a}^{\mathsf{T}}))^{\vee}.$$

Equation (6.40) is written more compactly as

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{b}_k, \tag{6.41}$$

where

$$\mathbf{y}_k = \begin{bmatrix} \Delta J_k \\ \vdots \\ \Delta J_{k-N+1} \end{bmatrix}, \quad \mathbf{H}_k = \begin{bmatrix} \boldsymbol{\phi}^{b_k b_{k-1}^{\mathsf{T}}} \\ \vdots \\ \boldsymbol{\phi}^{b_k b_{k-\ell}^{\mathsf{T}}} \end{bmatrix}. \tag{6.42}$$

Both $\mathbf{y}_k$ and $\mathbf{H}_k$ can be determined from the current and previous attitude estimates of the plant.

Several methods of determining $\mathbf{b}_k$ using (6.41) exist, such as a parameter estimation approach, where $\mathbf{b}_k$ contains the parameters and $\mathbf{H}_k$ is the regressor. In this chapter, a Kalman filter formulation is used to estimate $\mathbf{b}_k$. Equation (6.41), which forms the basis of the measurement model, is augmented with a process model, and is explained Section 6.4.2.

6.4.2 Kalman Filter Formulation

As in Section 5.2, a linear time-varying Kalman filter is used to estimate $\mathbf{b}_k$, by specifying a process model and by modifying (6.41) to be the measurement model, where both are given by

$$\mathbf{b}_k = \mathbf{b}_{k-1} + \mathbf{w}_{k-1}, \tag{6.43}$$
$$\mathbf{y}_k = \mathbf{H}_k \mathbf{b}_k + \mathbf{v}_k, \tag{6.44}$$

where $\mathbf{w}_{k-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$, $\mathbf{Q}_k \geq 0$, $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$, $\mathbf{R}_k > 0$, and where $\mathbf{y}_k$ and $\mathbf{H}_k$ are defined in (6.42). The notation $\mathbf{w} \sim \mathcal{N}(\bar{\mathbf{w}}, \mathbf{Q})$, $\mathbf{w}, \bar{\mathbf{w}} \in \mathbb{R}^n$, $\mathbf{Q} \in \mathbb{R}^{n \times n}$, indicates that $\mathbf{w}$ is Gaussian with a mean of $\bar{\mathbf{w}}$ and covariance $\mathbf{Q}$.

The Kalman filter estimates $\mathbf{b}_k$ using a prediction step

$$\begin{aligned}\hat{\mathbf{b}}_k^- &= \hat{\mathbf{b}}_{k-1},\\ \hat{\mathbf{y}}_k^- &= \mathbf{H}_k\hat{\mathbf{b}}_k^-,\\ \mathbf{P}_k^- &= \mathbf{P}_{k-1} + \mathbf{Q}_{k-1},\end{aligned}$$

and a correction step

$$\hat{\mathbf{b}}_k = \hat{\mathbf{b}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-), \tag{6.45}$$

$$\mathbf{P}_k = (\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{1} - \mathbf{K}_k\mathbf{H}_k)^\mathsf{T} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^\mathsf{T}, \tag{6.46}$$

where $\hat{\mathbf{b}}_k$ and $\hat{\mathbf{b}}_k^-$ are the predicted and estimated gradient, and $\mathbf{P}_k^-$ and $\mathbf{P}_k$ are the predicted and estimated error covariance. Equation (6.46) is known as the "Joseph form", and is more numerically stable than other forms, while still being valid for any gain $\mathbf{K}_k$ [34, pp. 464-465]. This is useful since in Section 6.5.2, (6.46) can be used to correct $\mathbf{P}_k^-$. In Section 6.5.2, the derived $\mathbf{K}_k$ is different from the traditional Kalman gain, which is found by minimizing $\mathrm{tr}(\mathbf{P}_k)$ [34, p. 464]. The Kalman gain $\mathbf{K}_k$ is given by

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^\mathsf{T}(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^\mathsf{T} + \mathbf{R}_k)^{-1}. \tag{6.47}$$

The step is determined using the estimated gradient,

$$\mathbf{d}_k = \kappa_k\hat{\mathbf{b}}_k + \mathbf{w}_k^{\mathrm{pe}}, \tag{6.48}$$

where $\mathbf{w}_k^{\mathrm{pe}}$ is persistent excitation required to ensure observability of the performance function [13]. The desired attitude is given by substituting (6.48) into (6.38). Unconstrained extremum-seeking guidance is summarized in Algorithm 6.4.

6.5 Constrained Guidance on $SO(3)$ with an Unknown Objective Function

Constrained extremum-seeking guidance is similar to the unconstrained version in that the step $\mathbf{d}_k$ from (6.38) must be determined. Two different constrained extremum-seeking guidance problems are considered in this section. First, the gain-projected Kalman filter solves an extremum-seeking problem to maximize (6.23), subject to the inclusion and exclusion zones (6.24). The gain-projected Kalman filter estimates the constrained gradient that can be used with the constrained gradient as-

Algorithm 6.4 Unconstrained Extremum-Seeking Guidance $SO(3) \to \mathbb{R}$

for $k = k_{\text{initial}}$ to k_{final} **do**
  Measure $\mathbf{C}_{b_{k-\ell}a}$ for $\ell = 0, \ldots, N$.
  Measure $J(\mathbf{C}_{b_{k-\ell}a})$ for $\ell = 0, \ldots, N$.
  Construct $\mathbf{y}_k$ and $\mathbf{H}_k$ using (6.42).
  Predict gradient using $\hat{\mathbf{b}}_k^- = \hat{\mathbf{b}}_{k-1}$.
  Predict covariance using $\mathbf{P}_k^- = \mathbf{P}_{k-1} + \mathbf{Q}_{k-1}$.
  Determine gain $\mathbf{K}_k$ using (6.47).
  Correct gradient to obtain $\hat{\mathbf{b}}_k$ using (6.45).
  Correct covariance to obtain $\mathbf{P}_k$ using (6.46).
  Determine $\mathbf{d}_k$ using (6.48).
  Determine the desired attitude for the next time step $\mathbf{C}_{d_k a}$ using (6.38).
end for

cent method from Section 6.3.2. As the name implies, the gain-projected Kalman filter estimates the gradient, and then projects it to a constrained set. Second, the LMI-based Kalman filter solves almost the same problem as the gain-projected Kalman filter. Unlike the gain-projected Kalman filter, which only handles linear equality and inequality constraints, the LMI-based Kalman filter can handle any type of LMI-based constraints. The disadvantage is that an SDP must be solved at each guidance step, but norm constraints can be efficiently handled with this formulation. Therefore, the norm constraint given by (6.31) is considered in addition to the inclusion and exclusion zones (6.24). Both constrained extremum-seeking algorithms are summarized in Algorithm 6.5.

6.5.1 Gain-Projected Kalman Filter

The projected gradient method can be incorporated in the Kalman filter by using the gain-projected Kalman filter [70], which is also used in Section 5.3. It is briefly reviewed here for completeness. The gradient is first estimated as in the unconstrained case to obtain $\hat{\mathbf{b}}_k$. The constrained gradient estimate, $\tilde{\mathbf{b}}_k$, is then obtained by solving

$$\min \frac{1}{2}(\tilde{\mathbf{b}}_k - \hat{\mathbf{b}}_k)^\mathsf{T}(\tilde{\mathbf{b}}_k - \hat{\mathbf{b}}_k),$$

subject to

$$\kappa_k \mathbf{A}_{\text{con},k} \tilde{\mathbf{b}}_k \geq \boldsymbol{\xi}_{\text{con},k}, \tag{6.49}$$

where (6.49) is obtained by substituting $\mathbf{d}_k = \kappa_k \tilde{\mathbf{b}}_k$ in (6.34). The solution to this minimization problem is given by

$$\tilde{\mathbf{b}}_k = \hat{\mathbf{b}}_k - \kappa_k^{-1} \mathbf{A}_{\text{con},k}^{\text{active}^\mathsf{T}} (\mathbf{A}_{\text{con},k}^{\text{active}} \mathbf{A}_{\text{con},k}^{\text{active}^\mathsf{T}})^{-1} (\mathbf{A}_{\text{con},k}^{\text{active}} \kappa_k \hat{\mathbf{b}}_k - \boldsymbol{\xi}_{\text{con},k}^{\text{active}}). \tag{6.50}$$

At the next iteration, where k has been incremented by one, the new prediction step is given by

$$\hat{\mathbf{b}}_k^- = \tilde{\mathbf{b}}_{k-1}.$$

There is no change to the predicted covariance matrix $\mathbf{P}_k$, given by (6.46). The gain-projected method does not bias the Kalman filter and is simple to implement, which makes this method effective for scenarios with limited computational resources [70]. The desired attitude $\mathbf{C}_{d_{k+1}a}$ is given by substituting

$$\mathbf{d}_k = \kappa_k \tilde{\mathbf{b}}_k + \mathbf{w}_k^{\text{pe}} \tag{6.51}$$

into (6.38).

6.5.2 Kalman Filter with LMI Constraints

The LMI-based Kalman filter does not require a step to constrain $\hat{\mathbf{b}}_k$ to $\tilde{\mathbf{b}}_k$. Rather, the gain $\mathbf{K}_k$ is determined so that the constraints are imposed directly on the estimate $\hat{\mathbf{b}}_k$. The constraints are given by

$$\kappa_k \mathbf{A}_{\text{con},k} \hat{\mathbf{b}}_k \geq \boldsymbol{\xi}_{\text{con},k} - \boldsymbol{\zeta}_k, \tag{6.52}$$

$$\kappa_k^2 \hat{\mathbf{b}}_k^\mathsf{T} \hat{\mathbf{b}}_k \leq d_{\max}^2, \tag{6.53}$$

where (6.53) is obtained from (6.31) by substituting $\mathbf{d}_k = \kappa \hat{\mathbf{b}}_k$. The matrix $\boldsymbol{\zeta}_k \geq \mathbf{0}$ is a slack variable, which transforms (6.52) to a soft constraint. This slack variable is critical because if $\mathbf{C}_{b_k a}$ is not feasible, a large $\hat{\mathbf{b}}_k$ might be necessary to satisfy (6.52), which could violate (6.53). In this situation, without $\boldsymbol{\zeta}_k$, it would be impossible to simultaneously satisfy both sets of constraints. Any optimization problem posed with these constraints would be infeasible.

The gain matrix $\mathbf{K}_k$ is chosen to minimize the objective function

$$\mathcal{J}_k(\mathbf{K}_k, \boldsymbol{\zeta}_k) = \frac{1}{2}\left[(\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{b}}_k)^\mathsf{T} \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathbf{H}_k \hat{\mathbf{b}}_k) + (\hat{\mathbf{b}}_k - \hat{\mathbf{b}}_k^-)^\mathsf{T} (\mathbf{P}_k^-)^{-1} (\hat{\mathbf{b}}_k - \hat{\mathbf{b}}_k^-)\right] + \mathbf{s}_k^\mathsf{T} \boldsymbol{\zeta}_k, \tag{6.54}$$

which is inspired by the recursive least-squares [83, p. 205–207] and maximum like-

lihood approach to deriving the Kalman filter [84]. This objective function weighs the term $\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{b}}_k$ with respect to the measurement covariance $\mathbf{R}_k$, and weighs the residual between the prior and posterior estimates with respect to the prior covariance $\mathbf{P}_k^-$. The matrix $\mathbf{s}_k \geq \mathbf{0}$ is a weighting column matrix which penalizes a nonzero slack variable $\boldsymbol{\zeta}_k$. The matrix $\mathbf{s}_k$ is chosen to be sufficiently large to render $\mathbf{C}_{b_k a}$ feasible by driving $\boldsymbol{\zeta}_k$ to zero. When the constraints are infeasible, the term $\mathbf{s}_k^\mathsf{T}\boldsymbol{\zeta}_k$ is the dominant term of $\mathcal{J}_k$, and thus the primary objective is to satisfy (6.52) with a zero slack variable. Once this occurs, the first term of $\mathcal{J}_k$ is the dominant term, and normal Kalman filtering behaviour ensues.

Using the correction of the state estimate $\hat{\mathbf{b}}_k = \hat{\mathbf{b}}_k^- + \mathbf{K}_k\mathbf{r}_k$ from (6.45), where $\mathbf{r}_k = \mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{b}}_k^-$ is the innovation term, the objective function can be rewritten as

$$\begin{aligned}
\mathcal{J}_k(\mathbf{K}_k, \boldsymbol{\zeta}_k) &= \frac{1}{2}\big[(\mathbf{y}_k - \mathbf{H}_k(\hat{\mathbf{b}}_k^- + \mathbf{K}_k\mathbf{r}_k))^\mathsf{T}\mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{H}_k(\hat{\mathbf{b}}_k^- + \mathbf{K}_k\mathbf{r}_k)) \\
&\quad + ((\hat{\mathbf{b}}_k^- + \mathbf{K}_k\mathbf{r}_k) - \hat{\mathbf{b}}_k^-)^\mathsf{T}(\mathbf{P}_k^-)^{-1}((\hat{\mathbf{b}}_k^- + \mathbf{K}_k\mathbf{r}_k) - \hat{\mathbf{b}}_k^-)\big] + \mathbf{s}_k^\mathsf{T}\boldsymbol{\zeta}_k \\
&= \frac{1}{2}\big[(\mathbf{r}_k - \mathbf{H}_k\mathbf{K}_k\mathbf{r}_k)^\mathsf{T}\mathbf{R}_k^{-1}(\mathbf{r}_k - \mathbf{H}_k\mathbf{K}_k\mathbf{r}_k) + (\mathbf{K}_k\mathbf{r}_k)^\mathsf{T}(\mathbf{P}_k^-)^{-1}(\mathbf{K}_k\mathbf{r}_k)\big] + \mathbf{s}_k^\mathsf{T}\boldsymbol{\zeta}_k \\
&= \frac{1}{2}\mathbf{r}_k^\mathsf{T}\big[(\mathbf{1} - \mathbf{H}_k\mathbf{K}_k)^\mathsf{T}\mathbf{R}_k^{-1}(\mathbf{1} - \mathbf{H}_k\mathbf{K}_k) + \mathbf{K}_k^\mathsf{T}(\mathbf{P}_k^-)^{-1}\mathbf{K}_k\big]\mathbf{r}_k + \mathbf{s}_k^\mathsf{T}\boldsymbol{\zeta}_k. \qquad (6.55)
\end{aligned}$$

Without the norm constraint, $\boldsymbol{\zeta}_k$ is ignored, and the Kalman gain given by (6.47) can be recovered by differentiating (6.54) with respect to $\mathbf{K}_k$ and solving for $\mathbf{K}_k$, as shown in Appendix B.3.

A new matrix variable $\mathbf{Z}_k$ is introduced to transform (6.55) to a linear convex objective function given by [80, p. 67]

$$\widehat{\mathcal{J}}_k(\mathbf{K}_k, \mathbf{Z}_k, \boldsymbol{\zeta}_k) = \frac{1}{2}\mathbf{r}_k^\mathsf{T}\mathbf{Z}_k\mathbf{r}_k + \mathbf{s}_k^\mathsf{T}\boldsymbol{\zeta}_k, \qquad (6.56)$$

where $\mathbf{Z}_k$ is subject to the constraint

$$\begin{aligned}
\mathbf{Z}_k &\geq (\mathbf{1} - \mathbf{H}_k\mathbf{K}_k)^\mathsf{T}\mathbf{R}_k^{-1}(\mathbf{1} - \mathbf{H}_k\mathbf{K}_k) + \mathbf{K}_k^\mathsf{T}(\mathbf{P}_k^-)^{-1}\mathbf{K}_k^\mathsf{T}, \\
\mathbf{Z}_k - (\mathbf{1} - \mathbf{H}_k\mathbf{K}_k)^\mathsf{T}\mathbf{R}_k^{-1}(\mathbf{1} - \mathbf{H}_k\mathbf{K}_k) - \mathbf{K}_k^\mathsf{T}(\mathbf{P}_k^-)^{-1}\mathbf{K}_k^\mathsf{T} &\geq 0. \qquad (6.57)
\end{aligned}$$

Using the Schur Complement [31], (6.57) can be converted to an LMI in terms of $\mathbf{Z}_k$ and $\mathbf{K}_k$,

$$\begin{bmatrix} \mathbf{Z}_k & (\mathbf{1} - \mathbf{H}_k\mathbf{K}_k)^\mathsf{T} & \mathbf{K}_k^\mathsf{T} \\ (\mathbf{1} - \mathbf{H}_k\mathbf{K}_k) & \mathbf{R}_k & \mathbf{0} \\ \mathbf{K}_k & \mathbf{0} & \mathbf{P}_k^- \end{bmatrix} \geq 0. \qquad (6.58)$$

Algorithm 6.5 Constrained Extremum-Seeking Guidance $SO(3) \to \mathbb{R}$

for $k = k_{\text{initial}}$ to k_{final} **do**
  Measure $\mathbf{C}_{b_{k-\ell}a}$ for $\ell = 0, \ldots, N$.
  Measure $J(\mathbf{C}_{b_{k-\ell}a})$ for $\ell = 0, \ldots, N$.
  Construct $\mathbf{y}_k$ and $\mathbf{H}_k$ using (6.42).
  Predict covariance using $\mathbf{P}_k^- = \mathbf{P}_{k-1} + \mathbf{Q}_{k-1}$.
  if gain-projected Kalman filter **then**
    Predict gradient using $\hat{\mathbf{b}}_k^- = \tilde{\mathbf{b}}_{k-1}$.
    Determine gain $\mathbf{K}_k$ using (6.47).
    Determine active constraints $\mathbf{A}_{\text{con},k}^{\text{active}}$ and $\boldsymbol{\xi}_{\text{con},k}^{\text{active}}$.
    Correct gradient to obtain $\tilde{\mathbf{b}}_k$ using (6.50) .
    Determine $\mathbf{d}_k$ using (6.51).
  else if LMI-based Kalman filter **then**
    Predict gradient using $\hat{\mathbf{b}}_k^- = \hat{\mathbf{b}}_{k-1}$.
    Solve for $\mathbf{K}_k$ by minimizing (6.56) subject to (6.58)-(6.60).
    Correct gradient to obtain $\hat{\mathbf{b}}_k$ using (6.45) .
    Determine $\mathbf{d}_k$ using (6.48).
  end if
  Correct covariance to obtain $\mathbf{P}_k$ using (6.46).
  Determine the desired attitude for the next time step $\mathbf{C}_{d_k a}$ using (6.38).
end for

In addition, $\hat{\mathbf{b}}_k = \hat{\mathbf{b}}_k^- + \mathbf{K}_k \mathbf{r}_k$ is substituted into (6.49) and (6.53) to obtain

$$\begin{bmatrix} d_{\max}^2 & \kappa_k(\hat{\mathbf{b}}_k^- + \mathbf{K}_k\mathbf{r}_k)^{\mathsf{T}} \\ \kappa_k(\hat{\mathbf{b}}_k^- + \mathbf{K}_k\mathbf{r}_k) & \mathbf{1} \end{bmatrix} \geq 0, \tag{6.59}$$

$$\kappa_k \mathbf{A}_{\text{con},k}(\hat{\mathbf{b}}_k^- + \mathbf{K}_k\mathbf{r}_k) \geq \boldsymbol{\xi}_{\text{con},k} - \boldsymbol{\zeta}_k. \tag{6.60}$$

The gain $\mathbf{K}_k$ is found by minimizing (6.56) subject to (6.58)-(6.60). The correction for $\mathbf{P}_k^-$ is still given by (6.46). Using $\hat{\mathbf{b}}_k$ from this section, $\mathbf{d}_k$ is determined using (6.48), and the desired attitude $\mathbf{C}_{d_{k+1}a}$ is once again found using (6.38).

6.6 Numerical Example

Three different extremum-seeking guidance algorithms are used for guidance of a spacecraft with a patch antenna. The guidance algorithms attempt to align the antenna with an unknown source, which could be from another spacecraft or from a ground station, to maximize received power, while satisfying inclusion and exclusion zones. For simplicity, in this numerical example, the position of the radiation source relative to the spacecraft does not change over time. The three Kalman-filter-based

guidance algorithms are 1) the unconstrained Kalman filter, 2) the gain-projected Kalman filter with inclusion and exclusion zones, and 3) the LMI-based Kalman filter with norm and attitude constraints.

6.6.1 Simulation Parameters

6.6.1.1 Spacecraft with Three Reaction Wheels

Consider a spacecraft with moment of inertia $\mathbf{J}_b^{\mathcal{B}c}$, equipped with three orthonormal reaction wheels that have a combined moment of inertia $\mathbf{J}_b^{\mathcal{W}}$, and rotation speed $\dot{\boldsymbol{\gamma}}$. The control input to the wheels is given by $\boldsymbol{\eta}$, a wheel acceleration. Frame $\mathcal{F}_a$ is inertial, $\mathcal{F}_b$ is the body frame of the spacecraft, and $\mathcal{F}_d$ is the desired body frame representing the desired spacecraft attitude. The DCM $\mathbf{C}_{ba}$ parameterizes the attitude of the spacecraft relative to $\mathcal{F}_a$, and is assumed to be known deterministically. In practice, $\mathbf{C}_{ba}$ can be estimated using using a suite of sensors such as a magnetometer, a sun sensor, a horizon sensor, and others. As in Section 4.2, the spacecraft's dynamics are given by [35, 55]

$$\mathbf{J}_b^{\mathcal{B}c}\dot{\boldsymbol{\omega}}_b^{ba} + \dot{\boldsymbol{\omega}}_b^{ba^\times}(\mathbf{J}_b^{\mathcal{B}c}\boldsymbol{\omega}_b^{ba} + \mathbf{J}_b^{\mathcal{W}}\dot{\boldsymbol{\gamma}}) + \mathbf{J}_b^{\mathcal{W}}\ddot{\boldsymbol{\gamma}} = \mathbf{0}, \tag{6.61}$$

$$\ddot{\boldsymbol{\gamma}} = -\boldsymbol{\eta}, \tag{6.62}$$

$$\dot{\mathbf{C}}_{ba} = -\boldsymbol{\omega}_b^{ba^\times}\mathbf{C}_{ba}. \tag{6.63}$$

In particular, the spacecraft's moment of inertia is $\mathbf{J}_b^{\mathcal{B}c} = \text{diag}\,(4, 4, 1)$ kg $\cdot$m^2, and the reaction wheel array's moment of inertia is $\mathbf{J}_b^{\mathcal{W}} = 0.04\mathbf{1}$ kg $\cdot$m^2. The initial conditions of the spacecraft are $\mathbf{C}_{ba}(0) = \mathbf{C}_2(60^\circ)\mathbf{C}_3(20^\circ)$, $\boldsymbol{\omega}_b^{ba}(0) = \mathbf{0}$ rad/s, and $\dot{\boldsymbol{\gamma}}(0) = \mathbf{0}$ rad/s.

6.6.1.2 Patch Antenna and Performance Function

The received power of an antenna, denoted by C, is given by [85, p. 554]

$$C = P_{\text{EIRP}} L_s L_a G_r \tag{6.64}$$

where P_{EIRP} is effective isotropic radiated power, L_s is the space loss, and L_a is the transmission path loss. For simplicity, let $P_{\text{EIRP}} L_s L_a = 1$ W, and thus $C = G_r \cdot 1$ W. Let the patch antenna be mounted on the spacecraft such that $\underrightarrow{b}^3$ associated with the body frame is normal to the patch antenna. The radiation pattern of the patch

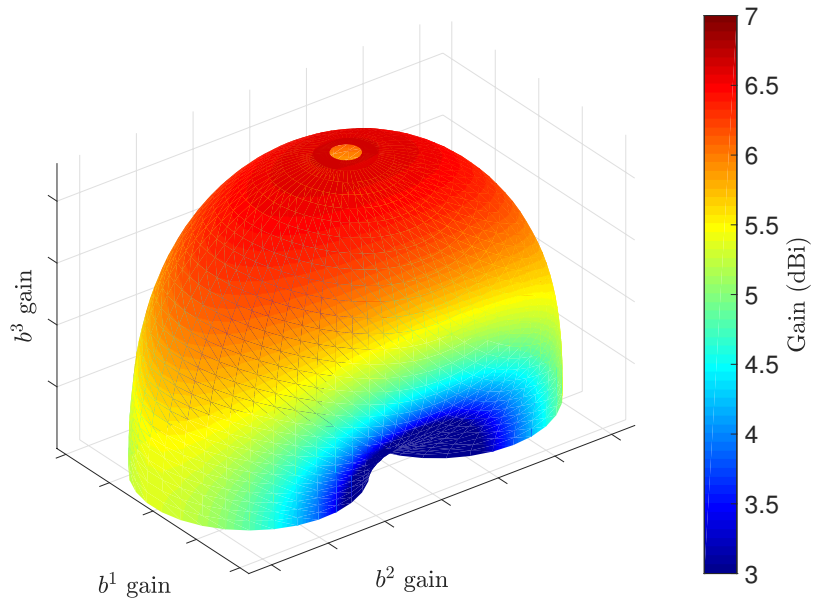


Figure 6.3: Radiation pattern of a patch antenna for the $m = 1$ and $n = 0$ mode [5, pp. 61–64]. Gain below 3 dBi not pictured for clarity.

antenna is given by the function $G_r(\theta, \phi)$ where θ and ϕ can be found from

$$\cos(\theta) = \mathbf{b}_a^{3^\mathsf{T}} \mathbf{a}_a^3,$$
$$\cos(\phi) = \mathbf{b}_a^{1^\mathsf{T}} \mathbf{a}_a^1,$$

where $\mathbf{a}_a^1 = \mathbf{1}_1$, $\mathbf{a}_a^3 = \mathbf{1}_3$, and $\mathbf{C}_{ba}^\mathsf{T} = \begin{bmatrix} \mathbf{b}_a^1 & \mathbf{b}_a^2 & \mathbf{b}_a^3 \end{bmatrix}$. The radiation pattern is shown in Fig. 6.3. The objective at hand is to maximize $J(\mathbf{C}_{ba}) = C(\mathbf{C}_{ba}) = G_r(\mathbf{C}_{ba}) \cdot 1$ (W), where J is all that is available to the guidance algorithm. To be clear, the guidance algorithm uses measurements of received power but does not know the explicit relationship between received power and attitude. The attitude that maximizes the antenna gain is given by $\mathbf{C}_{b^\star a}(\theta^\star, \phi^\star)$. This attitude corresponds to when $\underrightarrow{b}^3$ is aligned with $\underrightarrow{a}^3$ and, as such, $\theta^\star = 0°$ and $\phi^\star \in \mathbb{R}$. Since the guidance algorithm has no knowledge of where the radiation source is, the guidance algorithm has no knowledge of $\mathbf{C}_{b^\star a}$. The maximum performance is $J(\mathbf{C}_{b^\star a}) = 4.61$ W. The actual gradient of $J(\mathbf{C}_{ba})$ is determined numerically in simulation to determine the gradient estimate error. Received power is measured at 10 (Hz), and white noise with standard deviation of 0.05 W is added to each measurement of received power.

6.6.1.3 Attitude Constraints

Inclusion zones and exclusion zones often arise constraints on sensors or scientific payloads on-board the spacecraft. For example, a star tracker cannot point towards the Sun, an Earth surveillance sensor must always point towards the Earth, and solar cells should point towards the Sun. The exclusion and inclusion zones are given by

$$\mathbf{x}_b^1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \ \mathbf{y}_a^1 = -\begin{bmatrix} 0.3 \\ 1 \\ -.5 \end{bmatrix} \cdot \left\| \begin{bmatrix} 0.3 \\ 1 \\ -.5 \end{bmatrix} \right\|^{-1}, \ \alpha_1 = 180^\circ - 20^\circ, \tag{6.65}$$

$$\mathbf{x}_b^2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \ \mathbf{y}_a^2 = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix}, \ \alpha_2 = 20^\circ. \tag{6.66}$$

The constraint associated with $\mathbf{x}_b^1$, $\mathbf{y}_a^1$, and α_1 is the exclusion zone.

6.6.1.4 Guidance Algorithms

The guidance algorithms run at 2 Hz, have $N = 5$ measurements, with one measurement taken at a frequency of 5 Hz. These values are chosen so that the spacecraft has time to react to a change in desired attitude and so that enough new measurements of J can be made. The weight $\mathbf{s}_k$ is chosen as $\mathbf{s}_k = 10^5[1 \ \ 1 \ \ 1]^\mathsf{T}$, and the LMIs are solved using the MOSEK solver [56] with the YALMIP interface [57] in Matlab. The persistent excitation $\mathbf{w}_k^{\text{pe}}$ is set to $\mathbf{w}_k^{\text{pe}} = 0.001\sin(0.05\pi k)[1 \ \ 1 \ \ 1]^\mathsf{T}$ rad. As in [13], to ensure smooth commands to the attitude controller, the gradient estimate is filtered with a first-order low-pass filter with a cutoff frequency of 0.08 Hz.

The Kalman filter is used in a similar manner to filters used for parameter estimation. As such, the matrix $\mathbf{R}_k$ can be characterized using the noise of the received power measurements. A single element of $\mathbf{y}_k$ is the difference between two measurements of the objective function. Since the noise added to a measurement of the objective function has a standard deviation of 0.05 W, the standard deviation of noise of the difference of two measurements is $0.05\sqrt{2}$ W. Using this information and the update rate of the Kalman filter, $\mathbf{R}_k$ is set to $\mathbf{R}_k = 0.001\mathbf{1}$ W. It is more difficult to characterize the $\mathbf{Q}_{k-1}$ matrix. Various methods exist to tune $\mathbf{Q}_{k-1}$, such as a retrospective optimization [86], but trial and error is used in this chapter, where $\mathbf{Q}_{k-1} = 0.002\mathbf{1}$ W/rad. The initial error covariance estimate is $\mathbf{P}_0 = \mathbf{1}$ W/rad and the initial gradient estimate is $\hat{\mathbf{b}}_0 = \mathbf{0}$ W/rad.

The Kalman filters use a κ_k of 0.15, and the LMI-based Kalman filter uses the norm constraint (6.53) with $d_{\max} = 0.25$ W/rad. Considering the interpretation that the norm of $\mathbf{d}_k$ represents an angle of rotation in radians, $d_{\max} = 0.25$ W/rad corresponds to a maximum rotation of 14° for each iteration of the guidance algorithm, ensuring the small angle approximation used for the attitude constraints is always valid.

6.6.1.5 Control Algorithm

The desired attitude $\mathbf{C}_{d_k a}$ is fed to the discrete-time attitude controller, given by

$$\boldsymbol{\eta}_k = -\mathbf{J}_b^{\mathcal{W}^{-1}}(\mathbf{K}_\mathrm{d}\boldsymbol{\omega}_{b_k}^{b_k a} - \mathbf{K}_\mathrm{p}\boldsymbol{\mathcal{P}}_a(\mathbf{C}_{b_k d_k})^\vee), \tag{6.67}$$

where $\mathbf{C}_{b_k d_k} = \mathbf{C}_{b_k a}\mathbf{C}_{d_k a}^\mathsf{T}$ is the attitude error [35]. The attitude controller runs at 10 Hz, and thus $T = 0.1$ s. The gains of (6.67) are $\mathbf{K}_\mathrm{d} = 1.2\mathbf{1}$ N · m and $\mathbf{K}_\mathrm{p} = 0.96\mathbf{1}$ N · m/s. The controller is simulated in discrete time, and the dynamics of the spacecraft are simulated in continuous time.

6.6.2 Simulation Results

6.6.2.1 Unconstrained Kalman Filter

The simulation results with the unconstrained Kalman filter are shown in Fig. 6.4. Fig. 6.4a shows that J converges to 4.61 W, which corresponds to the maximum possible performance. Near $J = 4.61$ W, the magnitude of the gradient becomes quite small, and without $d_{\min}$, the convergence time would be much longer. The middle plot in Fig. 6.4a shows the values of the constraints. For the constraints to be satisfied, both lines must be greater than zero. A dotted black line is shown to emphasize the zero line on the plot. Fig. 6.4b shows that the gradient estimate error converges to zero and that the errors mostly remain within the $\pm 3\sigma$ bounds.

Fig. 6.7 shows a three-dimensional plot of the results from the unconstrained and the gain-projected Kalman filters. The red cone is the exclusion zone and the green cone is the inclusion zone. The red trace is the locus of points created by the tip of $\underrightarrow{x}^1$ from (6.65), and the green line is the locus of points created by $\underrightarrow{x}^2$ from (6.66). In the unconstrained formulation shown in Fig. 6.7a, the green line remains clear of the inclusion zone. As it happens, the red line does not venture in the exclusion zone, but there is no guarantee that this will happen given different initial conditions or noise characteristics. The black vectors form the basis for $\mathcal{F}_a$, and the dark-green/blue vectors are the bases for $\mathcal{F}_d$ and $\mathcal{F}_b$ respectively. However, since the controller drives

the error $\mathbf{C}_{bd}$ to $\mathbf{1}$, $\mathcal{F}_d$ and $\mathcal{F}_b$ are virtually indistinguishable in this plot.

6.6.2.2 Gain-Projected Kalman Filter with Attitude Constraints

The gain-projected Kalman filter considers the inclusion and exclusion zones from (6.65)-(6.66), and the results of this simulation are shown in Fig. 6.5. Fig. 6.5a shows that J converges to $J(\mathbf{C}_{b^\star a}) = 4.61$ W, the theoretical maximum of the constrained problem. In addition, the initial reaction of the extremum-seeking guidance algorithm is to ensure that the inclusion zone constraint is satisfied. As a result, $\tilde{\mathbf{b}}$ and $\mathbf{d}_k$ become large, and estimation of $\tilde{\mathbf{b}}$ is poor, as shown by $\tilde{\mathbf{b}}$ escaping the $\pm 3\sigma$ bounds in Fig. 6.5b. Once the attitude constraints are satisfied, the gradient estimates improve and the $\pm 3\sigma$ bounds are satisfied. Note that the gradient estimate errors are the difference between the constrained gradient estimate $\tilde{\mathbf{b}}_k$ and the constrained gradient $\mathbf{b}_k^{\mathrm{c}}$. The constrained gradient is obtained by constraining $\mathbf{b}_k$ using the same equation that is used for $\hat{\mathbf{b}}_k$, that is equation (6.50).

The three-dimensional plot of the gain-projected Kalman filter is shown in Fig. 6.7b. The green line begins outside the green cone and initially follows a trajectory perpendicular to the green cone. This part of the trajectory is when the guidance algorithm seeks to satisfy the attitude pointing constraints. The second part of the green line is when the guidance algorithm seeks to maximize $J(\mathbf{C}_{ba})$. In this plot $\underrightarrow{b}^3$ is not exactly aligned with $\underrightarrow{a}^3$, but this is expected since received power is not sensitive near the extremum, which is to say the gradient near the optimum is quite shallow.

Fig. 6.9 shows the performance of the attitude control law with the gain-projected Kalman filter simulation. Specifically, Fig. 6.9a shows that the angular velocity and attitude errors trend to zero, which indicates that $\mathbf{C}_{bd}$ does in fact trend to $\mathbf{1}$. Fig. 6.9b shows the reaction wheel rates and control torques, which also go to zero as there are no external torques acting on the spacecraft.

6.6.2.3 LMI-Based Kalman Filter with Norm and Attitude Constraints

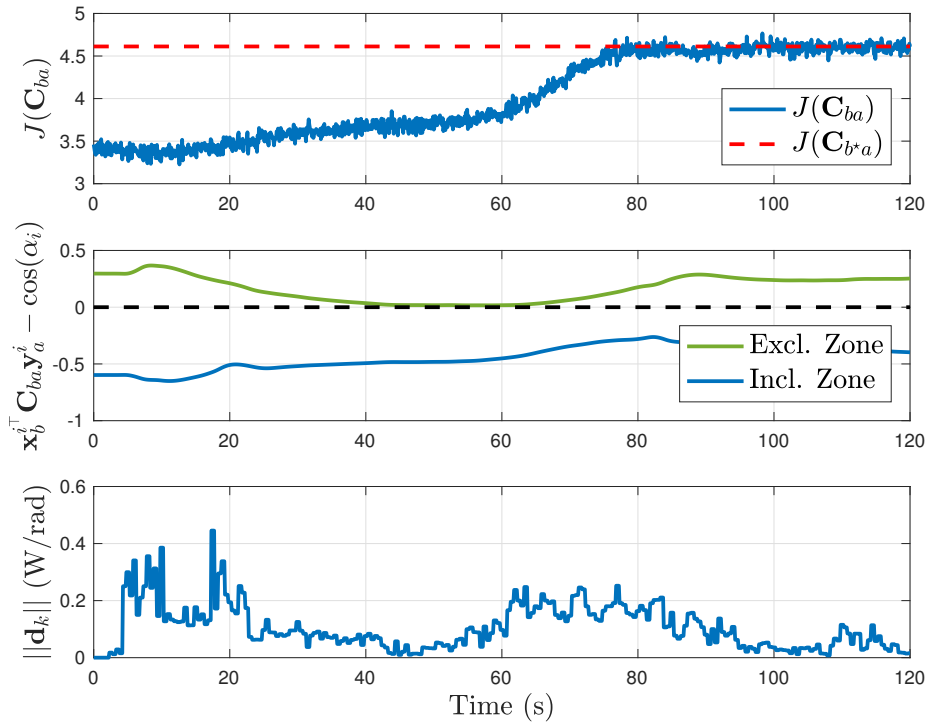
The third simulation uses the LMI-based Kalman filter, which considers both the attitude constraints (6.65)-(6.66), and the norm constraint (6.53). The results of this simulation are shown in Fig. 6.6, and the three-dimensional plot is shown in Fig. 6.8. The objective function converges to its constrained extremum in a slightly longer time than the gain-projected Kalman filter. In addition, the attitude constraints are not satisfied as quickly in this example since $\mathbf{d}_k$ is constrained. This is not a shortcoming of the LMI-based Kalman filter, but rather an advantage. The effect of

constraining $\mathbf{d}_k$ is shown by comparing Fig. 6.9 and Fig. 6.10. Constraining $\mathbf{d}_k$ results in a smoother commanded attitude, gradient estimates that remain within the $\pm 3\sigma$ bounds, and smoother and smaller control effort than with the projected-gain Kalman filter.

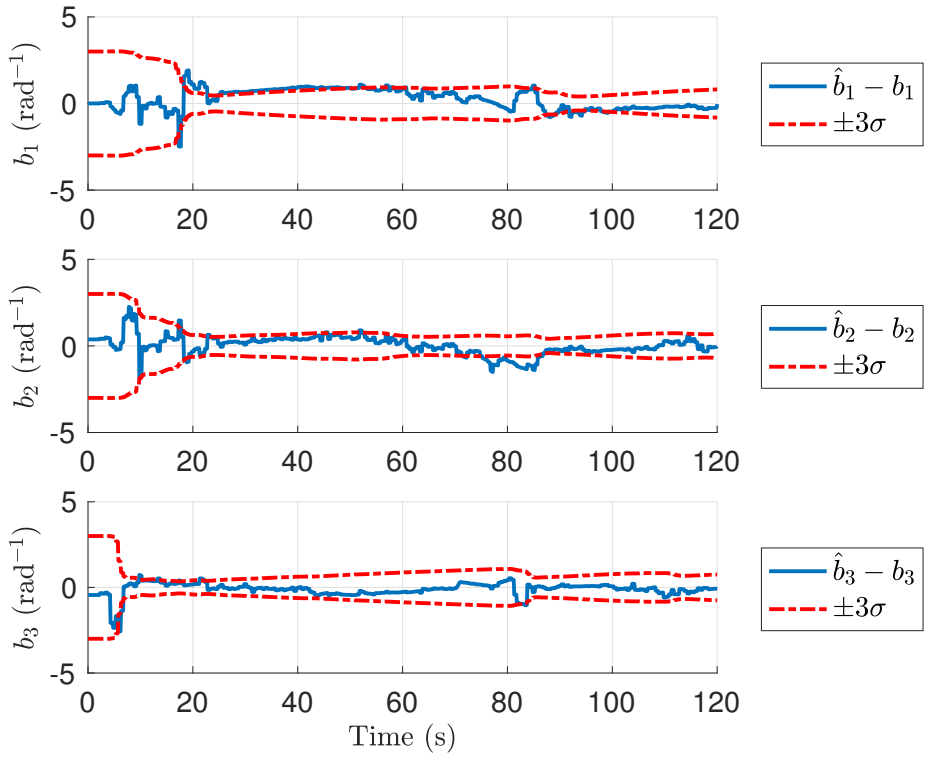
6.7 Closing Remarks

This chapter presents several useful results for unconstrained and constrained extremum-seeking guidance on $SO(3)$. First, the relationship between the gradient of a function on $SO(3)$ and the gradient of its Taylor series expansion is given. Second, it is shown how to transform inclusion and exclusion zone attitude constraints to a linear inequality constraint. Third, three different Kalman-filter-based extremum-seeking guidance algorithms are presented to estimate the gradient of an unknown function on $SO(3)$ to find an extremizing attitude. Each filter has its unique advantages, and can be chosen based on the level of complexity of the application. Numerical simulations are presented to demonstrate the effectiveness of the extremum-seeking guidance algorithms.

Several avenues exist to improve the proposed extremum-seeking algorithm. In the present chapter, the guidance law may cause the attitude controller to command unrealistic torques. To prevent this, the torques can either be saturated, or the guidance algorithm can be augmented with a model of the spacecraft dynamics. Using the spacecraft dynamics, torque limits can be written as a function of desired attitude using an LMI, which can be incorporated in the LMI-based Kalman filter. The resulting guidance algorithm will generate an attitude trajectory that will nominally keep control torques within acceptable limits. In addition, other optimization algorithms, such as conjugate-gradient methods, can improve the performance of the extremum-seeking algorithm. Finally, the transformation of inclusion and exclusion zone constraints to linear inequality constraints may prove to be a useful transformation for other areas of research, such as optimal control or path planning.

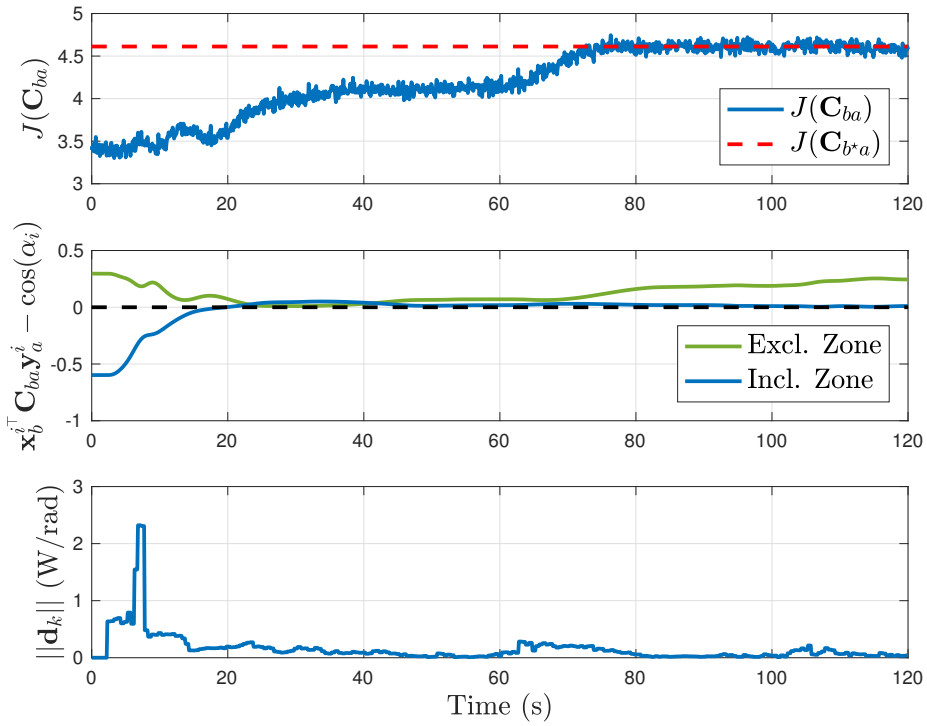


(a) Objective function, constraints, and step size. In middle plot, constraints are satisfied when they are above the black dashed line (i.e., above 0).

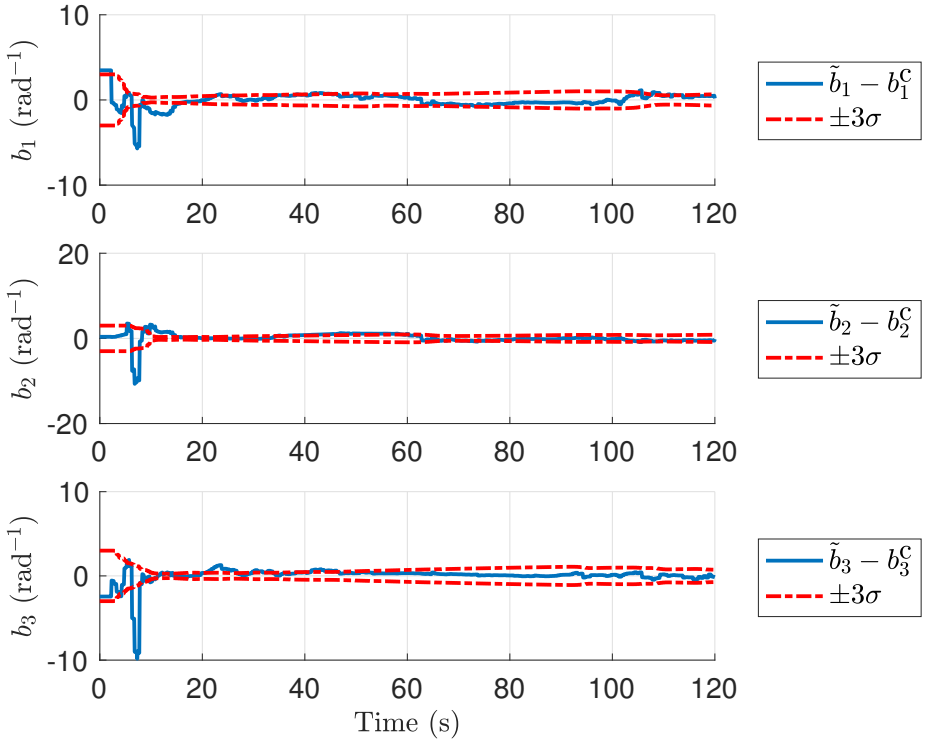


(b) Gradient estimate error with $\pm 3\sigma$ bounds.

Figure 6.4: Extremum-seeking enabled by the unconstrained Kalman filter, with no attitude constraints.

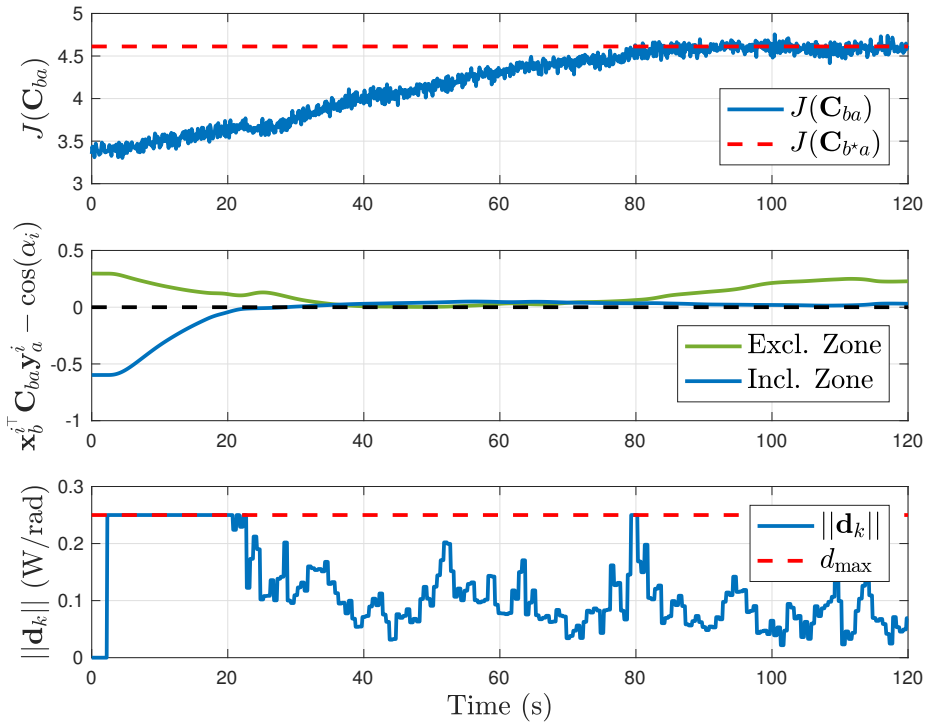


(a) Objective function, constraints, and step size. In middle plot, constraints are satisfied when they are above the black dashed line (i.e., above 0).

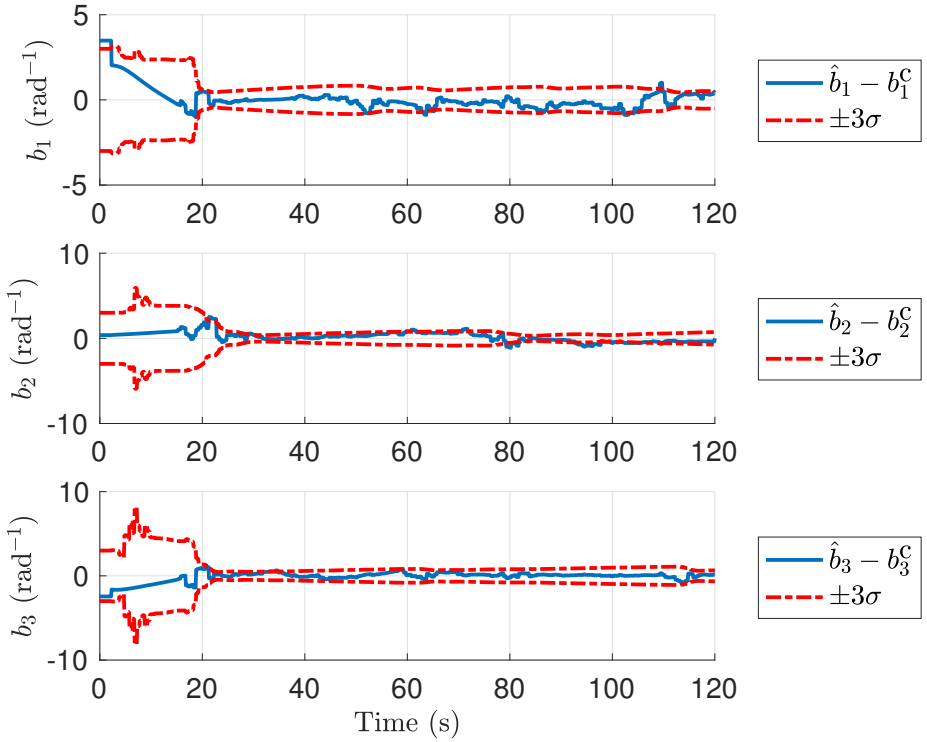


(b) Constrained gradient estimate error with $\pm 3\sigma$ bounds.

Figure 6.5: Extremum-seeking enabled by the gain-projected Kalman filter, with attitude inclusion and exclusion zones.

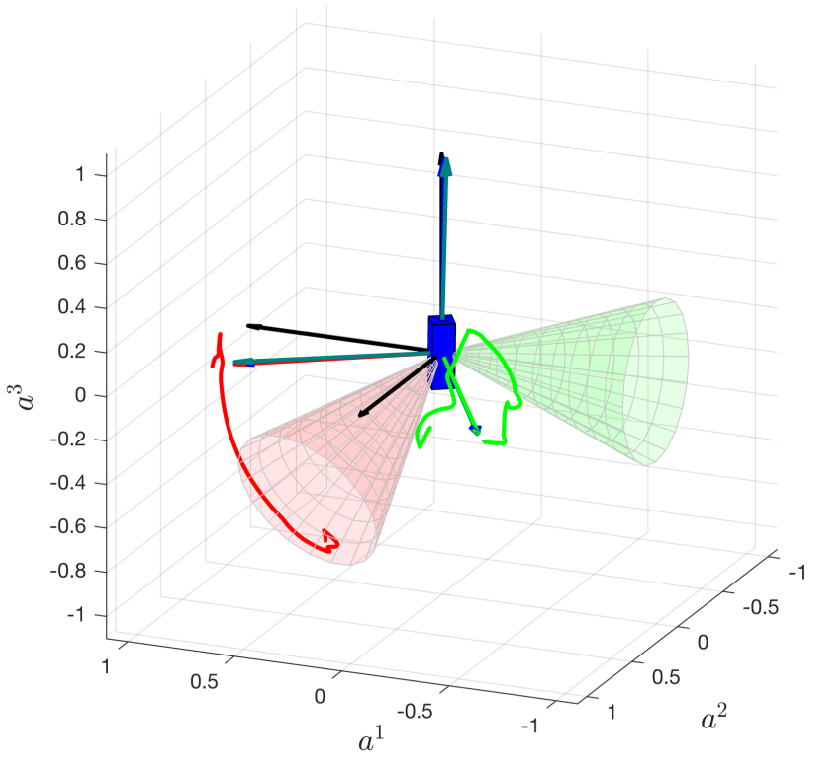


(a) Objective function, constraints, and step size. In middle plot, constraints are satisfied when they are above the black dashed line (i.e., above 0).

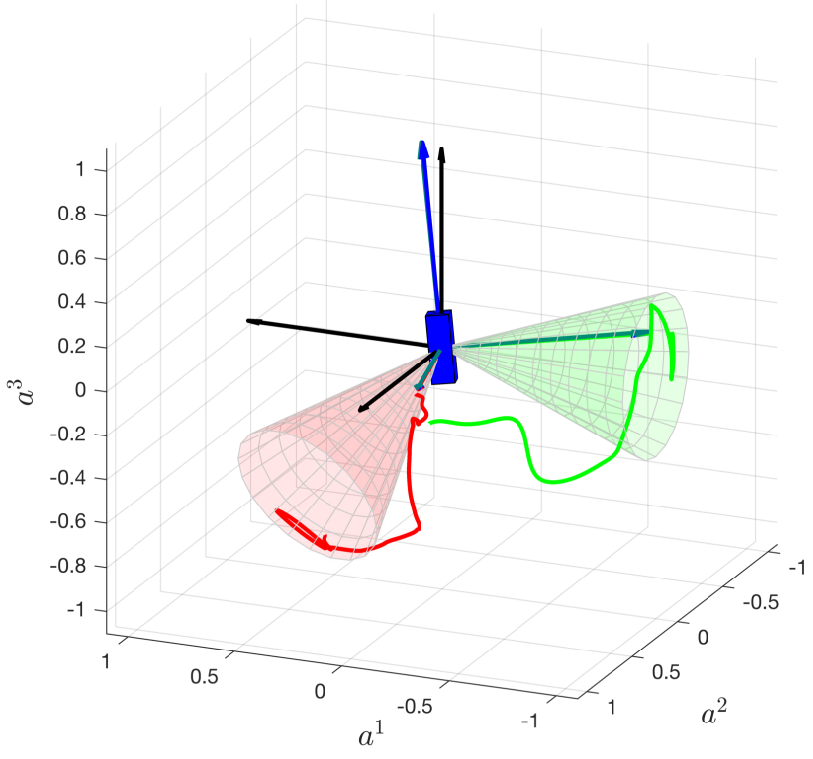


(b) Constrained gradient estimate error with $\pm 3\sigma$ bounds.

Figure 6.6: Extremum-seeking enabled by the LMI-based Kalman filter, with attitude inclusion and exclusion zones, and with norm constraint on $\mathbf{d}_k$.



(a) Extremum-seeking with unconstrained Kalman filter.



(b) Extremum-seeking with gain-projected Kalman filter and attitude constraints.

Figure 6.7: 3D-representation of extremum-seeking algorithms. Exclusion zone is shown in red on the left and inclusion zone shown in green on the right.

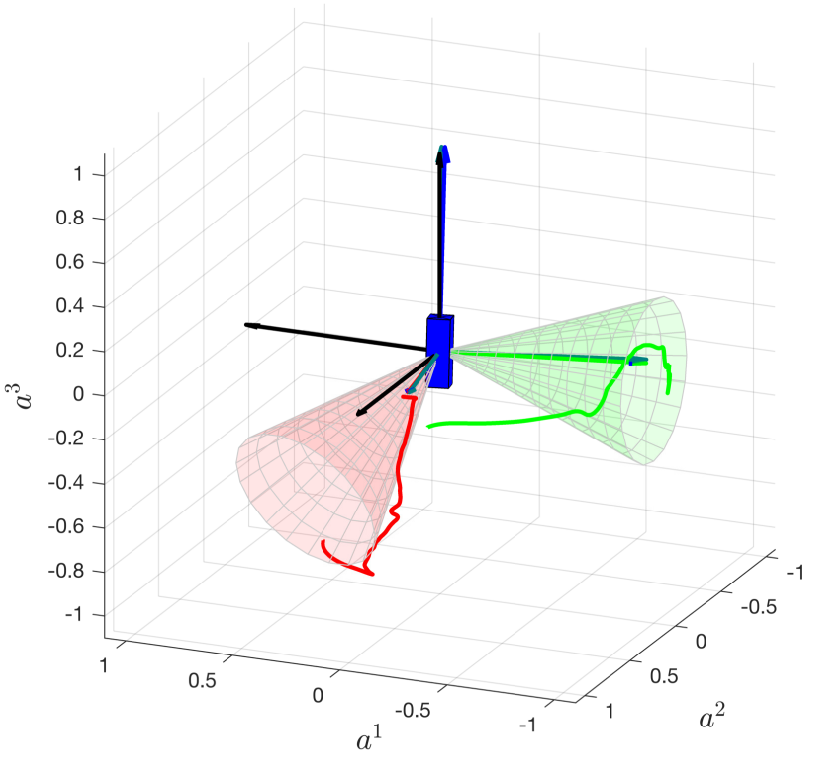
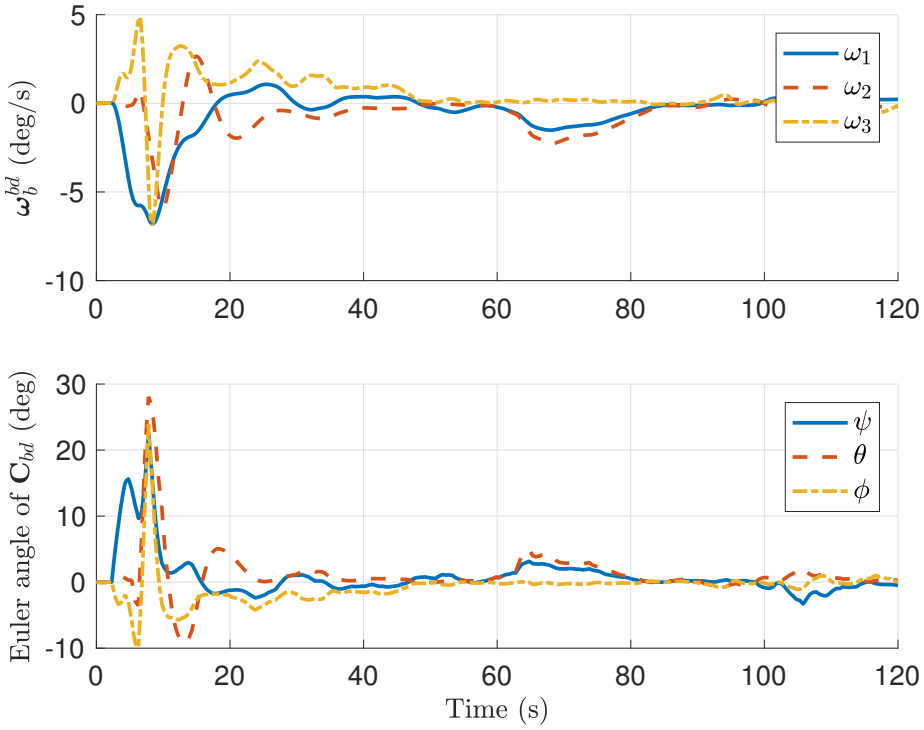
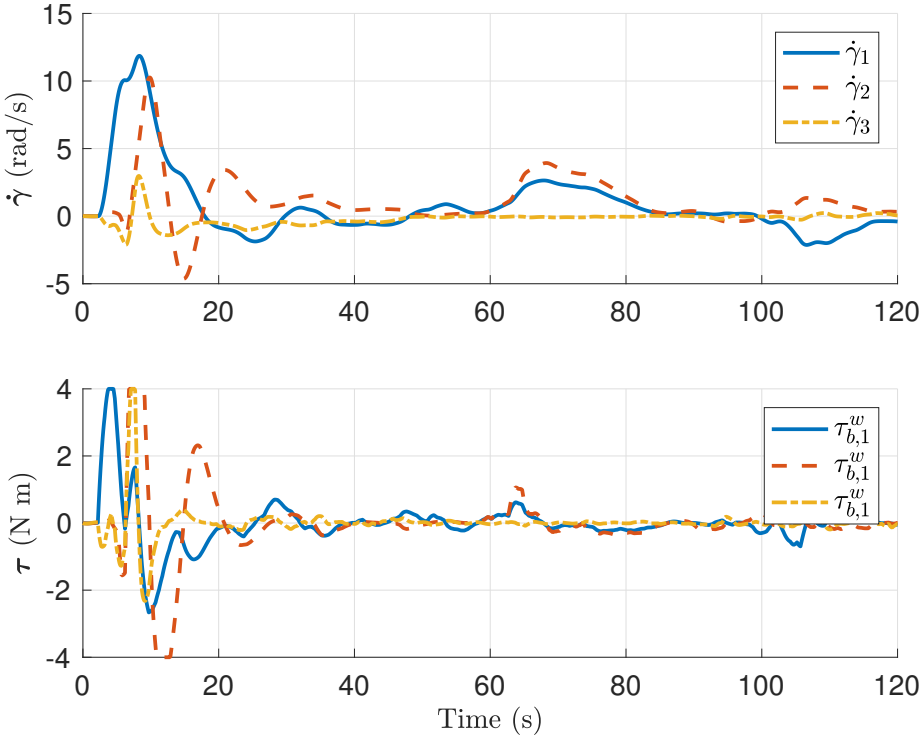


Figure 6.8: 3D-representation of extremum-seeking guidance with the LMI-based Kalman filter. Exclusion zone is shown in red on the left and inclusion zone shown in green on the right.

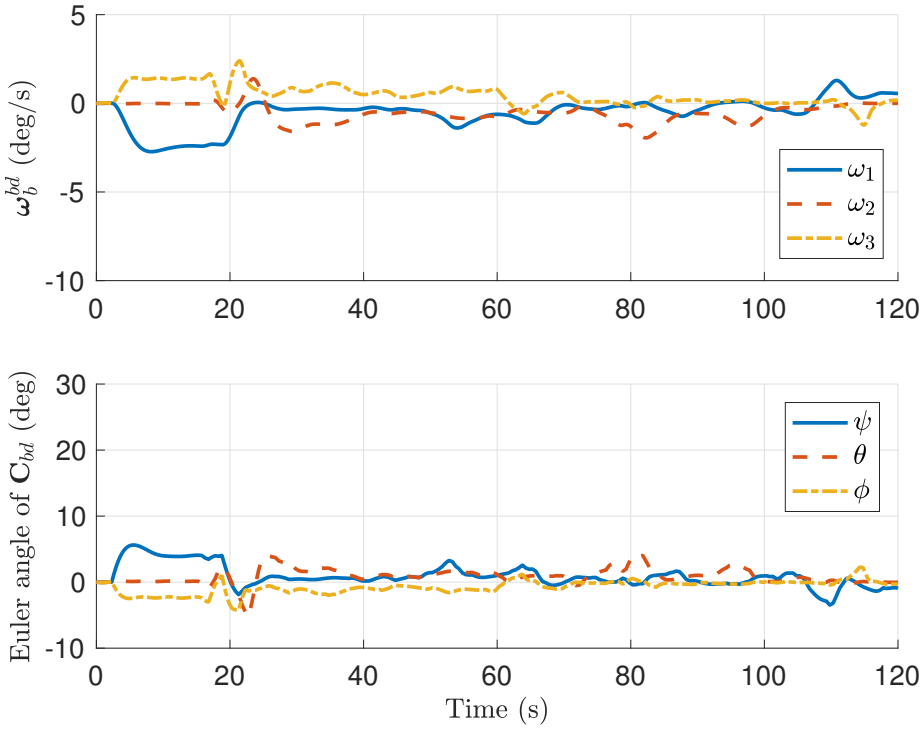


(a) Attitude and angular velocity errors.

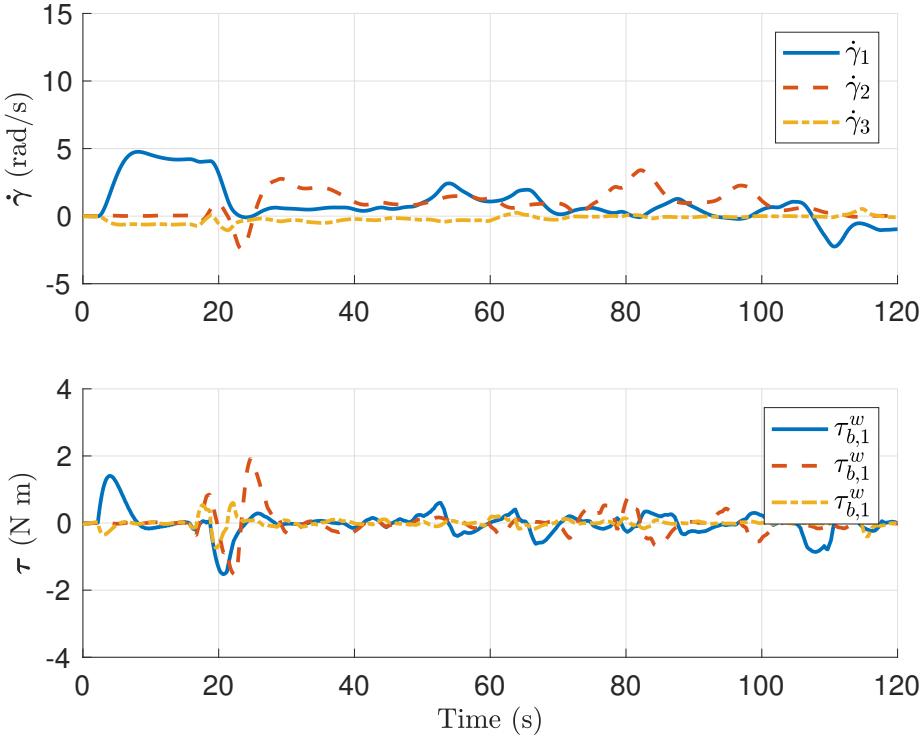


(b) Reaction wheel speeds and control torques.

Figure 6.9: Spacecraft state errors and control for gain-projected Kalman filter extremum-seeking simulation.



(a) Attitude and angular velocity errors.



(b) Reaction wheel speeds and control torques.

Figure 6.10: Spacecraft state errors and control for LMI-based Kalman filter extremum-seeking simulation.

Part III

Control of Nonlinear Passive and Conic Systems using Affine Controllers

Chapter 7

Review of Input-Output Theory

Input-output theory forms the basis for stability results in this part of this dissertation. Some definitions presented are applicable to all real inner product spaces, but this dissertation restricts itself to Lebesgue spaces. Much of the information is taken from [87, 27, 88, 26].

7.1 Gain

Definition 7.1 ($\mathcal{L}_2$ stable [87])**.** An operator $\boldsymbol{\mathcal{G}} : \mathcal{L}_{2e} \to \mathcal{L}_{2e}$ is $\mathcal{L}_2$ stable if $\boldsymbol{\mathcal{G}}\mathbf{u} \in \mathcal{L}_2$ for all $\mathbf{u} \in \mathcal{L}_2$.

Definition 7.2 (Finite $\mathcal{L}_2$ gain [87])**.** Consider an operator $\boldsymbol{\mathcal{G}} : \mathcal{L}_{2e} \to \mathcal{L}_{2e}$, with input $\mathbf{u} \in \mathcal{L}_{2e}$. The operator $\boldsymbol{\mathcal{G}}$ has finite $\mathcal{L}_2$ gain if there exists a constant γ and a function β such that

$$\|\boldsymbol{\mathcal{G}}\mathbf{u}\|_{2T} \leq \gamma \|\mathbf{u}\|_{2T} + \beta. \tag{7.1}$$

An operator that has finite $\mathcal{L}_2$ gain is $\mathcal{L}_2$ stable. If $\beta = 0$, then $\boldsymbol{\mathcal{G}}$ has finite $\mathcal{L}_2$ gain with zero bias [87, p. 4]. This property is also known as weakly finite-gain stable (WFGS) in [88, p. 191].

If $\boldsymbol{\mathcal{G}}$ is linear and $\beta = 0$, then (7.1) implies $\|\boldsymbol{\mathcal{G}}\|_\infty \leq \gamma$. Note that $\|\boldsymbol{\mathcal{G}}\|_\infty = \gamma$ when

$$\|\boldsymbol{\mathcal{G}}\|_\infty = \sup_{\mathbf{u} \in \mathcal{L}_{2e} \setminus \{\mathbf{0}\}} \frac{\|\boldsymbol{\mathcal{G}}\mathbf{u}\|_{2T}}{\|\mathbf{u}\|_{2T}}.$$

The Small Gain Theorem, written below in Theorem 7.3, is a key component to input-output stability theory. The Small Gain Theorem is applicable to operators that map from an extended normed linear subspace, but in this dissertation, it is

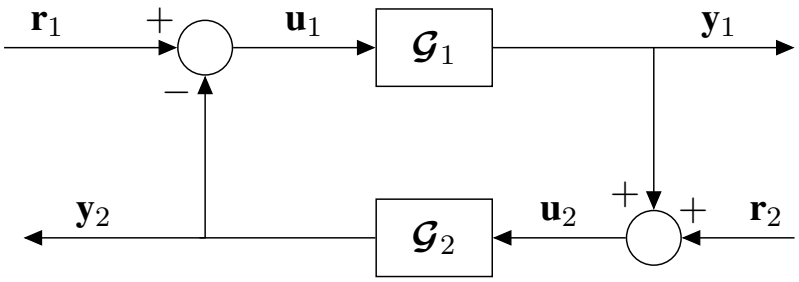


Figure 7.1: Negative feedback interconnection.

written only for $\mathcal{L}_{2e}$ since the more general version is not needed. In Theorem 7.3, an operator $\boldsymbol{\mathcal{G}}$ that maps $\mathcal{L}_{2e} \to \mathcal{L}_{2e}$ is written as $\mathbf{y} = \boldsymbol{\mathcal{G}}\mathbf{u}$, where $\mathbf{y}, \mathbf{u} \in \mathcal{L}_{2e}$.

Theorem 7.3 (Small Gain Theorem)**.** *Consider the systems $\boldsymbol{\mathcal{G}}_i : \mathcal{L}_{2e} \to \mathcal{L}_{2e}$ for $i \in \{1, 2\}$, where*

$$\mathbf{y}_1 = \boldsymbol{\mathcal{G}}_1\mathbf{u}_1, \qquad \mathbf{u}_1 = \mathbf{r}_1 - \mathbf{y}_2,$$
$$\mathbf{y}_2 = \boldsymbol{\mathcal{G}}_2\mathbf{u}_2, \qquad \mathbf{u}_2 = \mathbf{r}_2 + \mathbf{y}_1.$$

Suppose there are constants $\beta_i, \gamma_i \geq 0$ for $i \in \{1, 2\}$ such that for all $t \in \mathbb{R}_{\geq 0}$,

$$\|\mathbf{y}_i\|_T \leq \gamma_i \|\mathbf{u}_i\|_T + \beta_i.$$

If $\gamma_1\gamma_2 < 1$, $\mathbf{r}_1, \mathbf{r}_2 \in \mathcal{L}_2$, then $\mathbf{u}_1, \mathbf{u}_2, \mathbf{y}_1, \mathbf{y}_2 \in \mathcal{L}_2$. In other words, the closed-loop system $\mathbf{y} = \boldsymbol{\mathcal{G}}\mathbf{r}$, where $\mathbf{r} = [\mathbf{r}_1^\mathsf{T} \ \ \mathbf{r}_2^\mathsf{T}]^\mathsf{T}$ and $\mathbf{y} = [\mathbf{y}_1^\mathsf{T} \ \ \mathbf{y}_2^\mathsf{T}]^\mathsf{T}$, has finite $\mathcal{L}_2$ gain.

The Small Gain Theorem plays an important role for LPV control, as it used to design LPV controllers. The main disadvantage of using this method is when the plant gain is high, the controller gain must be low, and can lead to conservative controllers with poor closed-loop performance.

7.2 Passivity

Definition 7.4 (Passive, ISP, OSP, and VSP [28, 88])**.** A general square system with inputs $\mathbf{u} \in \mathcal{L}_{2e}$ and outputs $\mathbf{y} \in \mathcal{L}_{2e}$ mapped through the operator $\boldsymbol{\mathcal{G}} : \mathcal{L}_{2e} \to \mathcal{L}_{2e}$ is very strictly passive (VSP) if there exists constants β, $0 < \delta < \infty$, and $0 < \epsilon < \infty$ such that

$$\langle \mathbf{y}, \mathbf{u} \rangle_{2T} \geq \delta \|\mathbf{u}\|_{2T}^2 + \epsilon \|\mathbf{y}\|_{2T}^2 + \beta, \quad \forall \mathbf{u} \in \mathcal{L}_{2e}, \quad \forall T \in \mathbb{R}_{\geq 0}. \tag{7.2}$$

The system is

- input strictly passive (ISP) if $\delta \in \mathbb{R}_{>0}$ and $\epsilon = 0$,
- output strictly passive (OSP) if $\delta = 0$ and $\epsilon \in \mathbb{R}_{>0}$, and
- passive if $\delta = \epsilon = 0$.

Theorem 7.5 (Passivity Theorem [87])**.** *Consider the feedback interconnection in Fig. 7.1. If $\boldsymbol{\mathcal{G}}_i : \mathcal{L}_{2e} \to \mathcal{L}_{2e}$ for $i = 1, 2$ satisfies*

$$\langle \mathbf{y}_i, \mathbf{u}_i \rangle_{2T} \geq \delta_i \left\| \mathbf{u}_i \right\|_{2T}^2 + \epsilon_i \left\| \mathbf{y}_i \right\|_{2T}^2 + \beta, \quad \forall \mathbf{u}_i \in \mathcal{L}_{2e}, \quad \forall T \in \mathbb{R}_{\geq 0},$$

for some constant $\beta_i \in \mathbb{R}$. If $\epsilon_1 + \delta_2 > 0$, and $\epsilon_2 + \delta_1 > 0$, where ϵ_1 and δ_i may be negative, then the feedback interconnection has finite $\mathcal{L}_2$ gain.

Remark 7.6. To further understand Theorem 7.5, consider a passive plant in feedback interconnection with a controller. If the controller is ISP, then the closed-loop system is $\mathcal{L}_2$ stable, but if the controller is VSP, the closed-loop system has finite $\mathcal{L}_2$ gain. Closed-loop finite $\mathcal{L}_2$ gain is a motivating factor why a VSP controller is sought in Chapter 8.

Passivity properties hold for nonlinear operators, and it can be difficult to ascertain if an input-output map is passive. However, when considering a linear system, passivity and positive realness (PR) are synonymous, and the property of being very strictly passive and being strictly positive real (SPR) with positive definite feedthrough are synonymous. Lemma 7.7 provides conditions for a linear system to be SPR.

Lemma 7.7 (Kalman-Yakubovich-Popov (KYP) Lemma [89])**.** *Consider the transfer matrix $\mathbf{G}(s) = \mathbf{C}(s\mathbf{1} - \mathbf{A})^{-1}\mathbf{C} + \mathbf{D}$, where $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ forms a minimal state-space realization. The system $\mathbf{G}(s)$ is SPR if and only if there exists a $\mathbf{P} = \mathbf{P}^\mathsf{T} > 0$, an $\mathbf{L}$ and $\mathbf{W}$, and $\nu > 0$ such that*

$$\mathbf{P}\mathbf{A} + \mathbf{A}^\mathsf{T}\mathbf{P} = -\mathbf{L}^\mathsf{T}\mathbf{L} - 2\nu\mathbf{P} \tag{7.3}$$

$$\mathbf{P}\mathbf{B} = \mathbf{C}^\mathsf{T} - \mathbf{L}^\mathsf{T}\mathbf{W}, \tag{7.4}$$

$$\mathbf{D} + \mathbf{D}^\mathsf{T} = \mathbf{W}^\mathsf{T}\mathbf{W}. \tag{7.5}$$

If the above holds for $\nu = 0$, then $\mathbf{G}(s)$ is PR.

Lemma 7.7 is not directly used in this dissertation, but it is mentioned several times. In addition, (7.3)-(7.5) are used as inspiration for VSP controller synthesis. Several of the conditions to ensure an affine system is VSP are based on conditions similar to Lemma 7.7.

7.3 Conic Systems

There are many types of conic systems, including interior conic, exterior conic, degenerate conic, strictly interior and exterior conic, and strictly degenerate conic. Passive, ISP, OSP, and VSP systems are different cases of degenerate and strictly degenerate conic. Chapter 8 uses the Passivity Theorem directly, but Chapter 9 uses the Conic Sector Theorem for interior conic systems [21].

7.3.1 Definitions

Definition 7.8 (Interior Conic [90, 91])**.** A square system $\boldsymbol{\mathcal{G}} : \mathcal{L}_{2e} \to \mathcal{L}_{2e}$ that satisfies

$$-\|\boldsymbol{\mathcal{G}}\mathbf{u}\|_{2T}^2 + (a+b)\langle \boldsymbol{\mathcal{G}}\mathbf{u}, \mathbf{u}\rangle_{2T} - ab\|\mathbf{u}\|_{2T}^2 \geq \beta, \quad \forall \mathbf{u} \in \mathcal{L}_{2e}, \quad \forall T \in \mathbb{R}_{\geq 0}, \tag{7.6}$$

for $\beta \in \mathbb{R}$ and $a, b \in \mathbb{R}$, $a < b$, is interior conic with bounds a and b, denoted $\boldsymbol{\mathcal{G}} \in \text{cone}[a, b]$. The system $\boldsymbol{\mathcal{G}}$ is strictly interior conic, denoted $\boldsymbol{\mathcal{G}} \in \text{cone}(a, b)$, if (7.6) holds for bounds $a + \delta$ and $b - \delta$ for some $\delta > 0$.

Remark 7.9 (Gain). If $a = -\gamma$ and $b = \gamma$, then (7.6) reduces to (7.1).

Theorem 7.10 (Conic Sector Theorem [90, 91]). *Consider the negative feedback interconnection of two square systems, $\boldsymbol{\mathcal{G}}_1 : \mathcal{L}_{2e} \to \mathcal{L}_{2e}$ and $\boldsymbol{\mathcal{G}}_2 : \mathcal{L}_{2e} \to \mathcal{L}_{2e}$, where $\mathbf{y}_i = \boldsymbol{\mathcal{G}}_i \mathbf{u}_i$ for $i = 1, 2$. The closed-loop system with inputs $\mathbf{r} = [\mathbf{r}_1^\mathsf{T} \;\; \mathbf{r}_1^\mathsf{T}]^\mathsf{T}$ and outputs $\mathbf{y} = [\mathbf{y}_1^\mathsf{T} \;\; \mathbf{y}_1^\mathsf{T}]^\mathsf{T}$ is $\mathcal{L}_2$ stable if $\boldsymbol{\mathcal{G}}_1 \in \text{cone}[a, b]$ for $a < 0 < b$ and $\boldsymbol{\mathcal{G}}_2 \in \text{cone}(-\frac{1}{b}, -\frac{1}{a})$.*

Corollary 7.11 (Small Gain Theorem [88]). *Consider the system in Theorem 7.10, where $\boldsymbol{\mathcal{G}}_1 \in \text{cone}[-\gamma_1, \gamma_1]$ and $\boldsymbol{\mathcal{G}}_2 \in \text{cone}[-\gamma_2, \gamma_2]$. If $\gamma_1\gamma_2 < 1$ and $\mathbf{r} \in \mathcal{L}_2$, then $\mathbf{u} \in \mathcal{L}_2$ and $\mathbf{y} \in \mathcal{L}_2$.*

For linear systems with minimal state space realizations, LMI conditions exist to relate a plant to its conic bounds. These LMI conditions are known as the Conic Sector Lemma and the Modified Conic Sector Lemma. These two lemmas are not used in this dissertation, but analogous versions are presented in Section 9.2.2 for polytopic systems. As such, these two lemmas are presented within Theorem 7.12 below for completeness.

Theorem 7.12 (Conic Sector Lemma [92]). *Consider a square, asymptotically stable, LTI system, $\boldsymbol{\mathcal{G}} : \mathcal{L}_{2e} \to \mathcal{L}_{2e}$ with minimial state space realization $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$. For $a, b \in \mathbb{R}$, $a < b$, the following are equivalent.*

1. $\boldsymbol{\mathcal{G}} \in \text{cone}[a, b]$.

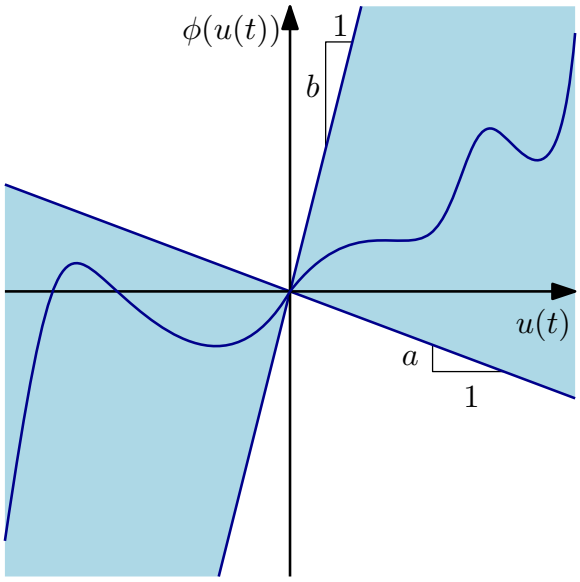


Figure 7.2: Conic bounds on a memoryless nonlinearity.

2. *There exists a* $\mathbf{P} = \mathbf{P}^\mathsf{T} > 0$*, and real matrices* $\mathbf{L}$*, and* $\mathbf{W}$ *such that*

$$\begin{aligned} \mathbf{P}\mathbf{A} + \mathbf{A}^\mathsf{T}\mathbf{P} + \mathbf{C}^\mathsf{T}\mathbf{C} &= -\mathbf{L}^\mathsf{T}\mathbf{L}, \\ \mathbf{P}\mathbf{B} - \frac{a+b}{2}\mathbf{C}^\mathsf{T} + \mathbf{C}^\mathsf{T}\mathbf{D} &= -\mathbf{L}^\mathsf{T}\mathbf{W}, \\ \mathbf{D}^\mathsf{T}\mathbf{D} - (a+b)\frac{\mathbf{D}+\mathbf{D}^\mathsf{T}}{2} + ab\mathbf{1} &= -\mathbf{W}^\mathsf{T}\mathbf{W}. \end{aligned}$$

3. *There exists a* $\mathbf{P} = \mathbf{P}^\mathsf{T} > 0$*, such that*

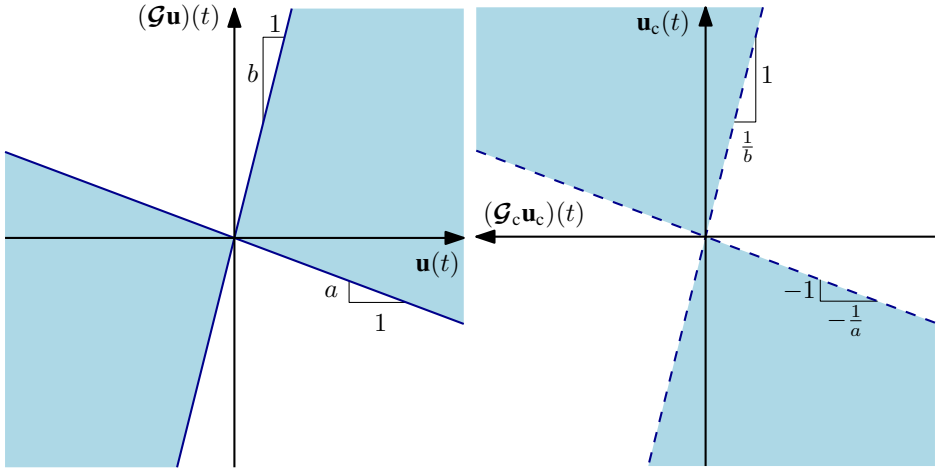
$$\begin{bmatrix} \mathbf{P}\mathbf{A} + \mathbf{A}^\mathsf{T}\mathbf{P} + \mathbf{C}^\mathsf{T}\mathbf{C} & \mathbf{P}\mathbf{B} - \frac{a+b}{2}\mathbf{C}^\mathsf{T} + \mathbf{C}^\mathsf{T}\mathbf{D} \\ \star & \mathbf{D}^\mathsf{T}\mathbf{D} - (a+b)\frac{\mathbf{D}+\mathbf{D}^\mathsf{T}}{2} + ab\mathbf{1} \end{bmatrix} \leq 0. \tag{7.7}$$

4. [Modified Conic Sector Lemma] *There exists a* $\mathbf{P} = \mathbf{P}^\mathsf{T} > 0$*, such that*

$$\begin{bmatrix} \mathbf{P}\mathbf{A} + \mathbf{A}^\mathsf{T}\mathbf{P} + \frac{1}{b}\mathbf{C}^\mathsf{T}\mathbf{C} & \mathbf{P}\mathbf{B} - \frac{1}{2}\left(\frac{a}{b}+1\right)\mathbf{C}^\mathsf{T} + \frac{1}{b}\mathbf{C}^\mathsf{T}\mathbf{D} \\ \star & \frac{1}{b}\mathbf{D}^\mathsf{T}\mathbf{D} - \left(\frac{a}{b}+1\right)\frac{\mathbf{D}+\mathbf{D}^\mathsf{T}}{2} + a\mathbf{1} \end{bmatrix} \leq 0. \tag{7.8}$$

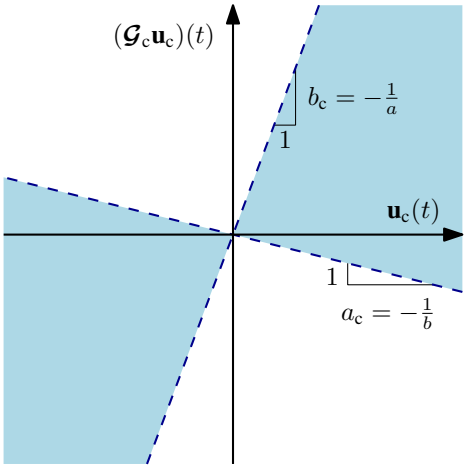
7.3.2 Graphical Interpretations of Conic Sectors

Graphical representations serve as a method to improve understanding of abstract notions, such as conic sectors and the Conic Sector Theorem. Consider the memoryless nonlinearity $\phi : \mathbb{R} \to \mathbb{R}$, such that $(\mathcal{G}u)(t) = \phi(u(t))$ for $t \in \mathbb{R}_{\geq 0}$, where $\mathcal{G} : \mathcal{L}_{2e} \to \mathcal{L}_{2e}$. If the trajectory $(u(t), \phi(u(t)))$ lies inside the sector bounded by two



(a) Conic plant sector, $\boldsymbol{\mathcal{G}} \in \mathrm{cone}[a, b]$.

(b) Plant conic sector complement, negative inverse of controller sector.



(c) Controller conic sector, $\boldsymbol{\mathcal{G}}_\mathrm{c} \in \mathrm{cone}(a_\mathrm{c}, b_\mathrm{c})$.

Figure 7.3: Visualization of the Conic Sector Theorem. Shaded areas are inside the specified conic sector. A solid line indicates a nonstrict conic bound and a dashed line indicates a strict conic bound. The visualization of the operator $\boldsymbol{\mathcal{G}}$ borrows from the visualization of a sector-bounded memoryless nonlinearity.

lines intersecting at the origin with slopes equal to a and b, then $\boldsymbol{\mathcal{G}} \in \mathrm{cone}[a, b]$, as shown in Fig. 7.2. A visualization of the Conic Sector Theorem is shown in Fig. 7.3. A controller (Fig. 7.3c) that guarantees input-output stability via the Conic Sector Theorem is one where the graph of the negative inverse of the controller lies in the plant's sector's complement (Fig. 7.3b).

Consider a stable, SISO LTI system $\mathcal{G} : \mathcal{L}_{2e} \to \mathcal{L}_{2e}$. The conic bounds of $\mathcal{G} : \mathcal{L}_{2e} \to \mathcal{L}_{2e}$ can be visualized on its Nyquist plot [90], as shown in Fig. 7.4. If $\mathcal{G} \in \mathrm{cone}[a, b]$, then the Nyquist plot of $\mathcal{G}$ lies entirely within the circle bounded on the real axis by

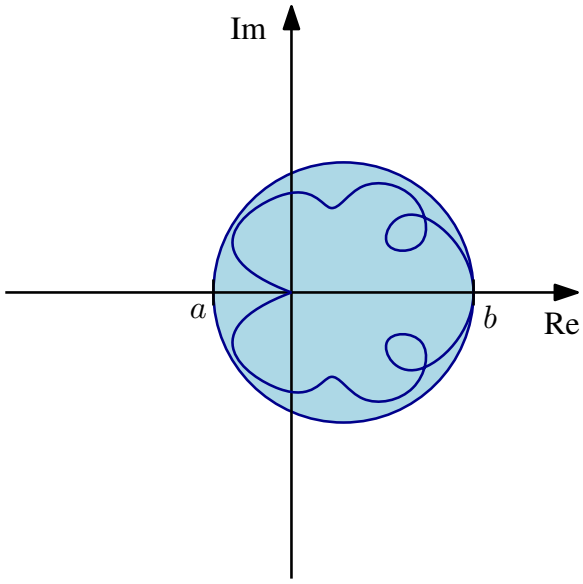


Figure 7.4: Nyquist of plot of plant $\mathcal{G} \in \text{cone}[a, b]$, with plot of circle defined by $\text{cone}[a, b]$. The shaded area represents $\text{cone}[a, b]$.

a and b. The Nyquist plot is useful when discerning how effective numerical methods are at determining conic bounds. The proximity of the circle defined by $\text{cone}[a, b]$ to the plot of $\mathcal{G}$ can give an indication of "tightness" of the bounds. If the plot of $\mathcal{G}$ does not lie within the circle defined by $\text{cone}[a, b]$, then it is also an efficient method to verify that $\mathcal{G} \notin \text{cone}[a, b]$.

7.4 LPV Systems

Linear parameter varying (LPV) control began as an extension to gain-scheduling by defining a system in state space form with state space matrices that are dependent on parameters [25, 93]. The goal of the LPV framework was to define new control methodologies that can take advantage of well-known linear control techniques, but that also provide closed-loop stability and performance guarantees of nonlinear systems. LPV systems are defined as

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{s})\mathbf{x} + \mathbf{B}(\mathbf{s})\mathbf{u}, \tag{7.9}$$

$$\mathbf{y} = \mathbf{C}(\mathbf{s})\mathbf{x} + \mathbf{D}(\mathbf{s})\mathbf{u}, \tag{7.10}$$

where $\mathbf{s}$ is a parameter that can be dependent on time. In the LPV literature, the parameter is often written as $\boldsymbol{\theta}$ or $\boldsymbol{\rho}$. In this dissertation, $\mathbf{s}$ is used due to its origin as a *scheduling signal*.

LPV systems are related to linear time-varying systems. A linear time-varying system is a linear system that depends only on time, a linear parameter-varying system depends on a exogenous variable or time, and a quasi-LPV system depends on state, exogenous variables, or time. Quasi-LPV systems that only depend on state are also known as state dependent coefficients [94]. In this dissertation, the signals $\mathbf{s} = \mathbf{s}(\boldsymbol{\sigma}(t), \mathbf{x}(t), t)$ can be dependent on time, external variables $\boldsymbol{\sigma}$, or the system state $\mathbf{x}$. In addition, the difference between LPV and quasi-LPV systems are not treated because how $\mathbf{s}$ changes has no effect on the results of this dissertation.

Definition 7.13 (Quadratic Stability). Let Ω be a set of all admissible $\mathbf{s}$. An LPV system is quadratically stable if there exists an $\mathbf{P}(\mathbf{s}) > 0$ such that

$$\mathbf{A}^{\mathsf{T}}(\mathbf{s})\mathbf{P}(\mathbf{s}) + \mathbf{P}(\mathbf{s})\mathbf{A}(\mathbf{s}) + \sum_{i=1}^{N} \dot{s}_i \frac{\partial \mathbf{P}(\mathbf{s})}{\partial s_i} < 0, \tag{7.11}$$

for all $\mathbf{s} \in \Omega$.

Equation (7.11) is infinite-dimensional since it must be evaluated for all $\mathbf{s} \in \Omega$, which poses a drawback to this formulation. There are several methods of circumventing this problem. A gridding solution can be accomplished by evaluating (7.11) at many $\mathbf{s}$ and $\dot{\mathbf{s}}$ combinations. A constant $\mathbf{P}$ can also be considered, which introduces conservatism in the result.

The $\mathcal{H}_\infty$ norm cannot be used for performance since an LPV system is parameter dependent and time varying. Instead, the induced $\mathcal{L}_2$ norm is used. The induced $\mathcal{L}_2$ norm for the system $\boldsymbol{\mathcal{G}} : \mathcal{L}_{2e} \to \mathcal{L}_{2e}$ defined by (7.9) and (7.10) is

$$\|\boldsymbol{\mathcal{G}}\|_{22} = \sup_{\mathbf{s}\in\Omega} \sup_{\mathbf{u}\in\mathcal{L}_{2e}\setminus\{\mathbf{0}\}} \frac{\|\boldsymbol{\mathcal{G}}\mathbf{u}\|_{2T}}{\|\mathbf{u}\|_{2T}}.$$

However, by abuse of language, the induced $\mathcal{L}_2$ norm for an LPV system is often written as the $\mathcal{H}_\infty$ gain $\|\boldsymbol{\mathcal{G}}\|_\infty$ of the system. A condition for $\mathcal{L}_2$ stability of an LPV system given a state-space realization is given by Theorem 7.14.

Theorem 7.14 (Generalized Bounded Real Lemma [26]). *Consider the system* $\boldsymbol{\mathcal{G}}$*, and that* $|\dot{s}_i| \leq \nu_i$*. If there exists a* $\mathbf{P}(\mathbf{s}) = \mathbf{P}^{\mathsf{T}}(\mathbf{s}) > 0$ *such that*

$$\begin{bmatrix} \mathbf{A}^{\mathsf{T}}(\mathbf{s})\mathbf{P}(\mathbf{s}) + \mathbf{P}(\mathbf{s})\mathbf{A}(\mathbf{s}) + \sum_{i=1}^{N} \nu_i \frac{\partial \mathbf{P}(\mathbf{s})}{\partial s_i} & \mathbf{P}(\mathbf{s})\mathbf{B}(\mathbf{s}) & \mathbf{C}^{\mathsf{T}}(\mathbf{s}) \\ \star & -\gamma\mathbf{1} & \mathbf{D}^{\mathsf{T}}(\mathbf{s}) \\ \star & \star & -\gamma\mathbf{1} \end{bmatrix} < 0, \tag{7.12}$$

then $\|\boldsymbol{\mathcal{G}}\|_{22} \leq \gamma$.

Theorem 7.14 is a state-space dependent result, and it is implied that the plant and closed-loop system can be written in LPV form. However, LPV systems are often approximations of nonlinear systems. Thus, any controllers synthesized using Theorem 7.14 do not actually have any closed-loop stability guarantees. Instead of using Theorem 7.14 for stability, this part focuses on using passive and conic input-output theory. The advantage of these input-output stability theories is that even if a plant LPV approximation is used for controller synthesis, closed-loop stability with the original nonlinear system can still be guaranteed if passive or conic bounds on the original nonlinear system are known.

Chapter 8

Very Strictly Passive Control with Affine Parameter Dependence

8.1 Introduction

Nonlinear systems that have a passive input-output map are pervasive, with examples including spacecraft, flexible robot manipulators, and electric circuits [15]. The passivity theorem states that the negative feedback interconnection of a very strictly passive (VSP) controller and a passive plant has finite $\mathcal{L}_2$ gain [88]. A strictly positive real (SPR) controller is often used in place of a VSP controller to control a nonlinear passive system. The reason is because the design of an SPR controller, which is a linear time-invariant (LTI) controller, is often more straightforward than the design of a nonlinear VSP controller [95]. However, an SPR controller optimally designed based on a linearization of the nonlinear passive system about a particular set point will not necessarily perform in an optimal way throughout the entire operating envelope of the nonlinear system. In order to improve performance and guarantee closed-loop $\mathcal{L}_2$ stability, a nonlinear VSP controller is sought. This chapter's focus is the analysis and synthesis of a particular class of nonlinear VSP controllers, which have an affine dependence on its parameters or scheduling signals.

When considering passive nonlinear systems, an LPV plant representation with affine parameter dependence can be used to design SPR subcontrollers that can be gain-scheduled [96, 95]. The SPR subcontrollers can be designed individually at the linearizations of each set point, $i = 1, \ldots, N$. The resulting system is VSP, and thus can stabilize a passive nonlinear system via the Passivity Theorem. However, unlike LPV controllers, the number of states associated with [96, 95] required to be integrated on-line in real time increases with $\mathcal{O}(N)$, which may not be acceptable

for some applications. Other attempts at passivity-based gain-scheduling requires a constant dynamics matrix [97].

This chapter introduces a controller that is a major step towards closing the gap between VSP gain-scheduled controllers and LPV controllers. The proposed controller is referred to as the VSP-LPV controller, and improves on both control paradigms. For example, the passivity-based gain-scheduled controller [96, 95], has in increasing number of integrating states for each set point. In [97, 98], the controllers have a fixed number of integrating states with each additional set point, but the controller dynamics matrix $\mathbf{A}_c$ is not a function of any parameter. The proposed VSP-LPV controller has a parameter dependent $\mathbf{A}_c$ and has a constant number of integrating states versus number of set points. The closest related work in the LPV-passivity framework are controllers that deal with integral quadratic constraints (IQCs) [99]. However, within the LPV framework, the IQC is placed on the closed-loop system, such that $\mathcal{L}_2$ input-output stability is achieved [100]. In addition, the Passivity Theorem is not used explicitly when using IQCs. This chapter constrains the controller and not the closed-loop system, and uses the Passivity Theorem for closed-loop stability results.

The synthesis of LPV controllers differs from traditional gain-scheduling. LPV controllers require a predefined synthesis method and are directly synthesized "en-masse" [24]. Characterization of stability and controller synthesis in the LPV framework is done through linear matrix inequalities (LMIs), which can be solved numerically very efficiently [31]. Traditional gain-scheduled controllers synthesize subcontrollers at specific set points, and then linearly interpolate or switch between the subcontrollers using scheduling signals [101].

The proposed VSP-LPV controller has features from traditional gain-scheduling and LPV control synthesis. An affine parameter LPV representation of the nonlinear passive system is required. As is discussed in Section 8.3, subcontrollers are synthesized at each set point using $\mathcal{H}_2$ or $\mathcal{H}_\infty$ control techniques, which allows for the flexibility to tune the controller at each set point. The second step ensures the controller is VSP, and is done "en-masse" using convex optimization with LMI constraints. Much like the polytopic LPV framework, where a constant Lyapunov matrix is used for controller synthesis [22], the VSP-LPV controller requires a constant Lyapunov matrix to guarantee the desired VSP property. The resulting VSP-LPV controllers are referred to as the $\mathcal{H}_2$-VSP and $\mathcal{H}_\infty$-VSP controllers so that their synthesis methods can be identified.

The VSP-LPV controller has similar form to the self-scheduled LPV controller found in [102]. As such, this chapter provides experimental results comparing the per-

formance of the VSP-LPV controller with $\mathcal{H}_2$-VSP and $\mathcal{H}_\infty$-VSP synthesis techniques to the controller from [102]. The controller from [102] is designed for a polytopic plant instead of an affine plant. However, a plant that is affine can be transformed to a polytopic form if lower and upper bounds on the scheduling parameters are known [22]. In the experimental results, the scheduling signals are such that no affine to polytopic transformation is needed.

This chapter has three main contributions. First, the VSP-LPV controller that guarantees stability of passive nonlinear systems is presented. Since the VSP-LPV controller is dependent on the Passivity Theorem for stability results, using an approximate LPV model is sufficient for controller design. Second, the $\mathcal{H}_2$-VSP and $\mathcal{H}_\infty$-VSP synthesis methods are described in Section 8.3. Third, the VSP-LPV controller using the $\mathcal{H}_2$-VSP and $\mathcal{H}_\infty$-VSP synthesis techniques are compared experimentally to the LPV controller from [102] by using each controller for trajectory tracking of a flexible-joint two-link manipulator. The VSP-LPV controller is presented in Section 8.2 with synthesis in Section 8.3. The application example with experimental results is in Section 8.4, followed by closing remarks in Section 8.5.

8.2 Very Strictly Passive Linear Parameter Varying Controller

8.2.1 Plant Description

Consider the nonlinear system $\boldsymbol{\mathcal{G}}$ defined by

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{w}, \mathbf{u}), \tag{8.1}$$

$$\mathbf{z} = \mathbf{g}(\mathbf{x}, \mathbf{w}, \mathbf{u}), \tag{8.2}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{w}, \mathbf{u}), \tag{8.3}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the system state, $\mathbf{w} \in \mathbb{R}^{n_w}$ is the disturbance or exogenous signal, $\mathbf{u} \in \mathbb{R}^m$ is the control input, $\mathbf{z} \in \mathbb{R}^{n_z}$ is the performance variable and $\mathbf{y} \in \mathbb{R}^m$ is the measurement signal. The map $\mathbf{u} \mapsto \mathbf{y}$ is passive. The map $\mathbf{w} \mapsto \mathbf{z}$ is often used for to asses performance and for synthesizing controllers, and is not necessarily passive. For controller synthesis, the nonlinear system (8.1)-(8.3) must be written in affine parameter form. The plant's LPV representation may be exact or approximate, and scheduling signals may be functions of state or exogenous signals. Regardless of what LPV plant representation is used, as long as the representation is affine in the

parameters (subject to some small constraints on the parameters), closed-loop $\mathcal{L}_2$ stability of the VSP-LPV controller connected in a negative feedback interconnection with this passive (or input strictly passive, or output strictly passive, or VSP) plant is guaranteed. Stability of the LPV plant representation in closed-loop with the controller is not investigated because it is the nonlinear plant that is under control in practice, not the LPV representation of the nonlinear plant.

Writing (8.1)-(8.3) in affine parameter form yields

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{s})\mathbf{x} + \mathbf{B}_1(\mathbf{s})\mathbf{w} + \mathbf{B}_2(\mathbf{s})\mathbf{u}, \tag{8.4}$$

$$\mathbf{z} = \mathbf{C}_1(\mathbf{s})\mathbf{x} + \mathbf{D}_{11}(\mathbf{s})\mathbf{w} + \mathbf{D}_{12}(\mathbf{s})\mathbf{u}, \tag{8.5}$$

$$\mathbf{y} = \mathbf{C}_2(\mathbf{s})\mathbf{x} + \mathbf{D}_{21}(\mathbf{s})\mathbf{w} + \mathbf{D}_{22}(\mathbf{s})\mathbf{u}, \tag{8.6}$$

where $\mathbf{s} = [s_1 \ \cdots \ s_N]^\mathsf{T}$ are the scheduling signals, and

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 \\ \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{D}_{22} \end{bmatrix} = \sum_{i=1}^{N} s_i \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_{1,i} & \mathbf{B}_{2,i} \\ \mathbf{C}_{1,i} & \mathbf{D}_{11,i} & \mathbf{D}_{12,i} \\ \mathbf{C}_{2,i} & \mathbf{D}_{21,i} & \mathbf{D}_{22_i} \end{bmatrix}. \tag{8.7}$$

Each $s_i = s_i(\boldsymbol{\sigma}(\cdot), \mathbf{x}, \cdot)$ can be dependent on time, external variables $\boldsymbol{\sigma}$, or the system state $\mathbf{x}$. For some $\alpha \in \mathbb{R}_{>0}$, it is assumed that the signals satisfy

$$\sum_{i=1}^{N} s_i^2(\boldsymbol{\sigma}, \mathbf{x}, t) \geq \alpha > 0 \quad \begin{cases} s_i(\boldsymbol{\sigma}, \mathbf{x}, t) \in \mathcal{L}_{2e}, & \forall \boldsymbol{\sigma}, \mathbf{x} \ \forall t \geq 0, \\ s_i(\boldsymbol{\sigma}, \mathbf{x}, t) \in \mathcal{L}_\infty, & \forall \boldsymbol{\sigma}, \mathbf{x} \in \mathcal{L}_\infty, \\ s_i(\boldsymbol{\sigma}, \mathbf{x}, t) \geq 0, \end{cases} \tag{8.8}$$

guaranteeing that the map $\mathbf{u} \mapsto \mathbf{y}$ exists, the scheduling signals are square integrable on any finite time interval, and the time dependence of each signal is bounded. There is no restriction on the time rate of change of the scheduling signals. Unlike other passivity-based gain-scheduling schemes [96, 95], scheduling signals must be positive. In [96, 95], scheduling signals only affect the input and output of the controller. Since both the input and output of the controller are scaled by the scheduling signals, a negative sign on the signal retains the structure of the inner product, that is $\langle \mathbf{u}, \mathbf{y} \rangle_T = \langle -\mathbf{u}, -\mathbf{y} \rangle_T$. Here, the dynamics matrix is also a function of the scheduling signal, and a negative scheduling signal can change the system's input-output properties. If bounds on negative scheduling signals are known, the system can be redefined so that scheduling signals are positive.

Some LPV systems are already described with an affine parameter representa-

tion [22]. One method of transforming (8.1)-(8.3) to (8.4)-(8.6) is by linearizing (8.1)-(8.3) at set points $(\mathbf{x}_i, \mathbf{w}_i, \mathbf{u}_i)$, which yields a realization for $\boldsymbol{\mathcal{G}}_i$ of the form

$$\begin{bmatrix} \delta\dot{\mathbf{x}} \\ \delta\mathbf{z} \\ \delta\mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_{1,i} & \mathbf{B}_{2,i} \\ \mathbf{C}_{1,i} & \mathbf{D}_{11,i} & \mathbf{D}_{12,i} \\ \mathbf{C}_{2,i} & \mathbf{D}_{21,i} & \mathbf{D}_{22,i} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x} \\ \delta\mathbf{w} \\ \delta\mathbf{u} \end{bmatrix}, \tag{8.9}$$

where $\delta(\cdot) = (\cdot) - (\cdot)_i$. The dynamics at state $\mathbf{x}$ can be defined by using (8.9) in (8.7) by linearly interpolating between different $\boldsymbol{\mathcal{G}}_i$ using the scheduling signals s_i. Furthermore, the plant in (8.4)-(8.6) is polytopic if each $s_i \geq 0$ and $\sum_{i=1}^{N} s_i = 1$. If bounds on s_i are known, an affine plant representation can be transformed to a polytopic plant representation [22]. The VSP-LPV controller requires a plant with an affine parameter representation (8.4)-(8.6) for controller synthesis, but the stability results are valid for the nonlinear system (8.1)-(8.3).

8.2.2 VSP-LPV Controller

The main contribution of this chapter is a VSP-LPV controller $\mathbf{y}_\mathrm{c} = \boldsymbol{\mathcal{G}}_\mathrm{c}\mathbf{u}_\mathrm{c}$ that is designed for passive plant (8.1)-(8.3) using (8.4)-(8.6). The controller's input is $\mathbf{y} = \mathbf{u}_\mathrm{c} \in \mathbb{R}^m$ and output is $\mathbf{u} = -\mathbf{y}_\mathrm{c} \in \mathbb{R}^m$. The VSP nature of $\boldsymbol{\mathcal{G}}_\mathrm{c}$ is proven in Theorem 8.1. The controller $\boldsymbol{\mathcal{G}}_\mathrm{c}$ has the state-space form

$$\dot{\mathbf{x}}_\mathrm{c} = \mathbf{A}_\mathrm{c}(\mathbf{s})\mathbf{x}_\mathrm{c} + \mathbf{B}_\mathrm{c}(\mathbf{s})\left(\mathbf{u}_\mathrm{c} - \bar{\epsilon}\mathbf{C}_\mathrm{c}(\mathbf{s})\mathbf{x}_\mathrm{c}\right), \tag{8.10}$$

$$\mathbf{y}_\mathrm{c} = \mathbf{C}_\mathrm{c}(\mathbf{s})\mathbf{x}_\mathrm{c} + \mathbf{D}_\mathrm{c}\mathbf{u}_\mathrm{c}, \tag{8.11}$$

where $\mathbf{x}_\mathrm{c} \in \mathbb{R}^{n_\mathrm{c}}$ is the controller state, $\mathbf{D}_\mathrm{c} = \bar{\delta}\mathbf{1}$, $\bar{\epsilon}, \bar{\delta} \in \mathbb{R}_{>0}$ and $\mathbf{s} = [s_1 \quad \cdots \quad s_N]^\mathsf{T}$, $N \in \mathbb{N}_{>0}$. The matrices $\mathbf{A}_\mathrm{c}(\mathbf{s})$, $\mathbf{B}_\mathrm{c}(\mathbf{s})$, and $\mathbf{C}_\mathrm{c}(\mathbf{s})$ are affine in $\mathbf{s}$ and are written as

$$\begin{bmatrix} \mathbf{A}_\mathrm{c}(\mathbf{s}) & \mathbf{B}_\mathrm{c}(\mathbf{s}) \\ \mathbf{C}_\mathrm{c}(\mathbf{s}) & \mathbf{D}_\mathrm{c} \end{bmatrix} = \mathrm{diag}\left\{\mathbf{0}, \bar{\delta}\mathbf{1}\right\} + \sum_{i=1}^{N} s_i \begin{bmatrix} \mathbf{A}_{\mathrm{c},i} & \mathbf{B}_{\mathrm{c},i} \\ \mathbf{C}_{\mathrm{c},i} & \mathbf{0} \end{bmatrix}. \tag{8.12}$$

Each $\mathbf{A}_{\mathrm{c},i}$ is assumed Hurwitz and satisfies the LMI

$$\mathbf{P}_\mathrm{c}\mathbf{A}_{\mathrm{c},i} + \mathbf{A}_{\mathrm{c},i}^\mathsf{T}\mathbf{P}_\mathrm{c} < 0, \quad i = 1, \ldots, N, \tag{8.13}$$

where $\mathbf{P}_\mathrm{c} = \mathbf{P}_\mathrm{c}^\mathsf{T} > 0$. Equation (8.13) is equivalent to

$$\mathbf{P}_\mathrm{c}\mathbf{A}_{\mathrm{c},i} + \mathbf{A}_{\mathrm{c},i}^\mathsf{T}\mathbf{P}_\mathrm{c} = -\mathbf{Q}_{\mathrm{c},i}, \tag{8.14}$$

where $\mathbf{Q}_{c,i} = \mathbf{Q}_{c,i}^{\mathsf{T}} > 0$. Each pair $(\mathbf{B}_{c,i}, \mathbf{C}_{c,i})$ must satisfy

$$\mathbf{P}_c \mathbf{B}_{c,i} = \mathbf{C}_{c,i}^{\mathsf{T}}. \tag{8.15}$$

Due to the linearity of (8.14), a matrix $\mathbf{Q}_c = \mathbf{Q}_c^{\mathsf{T}} > 0$ can be constructed such that $\mathbf{P}_c \mathbf{A}_c + \mathbf{A}_c^{\mathsf{T}} \mathbf{P}_c = -\mathbf{Q}_c$ where $\mathbf{Q}_c = \sum_{i=1}^{N} s_i \mathbf{Q}_{c,i}$. In (8.13), the matrix $\mathbf{P}_c$ is a common Lyapunov matrix that is the same for each $i = 1, \ldots, N$. The existence of common Lyapunov matrices has been given much attention [103], and conditions for Hurwitz matrices, 2×2, and 3×3 complex matrices can be found in [104]. Generally, the conditions for the existence of a common Lyapunov matrix remains an open problem. Nevertheless, these are still commonly used in LPV control [22].

With a nonzero $\bar{\epsilon}$, the effective controller is

$$\dot{\mathbf{x}}_c = (\mathbf{A}_c(\mathbf{s}) - \bar{\epsilon}\mathbf{B}_c(\mathbf{s})\mathbf{C}_c(\mathbf{s}))\mathbf{x}_c + \mathbf{B}_c(\mathbf{s})\mathbf{u}_c,$$

and the effective controller dynamics matrix is $(\mathbf{A}_c(\mathbf{s}) - \bar{\epsilon}\mathbf{B}_c(\mathbf{s})\mathbf{C}_c(\mathbf{s}))$. The role of $\bar{\epsilon}$ is to ensure that the controller is at least OSP. When combined with a nonzero $\bar{\delta}$, the controller is VSP and thus the controller is guaranteed to stabilize a passive plant via the Passivity Theorem. The VSP property of the controller is proven in the following theorem, Theorem 1.

Theorem 8.1. *Consider the controller* $\mathbf{y}_c = \boldsymbol{\mathcal{G}}_c \mathbf{u}_c$ *described by* (8.10)-(8.11)*, where* $N \in \mathbb{N}_{>0}$*, with* $\bar{\epsilon}, \bar{\delta} \in \mathbb{R}_{>0}$ *and with the scheduling signals satisfying* (8.8)*. Furthermore, if all* $\mathbf{A}_{c,i}, i = 1, \ldots, N$ *are Hurwitz, if* $\mathbf{P}_c$ *is obtained from* (8.13)*, and if* $(\mathbf{B}_{c,i}, \mathbf{C}_{c,i})$ *satisfies* (8.15)*, then the controller* $\mathbf{y}_c = \boldsymbol{\mathcal{G}}_c \mathbf{u}_c$ *is VSP.*

Proof. Consider the Lyapunov-type function

$$\mathcal{V} = \frac{1}{2} \mathbf{x}_c^{\mathsf{T}} \mathbf{P}_c \mathbf{x}_c. \tag{8.16}$$

The time derivative of (8.16) is

$$\begin{aligned}
\dot{\mathcal{V}} &= \frac{1}{2}\mathbf{x}_\mathrm{c}^\mathsf{T}\left(\mathbf{P}_\mathrm{c}\mathbf{A}_\mathrm{c} + \mathbf{A}_\mathrm{c}^\mathsf{T}\mathbf{P}_\mathrm{c}\right)\mathbf{x}_\mathrm{c} + \mathbf{x}_\mathrm{c}^\mathsf{T}\mathbf{P}_\mathrm{c}\mathbf{B}_\mathrm{c}\mathbf{u}_\mathrm{c} - \bar{\epsilon}\mathbf{x}_\mathrm{c}^\mathsf{T}\mathbf{P}_\mathrm{c}\mathbf{B}_\mathrm{c}\mathbf{C}_\mathrm{c}\mathbf{x}_\mathrm{c} \\
&= -\frac{1}{2}\mathbf{x}_\mathrm{c}^\mathsf{T}\mathbf{Q}_\mathrm{c}\mathbf{x}_\mathrm{c} + \mathbf{x}_\mathrm{c}^\mathsf{T}\mathbf{C}_\mathrm{c}^\mathsf{T}\mathbf{u}_\mathrm{c} - \bar{\epsilon}\mathbf{x}_\mathrm{c}^\mathsf{T}\mathbf{C}_\mathrm{c}^\mathsf{T}\mathbf{C}_\mathrm{c}\mathbf{x}_\mathrm{c} \\
&= -\frac{1}{2}\mathbf{x}_\mathrm{c}^\mathsf{T}\mathbf{Q}_\mathrm{c}\mathbf{x}_\mathrm{c} + \left(\mathbf{y}_\mathrm{c} - \mathbf{D}_\mathrm{c}\mathbf{u}_\mathrm{c}\right)^\mathsf{T}\mathbf{u}_\mathrm{c} \\
&\quad - \bar{\epsilon}\left(\mathbf{y}_\mathrm{c} - \mathbf{D}_\mathrm{c}\mathbf{u}_\mathrm{c}\right)^\mathsf{T}\left(\mathbf{y}_\mathrm{c} - \mathbf{D}_\mathrm{c}\mathbf{u}_\mathrm{c}\right) \\
&= -\frac{1}{2}\mathbf{x}_\mathrm{c}^\mathsf{T}\mathbf{Q}_\mathrm{c}\mathbf{x}_\mathrm{c} + \mathbf{y}_\mathrm{c}^\mathsf{T}\mathbf{u}_\mathrm{c} - \mathbf{u}_\mathrm{c}^\mathsf{T}\mathbf{D}_\mathrm{c}\mathbf{u}_\mathrm{c} - \bar{\epsilon}\mathbf{u}_\mathrm{c}^\mathsf{T}\mathbf{D}_\mathrm{c}^\mathsf{T}\mathbf{D}_\mathrm{c}\mathbf{u}_\mathrm{c} \\
&\quad - \bar{\epsilon}\mathbf{y}_\mathrm{c}^\mathsf{T}\mathbf{y}_\mathrm{c} + 2\bar{\epsilon}\mathbf{y}_\mathrm{c}^\mathsf{T}\mathbf{D}_\mathrm{c}\mathbf{u}_\mathrm{c}.
\end{aligned} \tag{8.17}$$

Let $q = \alpha \min_{i=1,\ldots,N}(\lambda_{\min}(\mathbf{Q}_{\mathrm{c},i}))$. Integrating (8.17) in time, and simplifying, yields

$$\begin{aligned}
\mathcal{V}(T) - \mathcal{V}(0) &= -\frac{1}{2}\int_0^T \mathbf{x}_\mathrm{c}^\mathsf{T}\mathbf{Q}_\mathrm{c}\mathbf{x}_\mathrm{c}\mathrm{d}t - \langle \mathbf{u}_\mathrm{c}, \mathbf{D}_\mathrm{c}\mathbf{u}_\mathrm{c}\rangle_T \\
&\quad - \bar{\epsilon}\left\|\mathbf{D}_\mathrm{c}\mathbf{u}_\mathrm{c}\right\|_{2T}^2 - \bar{\epsilon}\left\|\mathbf{y}_\mathrm{c}\right\|_{2T}^2 + \langle \mathbf{y}_\mathrm{c}, \mathbf{u}_\mathrm{c}\rangle_T + 2\bar{\epsilon}\left\langle \mathbf{y}_\mathrm{c}, \mathbf{D}_\mathrm{c}\mathbf{u}_\mathrm{c}\right\rangle_T, \\
(1 + 2\bar{\epsilon}\bar{\delta})\left\langle \mathbf{y}_\mathrm{c}, \mathbf{u}_\mathrm{c}\right\rangle_T &\geq -\mathcal{V}(0) + \frac{1}{2}q\left\|\mathbf{x}_\mathrm{c}\right\|_{2T}^2 + \bar{\epsilon}\left\|\mathbf{y}_\mathrm{c}\right\|_{2T}^2 + (\bar{\delta} + \bar{\epsilon}\bar{\delta}^2)\left\|\mathbf{u}_\mathrm{c}\right\|_{2T}^2, \\
\left\langle \mathbf{y}_\mathrm{c}, \mathbf{u}_\mathrm{c}\right\rangle_T &\geq \frac{1}{1 + 2\bar{\epsilon}\bar{\delta}}\Big(-\mathcal{V}(0) + \frac{1}{2}q\left\|\mathbf{x}_\mathrm{c}\right\|_{2T}^2 + \bar{\epsilon}\left\|\mathbf{y}_\mathrm{c}\right\|_{2T}^2 + \bar{\delta}(1 + \bar{\epsilon}\bar{\delta})\left\|\mathbf{u}_\mathrm{c}\right\|_{2T}^2 \Big) \\
&\geq \beta + \epsilon\left\|\mathbf{y}_\mathrm{c}\right\|_{2T}^2 + \delta\left\|\mathbf{u}_\mathrm{c}\right\|_{2T}^2,
\end{aligned} \tag{8.18}$$

where

$$\epsilon = \frac{\bar{\epsilon}}{1 + 2\bar{\epsilon}\bar{\delta}}, \quad \delta = \frac{\bar{\delta}(1 + \bar{\epsilon}\bar{\delta})}{1 + 2\bar{\epsilon}\bar{\delta}}. \tag{8.19}$$

Noting that $\beta = -\frac{1}{1+2\bar{\epsilon}\bar{\delta}}\mathcal{V}(0)$ is only dependent on the initial condition, and $\epsilon, \delta \in \mathbb{R}_{>0}$, proves $\mathbf{y}_\mathrm{c} = \boldsymbol{\mathcal{G}}_\mathrm{c}\mathbf{u}_\mathrm{c}$ is VSP. □

Corollary 8.2. *Consider the controller* $\mathbf{y}_c = \boldsymbol{\mathcal{G}}_c\mathbf{u}_c$ *in Theorem 8.1. The controller* $\mathbf{y}_c = \boldsymbol{\mathcal{G}}_c\mathbf{u}_c$ *is*

1. *ISP if* $\bar{\delta} \in \mathbb{R}_{>0}$, $\bar{\epsilon} = 0$,
2. *OSP if* $\bar{\delta} = 0$, $\bar{\epsilon} \in \mathbb{R}_{>0}$,
3. *passive if* $\bar{\delta} = 0$, $\bar{\epsilon} = 0$.

Proof. The proof follows by substituting the values for $\bar{\delta}$ and $\bar{\epsilon}$ into (8.19), and then by comparing (8.18) to the definition of interconnected passive systems from [88]. □

8.3 Controller Synthesis

Provided unknown uncertainties do not violate the plant's passive map, any controller that satisfies Theorem 8.1 can robustly stabilize (8.1)-(8.3). The next challenge is to synthesize a controller that meets some closed-loop performance measure. A linear system's closed-loop performance can be measured using $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norms, and an optimal controller would be one that minimizes said norms. Ideally, the VSP-LPV controller could be designed to minimize the induced $\mathcal{L}_2$ norm of the closed-loop nonlinear system. Even minimizing the closed-loop $\mathcal{H}_2$ or $\mathcal{H}_\infty$ norms of an LTI plant in negative feedback with an LTI controller subject to (8.13) and (8.15) with $N = 1$ is difficult. Methods only exist that minimize an upper-bound on the $\mathcal{H}_2$ or $\mathcal{H}_\infty$ norm [105, 106, 107, 108]. In [109, 110], an LTI controller is derived by minimizing an upper bound of $\mathcal{H}_2$ and $\mathcal{H}_\infty$ for systems with polytopic uncertainty. An upper bound on $\mathcal{H}_2$ and induced $\mathcal{L}_2$ norm of affine parameter systems with affine parameter controllers is found in [111].

To overcome these challenges, given either the $\mathcal{H}_2$ or the $\mathcal{H}_\infty$ framework, we consider two general methods and their dual to design a controller that satisfies the conditions of Theorem 8.1. The method consists of transforming the passive nonlinear system to an LPV system with an affine representation, as shown in Fig. 8.1. Next, either the $\mathcal{H}_2$ or the $\mathcal{H}_\infty$ closed-loop norm at each set point is minimized. The resulting controller is forced to satisfy (8.13) and (8.15) using an $\mathcal{H}_2$-inspired or SPR-inspired method. The controller synthesis methods inspired from $\mathcal{H}_2$ or $\mathcal{H}_\infty$ control techniques are here named $\mathcal{H}_2$-VSP and $\mathcal{H}_\infty$-VSP methods. These synthesis methods are influenced by previous work on passivity-based gain-scheduled controller [96, 95], and on conic controllers [19, 18].

Neglecting a $\mathbf{D}_\mathrm{c}$ matrix during the initial parts of the synthesis is required for the $\mathcal{H}_2$ synthesis so that there is roll-off at higher frequency. In addition, the $\mathbf{D}_\mathrm{c}$ matrix is omitted for the $\mathcal{H}_\infty$ controller synthesis so that a large $\mathbf{D}_\mathrm{c}$ is not provided by the synthesis method, which would excite high-frequency measurement noise. That being said, a zero $\mathbf{D}_\mathrm{c}$ would violate the VSP condition of the controller, and thus $\mathbf{D}_\mathrm{c}$ is set to $\mathbf{D}_\mathrm{c} = \bar{\delta}\mathbf{1}$, where $\bar{\delta}$ is small. We choose $\bar{\delta} = 10^{-5}$ since this has an insignificant impact on the controller performance, but has a critical impact on its passivity properties.

8.3.1 $\mathcal{H}_2$-VSP Controller

In this section the plant considered is described by (8.4)-(8.6) assuming

1. the scheduling signals satisfy (8.8),

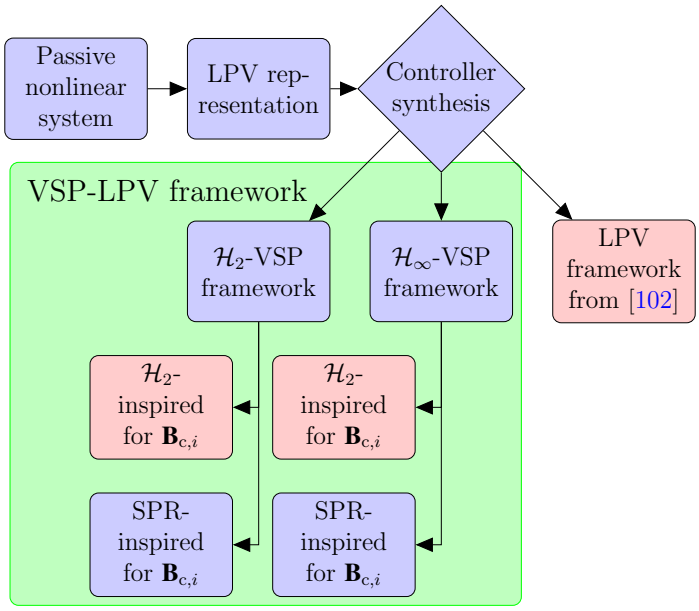


Figure 8.1: Workflow for LPV controller synthesis. Specific controllers used for experimentation are in red. The "$\mathcal{H}_2$-inspired for $\mathbf{B}_{c,i}$" and "SPR-inspired for $\mathbf{B}_{c,i}$" methods are also applicable to $\mathbf{C}_{c,i}$, but are not shown.

2. $(\mathbf{A}_{c,i}, \mathbf{B}_{1,i})$ is controllable and $(\mathbf{A}_{c,i}, \mathbf{C}_{1,i})$ is observable,
3. $(\mathbf{A}_{c,i}, \mathbf{B}_{2,i})$ is controllable and $(\mathbf{A}_{c,i}, \mathbf{C}_{2,i})$ is observable,
4. $\mathbf{D}_{12,i}^\mathsf{T}\mathbf{C}_{1,i} = \mathbf{0}$ and $\mathbf{D}_{12,i}^\mathsf{T}\mathbf{D}_{12,i} > 0$,
5. $\mathbf{D}_{21,i}\mathbf{B}_{1,i}^\mathsf{T} = \mathbf{0}$ and $\mathbf{D}_{21,}\mathbf{D}_{21,i}^\mathsf{T} > 0$, and
6. $\mathbf{D}_{11,i} = \mathbf{D}_{22,i} = \mathbf{0}$.

The $\mathcal{H}_2$-VSP controller $\boldsymbol{\mathcal{G}}_c$ is designed at each set point, to obtain $\boldsymbol{\mathcal{G}}_{c,i}$ with state-space matrices $(\mathbf{A}_{c,i}, \mathbf{B}_{c,i}, \mathbf{C}_{c,i}, \mathbf{D}_c)$. Let $\mathbf{Q}_{K,i} = \mathbf{C}_{1,i}^\mathsf{T}\mathbf{C}_{1,i}$, $\mathbf{R}_{K,i} = \mathbf{D}_{12,i}^\mathsf{T}\mathbf{D}_{12,i}$, $\mathbf{Q}_{L,i} = \mathbf{B}_{1,i}\mathbf{B}_{1,i}^\mathsf{T}$, and $\mathbf{R}_{L,i} = \mathbf{D}_{21,i}\mathbf{D}_{21,i}^\mathsf{T}$, and then minimize the closed-loop $\mathcal{H}_2$ norm. This can be done as in [112], or as a convex optimization problem in two steps [31, p. 115], by first, for $i = 1, \ldots, N$, maximizing $\mathcal{J}_K(\mathbf{P}_i) = \mathrm{tr}(\mathbf{B}_{1,i}^\mathsf{T}\mathbf{P}_i\mathbf{B}_{1,i})$ subject to the constraint

$$\begin{bmatrix} \mathbf{A}_i^\mathsf{T}\mathbf{P}_i + \mathbf{P}_i\mathbf{A}_i + \mathbf{Q}_{K,i} & \mathbf{P}_i\mathbf{B}_{2,i} \\ \star & \mathbf{R}_{K,i} \end{bmatrix} \geq 0, \tag{8.20}$$

and then by maximizing $\mathcal{J}_L(\boldsymbol{\Pi}_i) = \mathrm{tr}(\mathbf{C}_{1,i}\boldsymbol{\Pi}_i\mathbf{C}_{1,i}^\mathsf{T})$ subject to

$$\begin{bmatrix} \mathbf{A}_i\boldsymbol{\Pi}_i + \boldsymbol{\Pi}_i\mathbf{A}_i^\mathsf{T} + \mathbf{Q}_{L,i} & \mathbf{C}_{2,i}^\mathsf{T}\boldsymbol{\Pi}_i \\ \star & \mathbf{R}_{L,i} \end{bmatrix} \geq 0. \tag{8.21}$$

Given $\mathbf{P}_i$, set $\mathbf{K}_i = \mathbf{R}_{K,i}^{-1}\mathbf{B}_{2,i}^{\mathsf{T}}\mathbf{P}_i$ and given $\mathbf{\Pi}_i$, set $\mathbf{L}_i = \mathbf{\Pi}_i\mathbf{C}_{2,i}^{\mathsf{T}}\mathbf{R}_{L,i}^{-1}$, leading to a controller of the form

$$\dot{\mathbf{x}}_{\mathrm{c}} = \underbrace{(\mathbf{A}_i - \mathbf{B}_{2,i}\mathbf{K}_i - \mathbf{L}_i\mathbf{C}_{2,i})}_{\mathbf{A}_{\mathrm{c},i}}\mathbf{x}_{\mathrm{c}} + \underbrace{\mathbf{L}_i}_{\mathbf{B}_{\mathrm{c},i}}\mathbf{u}_{\mathrm{c}}, \tag{8.22}$$

$$\mathbf{y}_{\mathrm{c}} = \underbrace{\mathbf{K}_i}_{\mathbf{C}_{\mathrm{c},i}}\mathbf{x}_{\mathrm{c}}. \tag{8.23}$$

The matrix $\mathbf{A}_{\mathrm{c},i}$ must be Hurwitz, and thus it must be verified that $\mathbf{A}_i - \mathbf{B}_{2,i}\mathbf{K}_i - \mathbf{L}_i\mathbf{C}_{2,i}$ does satisfy this property. However, simply restricting $\mathbf{A}_{\mathrm{c},i}$ to be $\mathbf{A}_{\mathrm{c},i} = \mathbf{A}_i - \mathbf{B}_{2,i}\mathbf{K}_i$ or $\mathbf{A}_{\mathrm{c},i} = \mathbf{A}_i - \mathbf{L}_i\mathbf{C}_{2,i}$ can suffice. The controller satisfying Theorem 8.1 at this set point has a similar form with

$$\mathbf{A}_{\mathrm{c},i} = \mathbf{A}_i - \mathbf{B}_{2,i}\mathbf{K}_i - \mathbf{L}_i\mathbf{C}_{2,i}, \quad \mathbf{C}_{\mathrm{c},i} = \mathbf{K}_i, \quad \mathbf{D}_{\mathrm{c}} = \bar{\delta}\mathbf{1},$$

where $\mathbf{B}_{\mathrm{c},i}$ is designed so that (8.13) and (8.15) are satisfied, and $\bar{\delta} \in \mathbb{R}_{>0}$. Various design methods for $\mathbf{B}_{\mathrm{c},i}$ exist, explained in Sections 8.3.1.1 and 8.3.1.2. The dual method to ensure Theorem 8.1 is satisfied is to let $\mathbf{B}_{\mathrm{c},i} = \mathbf{L}_i$ and use $\mathbf{C}_{\mathrm{c},i}$ as the design variable. The matrix $\mathbf{C}_{\mathrm{c},i}$ is determined so that (8.13) and (8.15) hold.

8.3.1.1 $\mathcal{H}_2$-inspired synthesis

Modifying the procedure in [108], the $\mathbf{B}_{\mathrm{c},i}$ matrix is determined by minimizing the $\mathcal{H}_2$-norm of the difference between the $\mathcal{H}_2$ controller and a controller that satisfies the conditions of (8.13) and (8.15). This optimization problem is given by minimizing

$$\mathcal{J}(\mathbf{P}_{\mathrm{c}}^{-1}, \mathbf{B}_{\mathrm{c},1}, \ldots, \mathbf{B}_{\mathrm{c},N}) = \sum_{i=1}^{N} \mathrm{tr}(\mathbf{B}_{\mathrm{c},i} - \mathbf{L}_i)^{\mathsf{T}}\mathbf{W}_i(\mathbf{B}_{\mathrm{c},i} - \mathbf{L}_i), \tag{8.24}$$

subject to, for $i = 1, \ldots, N$,

$$\mathbf{A}_{\mathrm{c},i}\mathbf{P}_{\mathrm{c}}^{-1} + \mathbf{P}_{\mathrm{c}}^{-1}\mathbf{A}_{\mathrm{c},i}^{\mathsf{T}} < 0, \tag{8.25}$$

$$\mathbf{B}_{\mathrm{c},i} = \mathbf{P}_{\mathrm{c}}^{-1}\mathbf{C}_{\mathrm{c},i}^{\mathsf{T}}, \tag{8.26}$$

where (8.25)-(8.26) are a modification of (8.13) and (8.15). The matrix $\mathbf{W}_i$ is the observability Grammian that satisfies

$$\mathbf{A}_{\mathrm{c},i}^{\mathsf{T}}\mathbf{W}_i + \mathbf{W}_i\mathbf{A}_{\mathrm{c},i} + \mathbf{C}_{\mathrm{c},i}^{\mathsf{T}}\mathbf{C}_{\mathrm{c},i} = \mathbf{0}. \tag{8.27}$$

The objective function (8.24) is chosen because when the difference between $\mathbf{B}_{c,i}$ and $\mathbf{L}_i$ subject to the weight $\mathbf{W}_i$ is minimized, the difference between the $\mathcal{H}_2$ controller and the controller that satisfies the conditions of (8.13) and (8.15) is minimized as desired. This method to determine $\mathbf{B}_{c,i}$ is used in this chapter.

The solution to the optimization problem given by (8.24)-(8.27) can be transformed to an SDP by introducing a slack variable $\mathbf{Z}_i$, where

$$\mathbf{Z}_i \geq (\mathbf{B}_{c,i} - \mathbf{L}_i)^\mathsf{T} \mathbf{W}_i (\mathbf{B}_{c,i} - \mathbf{L}_i). \tag{8.28}$$

SDPs have a linear objective function with LMI constraints, and can be solved numerically very efficiently [31]. The Schur complement and (8.26) can then be used to expand (8.28). The optimization problem's objective becomes to minimize

$$\mathcal{J}(\mathbf{P}_c^{-1}, \mathbf{Z}_1, \ldots, \mathbf{Z}_N) = \sum_{i=1}^{N} \mathrm{tr}(\mathbf{Z}_i),$$

such that for $i = 1, \ldots, N$,

$$\begin{bmatrix} \mathbf{Z}_i & (\mathbf{P}_c^{-1}\mathbf{C}_{c,i} - \mathbf{L}_i)^\mathsf{T} \\ \star & \mathbf{W}_i^{-1} \end{bmatrix} \geq 0,$$

and (8.25) are satisfied, where $\mathbf{W}_i$ is given by (8.27). The matrix $\mathbf{B}_{c,i}$ is then determined from (8.26) using $\mathbf{P}_c^{-1}$ found from the solution of the optimization problem.

Similar to [19], the dual version of this synthesis method is to let $\mathbf{B}_{c,i} = \mathbf{L}_i$ and to allow $\mathbf{C}_{c,i}$ be the design variable. In this case, (8.24) is modified so that $\mathcal{J}(\mathbf{C}_{c,1}, \ldots, \mathbf{C}_{c,N}) = \sum_{i=1}^{N} \mathrm{tr}(\mathbf{C}_{c,i} - \mathbf{K}_i)^\mathsf{T} \mathbf{V}_i (\mathbf{C}_{c,i} - \mathbf{K}_i)$, subject to (8.13) and (8.15), where $\mathbf{V}_i$ is the controllability Grammian.

8.3.1.2 SPR-inspired synthesis

Recall via the Kalman-Yakubovich-Popov (KYP) Lemma that an LTI system with state-space realization $(\mathbf{A}, \mathbf{B}, \mathbf{C})$ is SPR if and only if $\mathbf{A}^\mathsf{T}\mathbf{P} + \mathbf{A}\mathbf{P} = -\mathbf{Q}$, and $\mathbf{P}\mathbf{B} = \mathbf{C}^\mathsf{T}$ for positive definite matrices $\mathbf{P}$ and $\mathbf{Q}$ [89]. An LTI controller with state-space realization $(\mathbf{A}_c, \mathbf{B}_c, \mathbf{C}_c)$ with $\mathbf{A}_c$ Hurwitz is made SPR by choosing a positive definite $\mathbf{Q}_c$, by solving $\mathbf{A}_c^\mathsf{T}\mathbf{P}_c + \mathbf{A}_c\mathbf{P}_c = -\mathbf{Q}_c$ for $\mathbf{P}_c$, and then solving for a new $\mathbf{B}_c$ using $\mathbf{B}_c = \mathbf{P}_c^{-1}\mathbf{C}_c^\mathsf{T}$ [113].

Inspired by this method, the second method to determine each $\mathbf{B}_{c,i}$ begins by

choosing $\mathbf{Q}_{c,i} = \mathbf{Q}_{c,i}^{\mathsf{T}} > 0$ and solving for $\mathbf{P}_{c,i}$ in

$$\mathbf{P}_{c,i}\mathbf{A}_{c,i} + \mathbf{A}_{c,i}^{\mathsf{T}}\mathbf{P}_{c,i} = -\mathbf{Q}_{c,i}.$$

However, for the LPV controller to be VSP, there cannot be different $\mathbf{P}_{c,i}$ matrices, but rather a single $\mathbf{P}_c$. The matrix $\mathbf{P}_c$ is determined by minimizing the difference between $\mathbf{P}_c$ and each $\mathbf{P}_{c,i}$ using the objective function

$$\mathcal{J}(\mathbf{P}_c) = \sum_{i=1}^{N} \mathrm{tr}(\mathbf{P}_{c,i} - \mathbf{P}_c)^{\mathsf{T}}(\mathbf{P}_{c,i} - \mathbf{P}_c), \tag{8.29}$$

subject to the constraints (8.13). The matrix $\mathbf{B}_{c,i}$ is then found by solving (8.15). Although not shown in this chapter, this problem can be also formulated as an SDP using a procedure similar to the one outlined in Section 8.3.1.1. A dual version of this method exists by setting $\mathbf{B}_{c,i} = \mathbf{L}_i$ and then using the resulting $\mathbf{P}_c$ from the solution to (8.29) in (8.15) to determine $\mathbf{C}_{c,i}$.

8.3.2 $\mathcal{H}_\infty$-VSP Controller

The assumptions on (8.4)-(8.6) are [114]

1. the scheduling signals satisfy (8.8),
2. $(\mathbf{A}_{c,i}, \mathbf{B}_{2,i})$ is stabilizable and $(\mathbf{A}_{c,i}, \mathbf{C}_{2,i})$ is detectable, and
3. $\mathbf{D}_{22} = \mathbf{0}$ for simplicity.

These assumptions are less restrictive than in the $\mathcal{H}_2$-VSP case. In this framework, a controller in the form of (8.10)-(8.11) is needed with matrices

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{A}_{c,i} & \mathbf{B}_{c,i} \\ \mathbf{C}_{c,i} & \mathbf{D}_{c,i} \end{bmatrix}, \tag{8.30}$$

that minimizes the closed-loop $\mathcal{H}_\infty$-norm at each set point. When finding this nominal controller, $\mathbf{D}_{c,i}$ is set to zero, and then later adjusted to have a constant $\mathbf{D}_c = \bar{\delta}\mathbf{1}$ as in the $\mathcal{H}_2$ case. Notation borrows from [27], but the method to determine (8.30) is from [114]. Define the matrices $\mathbf{N}_{o,i}$ and $\mathbf{N}_{c,i}$ as

$$\mathcal{R}(\mathbf{N}_{o,i}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{C}_{2,i} & \mathbf{D}_{21,i} \end{bmatrix}\right), \tag{8.31}$$

$$\mathcal{R}(\mathbf{N}_{c,i}) = \mathcal{N}\left(\begin{bmatrix} \mathbf{B}_{2,i}^{\mathsf{T}} & \mathbf{D}_{12,i}^{\mathsf{T}} \end{bmatrix}\right). \tag{8.32}$$

The matrices $\mathbf{X}_i = \mathbf{X}_i^{\mathsf{T}} > 0$ and $\mathbf{Y}_i = \mathbf{Y}_i^{\mathsf{T}} > 0$ are found by minimizing $\gamma_i > 0$ subject to the LMI constraints

$$\begin{bmatrix} \mathbf{N}_{\mathrm{o},i} & \mathbf{0} \\ \star & \mathbf{1} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mathbf{A}_i^{\mathsf{T}}\mathbf{X}_i + \mathbf{X}_i\mathbf{A}_i & \mathbf{X}_i\mathbf{B}_{1,i} & \mathbf{C}_{1,i}^{\mathsf{T}} \\ \star & -\gamma_i\mathbf{1} & \mathbf{D}_{11,i}^{\mathsf{T}} \\ \star & \star & -\gamma_i\mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{N}_{\mathrm{o},i} & \mathbf{0} \\ \star & \mathbf{1} \end{bmatrix} < 0, \tag{8.33}$$

$$\begin{bmatrix} \mathbf{N}_{\mathrm{c},i} & \mathbf{0} \\ \star & \mathbf{1} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mathbf{A}_i\mathbf{Y}_i + \mathbf{Y}_i\mathbf{A}_i^{\mathsf{T}} & \mathbf{Y}_i\mathbf{C}_{1,i}^{\mathsf{T}} & \mathbf{B}_{1,i} \\ \star & -\gamma_i\mathbf{1} & \mathbf{D}_{11,i} \\ \star & \star & -\gamma_i\mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{N}_{\mathrm{c},i} & \mathbf{0} \\ \star & \mathbf{1} \end{bmatrix} < 0, \tag{8.34}$$

$$\begin{bmatrix} \mathbf{X}_i & \mathbf{1} \\ \star & \mathbf{Y}_i \end{bmatrix} \geq 0. \tag{8.35}$$

The matrix $\bar{\mathbf{X}}_i$ is determined from $\mathbf{X}_i - \mathbf{Y}_i^{-1} = \bar{\mathbf{X}}_i\bar{\mathbf{X}}_i^{\mathsf{T}}$. The matrix $\bar{\mathbf{X}}_i$ is used to construct $\mathbf{X}_{\mathrm{cl},i}$, where

$$\mathbf{X}_{\mathrm{cl},i} = \begin{bmatrix} \mathbf{X}_i & \bar{\mathbf{X}}_i^{\mathsf{T}} \\ \star & \mathbf{1} \end{bmatrix}. \tag{8.36}$$

The matrix $\mathbf{J}_i$ is found by solving

$$\mathbf{H}_{X_{\mathrm{cl},i}} + \mathbf{Q}_i^{\mathsf{T}}\mathbf{J}_i^{\mathsf{T}}\mathbf{P}_{X_{\mathrm{cl},i}} + \mathbf{P}_{X_{\mathrm{cl},i}}^{\mathsf{T}}\mathbf{J}_i\mathbf{Q}_i < 0, \tag{8.37}$$

using the parameterizations

$$\overline{\mathbf{A}}_i = \begin{bmatrix} \mathbf{A}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \qquad \underline{\mathbf{B}}_i = \begin{bmatrix} \mathbf{0} & \mathbf{B}_{2,i} \\ \mathbf{1} & \mathbf{0} \end{bmatrix}, \tag{8.38}$$

$$\underline{\mathbf{C}}_i = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{C}_{2,i} & \mathbf{0} \end{bmatrix}, \qquad \overline{\mathbf{D}}_{1,i} = \begin{bmatrix} \mathbf{D}_{1,i} \\ \mathbf{0} \end{bmatrix}, \tag{8.39}$$

$$\overline{\mathbf{C}}_i = \begin{bmatrix} \mathbf{C}_{1,i} & \mathbf{0} \end{bmatrix}, \qquad \underline{\mathbf{D}}_{12,i} = \begin{bmatrix} \mathbf{0} & \mathbf{D}_{12,i} \end{bmatrix}, \tag{8.40}$$

$$\underline{\mathbf{D}}_{21,i} = \begin{bmatrix} \mathbf{0} \\ \mathbf{D}_{21,i} \end{bmatrix}, \tag{8.41}$$

and where

$$\mathbf{P}_{X_{\mathrm{cl},i}} = \begin{bmatrix} \underline{\mathbf{B}}_i^{\mathsf{T}} \mathbf{X}_{\mathrm{cl},i} & \mathbf{0} & \underline{\mathbf{D}}_{12,i}^{\mathsf{T}} \end{bmatrix}, \tag{8.42}$$

$$\mathbf{Q}_i = \begin{bmatrix} \underline{\mathbf{C}}_i & \underline{\mathbf{D}}_{21,i} & \mathbf{0} \end{bmatrix}, \tag{8.43}$$

$$\mathbf{H}_{X_{\mathrm{cl},i}} = \begin{bmatrix} \overline{\mathbf{A}}_i^{\mathsf{T}} \mathbf{X}_{\mathrm{cl},i} + \mathbf{X}_{\mathrm{cl},i} \overline{\mathbf{A}}_i & \mathbf{X}_{\mathrm{cl},i} \overline{\mathbf{D}}_{1,i} & \overline{\mathbf{C}}_i^{\mathsf{T}} \\ \star & -\gamma_i \mathbf{1} & \mathbf{D}_{11,i}^{\mathsf{T}} \\ \star & \star & -\gamma_i \mathbf{1} \end{bmatrix}. \tag{8.44}$$

This synthesis procedure yields an $\mathcal{H}_\infty$ controller for a specific set point. As in the $\mathcal{H}_2$ case, the nominal $\mathbf{B}_{\mathrm{c},i}$ can ignored and replaced by a $\mathbf{B}_{\mathrm{c},i}$ matrix that ensures (8.13) and (8.15) are satisfied. In this chapter, the method in Section 8.3.1.1 is used to determine $\mathbf{B}_{\mathrm{c},i}$. The resulting $\mathbf{J}_i$ is not guaranteed to satisfy (8.37) for the minimal γ_i, as expected due to the added robustness caused by the VSP nature of the controller.

8.3.3 Self-Scheduled LPV Controller

The LPV controller used is from [102] as this controller has a very similar structure to the $\mathcal{H}_\infty$-VSP controller. The assumptions are

1. the scheduling signals satisfy (8.8),
2. the plant is polytopic,
3. $(\mathbf{A}, \mathbf{B}_2)$ and $(\mathbf{A}, \mathbf{C}_2)$ are quadratically stabilizable and detectable (see [102]),
4. $\mathbf{D}_{22} = \mathbf{0}$ for simplicity, and
5. $\mathbf{B}_2$, $\mathbf{C}_2$, $\mathbf{D}_{12}$, and $\mathbf{D}_{21}$ must all be constant and parameter independent.

In the application example, the LPV representation is already polytopic. The polytopic LPV control synthesis essentially determines control matrices at the vertices of the polytope of the plant with a common Lyapunov function, and are then scheduled as in (8.12).

The fifth assumption is unlike the VSP-LPV controller, where the matrices $\mathbf{B}_2$, $\mathbf{C}_2$, $\mathbf{D}_{12}$, $\mathbf{D}_{21}$ can be different at each set point. In the application example, $\mathbf{B}_2$ is dependent on the mass matrix of the system, and thus $\mathbf{B}_2$ varies as a function of the scheduling signal. If $\mathbf{B}_2$ is not constant, there are an infinite number of constraints in the resulting LMIs required for controller synthesis [115], which can be overcome by pre-filtering the control inputs as in [102]. A first-order low-pass filter is used for both inputs $G(s) = \frac{\omega_i}{s+\omega_i}$ that has large enough bandwidth to capture plant dynamics. The filter increases the effective order of the controller.

8.3.4 VSP-LPV Controller Synthesis Summary

Each controller synthesis method begins with linearized plant in (8.9). The plant must satisfy the assumptions for the $\mathcal{H}_2$-VSP controller in Section 8.3.1.

1. Define $\mathbf{Q}_{K,i} = \mathbf{C}_{1,i}^{\mathsf{T}}\mathbf{C}_{1,i}$, $\mathbf{R}_{K,i} = \mathbf{D}_{12,i}^{\mathsf{T}}\mathbf{D}_{12,i}$, $\mathbf{Q}_{L,i} = \mathbf{B}_{1,i}\mathbf{B}_{1,i}^{\mathsf{T}}$ and $\mathbf{R}_{L,i} = \mathbf{D}_{21,i}\mathbf{D}_{21,i}^{\mathsf{T}}$.
2. Maximize $\mathcal{J}_K(\mathbf{P}_i) = \mathrm{tr}(\mathbf{B}_{1,i}^{\mathsf{T}}\mathbf{P}_i\mathbf{B}_{1,i})$ subject to (8.20).
3. Maximize $\mathcal{J}_L(\mathbf{\Pi}_i) = \mathrm{tr}(\mathbf{C}_{1,i}\mathbf{\Pi}_i\mathbf{C}_{1,i}^{\mathsf{T}})$ subject to (8.21).
4. Set $\mathbf{C}_{\mathrm{c},i} = \mathbf{R}_{K,i}^{-1}\mathbf{B}_{2,i}^{\mathsf{T}}\mathbf{P}_i$, $\mathbf{L}_i = \mathbf{\Pi}_i\mathbf{C}_{2,i}\mathbf{R}_{L,i}^{-1}$and $\mathbf{A}_{\mathrm{c},i} = \mathbf{A}_i - \mathbf{B}_2\mathbf{C}_{\mathrm{c},i} - \mathbf{L}_i\mathbf{C}_{2,i}$. Verify that $\mathbf{A}_{\mathrm{c},i}$ is Hurwitz.
5. Determine a $\mathbf{B}_{\mathrm{c},i}$ by following the procedure in Section 8.3.1.1 or 8.3.1.2.
6. Set $\mathbf{D}_{\mathrm{c}} = \bar{\delta}\mathbf{1}$.

The $\mathcal{H}_\infty$-VSP controller requires $(\mathbf{A}_{\mathrm{c},i}, \mathbf{B}_{2,i}, \mathbf{C}_{2,i})$ is stabilizable and detectable with $\mathbf{D}_{22} = \mathbf{0}$ for simplicity.

1. Minimize $\gamma_i > 0$ subject to (8.33)-(8.35) and find an $\mathbf{X}_i$ and $\mathbf{Y}_i$. This step involves N minimizations with 3 LMI constraints each.
2. Solve for $\mathbf{X}_{\mathrm{cl},i}$ using (8.36) N times.
3. Determine $\mathbf{J}_i$ using (8.37).
4. Determine a $\mathbf{B}_{\mathrm{c},i}$ by following the procedure in Section 8.3.1.1 or 8.3.1.2.
5. Set $\mathbf{D}_{\mathrm{c}} = \bar{\delta}\mathbf{1}$.

8.4 Application Example

The content of this work is directed at controlling a general nonlinear passive system. The purpose of this application example is to show how to use the VSP-LPV controller on a real system. Although the $\mathcal{H}_\infty$-VSP controller has relatively superior performance, the main thrust is that regardless of how other controllers perform, the VSP-LPV controllers guarantee closed-loop input-output stability as it exploits passivity properties of the plant under consideration, while other existing LPV control design methods do not.

8.4.1 Plant Description

The three controller types are used on the Quanser two-link planar flexible-joint manipulator [116], shown in Fig. 8.2. The manipulator's physical properties are in Table 8.1. Control of the flexible joint manipulator is difficult because the system

Table 8.1: Two-link manipulator properties.

Parameter	Joint 1		Joint 2	
	Symbol	Value	Symbol	Value
Length (m)	L_1	0.343	L_2	0.267
Link mass (kg)	m_1	1.510	m_2	0.873
Link moment of inertia (kg·m^2)	I_1	0.0392	I_2	0.00808
Centre of mass of link relative to joint (m)	$r_{\text{cm},1}$	0.159	$r_{\text{cm},2}$	0.0550
Joint stiffness (N·m/rad)	k_1	9.0	k_2	4.0
Hub moment of inertia (kg·m^2)	$I_{\text{hub},1}$	0.011	$I_{\text{hub},2}$	0.0094

consists of a rigid body and lightly damped flexible joints. The flexibility of the joints is a source of modelling uncertainty, and the varying dynamics of the manipulator are a source of nonlinearity. The VSP-LPV controller is robust to uncertain mass and stiffness of the manipulator, and the gain-scheduling provides a controller that is able to effectively function throughout various manipulator configurations.

The manipulator's equations of motion are [96]

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \hat{\mathbf{B}}\boldsymbol{\tau} + \mathbf{f}_{\text{non}}(\mathbf{q}, \dot{\mathbf{q}}), \tag{8.45}$$

where $\mathbf{M} = \mathbf{M}^\mathsf{T} > 0$, $\mathbf{D} = \mathbf{D}^\mathsf{T} \geq 0$, and $\mathbf{K} = \mathbf{K}^\mathsf{T} \geq 0$ are the mass, damping and stiffness matrices, $\hat{\mathbf{B}} = [\mathbf{1}_{m\times m} \quad \mathbf{0}]^\mathsf{T}$ is the input matrix, $\mathbf{f}_{\text{non}} = [\mathbf{f}_{\text{non},\theta}^\mathsf{T} \quad \mathbf{f}_{\text{non,e}}^\mathsf{T}]^\mathsf{T}$ are the nonlinear inertial and Coriolis forces, $\boldsymbol{\tau} \in \mathbb{R}^m$ are the joint torques, and $\mathbf{q} \in \mathbb{R}^{n_\text{p}}$ are the generalized coordinates. The generalized coordinates are partitioned to joint angles $\boldsymbol{\theta} = [\theta_1 \quad \cdots \quad \theta_m]^\mathsf{T} \in \mathbb{R}^m$ and elastic coordinates $\mathbf{q}_\text{e} \in \mathbb{R}^{n_\text{e}}$ such that $\mathbf{q} = [\boldsymbol{\theta}^\mathsf{T} \quad \mathbf{q}_\text{e}^\mathsf{T}]^\mathsf{T}$. Additionally,

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{\theta\theta} & \mathbf{M}_{\theta\text{e}} \\ \mathbf{M}_{\text{e}\theta} & \mathbf{M}_{\text{ee}} \end{bmatrix}, \tag{8.46}$$

and $\mathbf{K} = \text{diag}\,(\mathbf{0}_{m\times m}, \mathbf{K}_{\text{ee}})$. The plant dimensions for a two-link flexible joint manipulator are $m = 2$, $n_\text{e} = 2$, and $n_\text{p} = 4$.

The major source of nonlinearity in (8.45) is the second joint angle θ_2. The

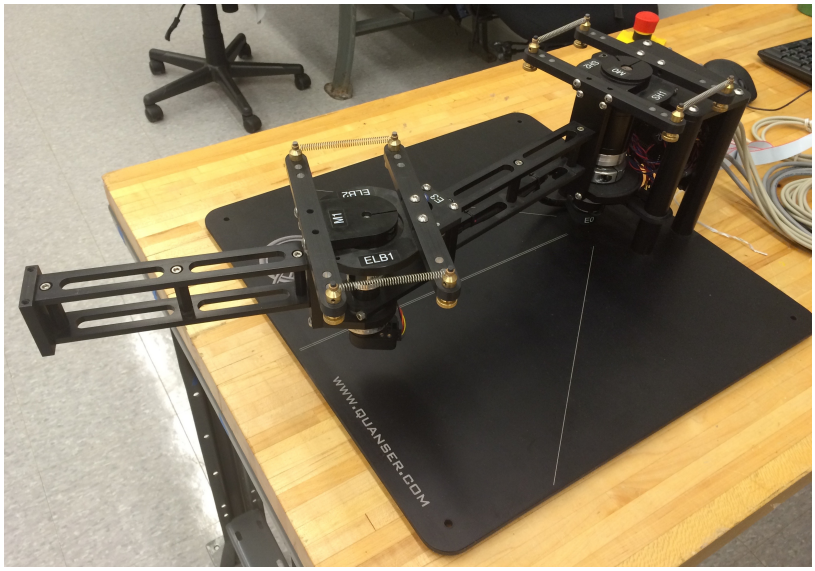


Figure 8.2: Quanser two-link flexible-joint manipulator.

mass matrix is dependent on $\cos(\theta_2)$ and the manipulator's second joint angle is in the range of $[-90°, 90°]$. Thus to cover the entire operating envelope of the second joint, set points can be chosen with $\theta_2 \in [0°, 90°]$. Three set points are chosen with $\theta_{2,i} \in \{0°, 45°, 90°\}$, and with all other states zero so that

$$\mathbf{q}_i = \begin{bmatrix} 0 & \theta_{2,i} & \mathbf{0} \end{bmatrix}^\mathsf{T}. \tag{8.47}$$

Equation (8.45) is linearized at $\mathbf{q}_i$ to form $\mathbf{A}_i$, $\mathbf{B}_{2,i}$, and $\mathbf{C}_{2,i}, i = 1, \ldots, 3$. The scheduling signals s_i are then determined by linear interpolation so that $\theta_2 = \sum_{i=1}^{N} s_i \theta_{2,i}$. For example, if $\theta_2 = 22.5°$, then $\mathbf{s} = [0.5 \;\; 0.5 \;\; 0]^\mathsf{T}$. The linearization of (8.45) in the form of (8.9) is

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{0} & \mathbf{1} \\ -\mathbf{M}(\mathbf{q}_i)^{-1}\mathbf{K} & -\mathbf{M}(\mathbf{q}_i)^{-1}\mathbf{D} \end{bmatrix}, \tag{8.48}$$

$$\mathbf{B}_{2,i} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{M}(\mathbf{q}_i)^{-1}\hat{\mathbf{B}} \end{bmatrix}, \tag{8.49}$$

$$\mathbf{C}_{2,i} = \begin{bmatrix} \mathbf{0}_{2\times 4} & \mathbf{1}_{2\times 2} & \mathbf{0}_{2\times 2} \end{bmatrix}, \tag{8.50}$$

where the measurement is the joint rates. All three controllers use the same weighting

matrices

$$\mathbf{B}_{1,i} = \begin{bmatrix} \mathbf{B}_{2,i} \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix} & \mathbf{0} \end{bmatrix}, \qquad \mathbf{D}_{11} = \mathbf{0}, \tag{8.51}$$

$$\mathbf{C}_1 = \begin{bmatrix} \mathbf{0} & 3\mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 0.3\mathbf{1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \qquad \mathbf{D}_{12} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \end{bmatrix}^{\mathsf{T}}, \tag{8.52}$$

$$\mathbf{D}_{21} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \end{bmatrix}, \tag{8.53}$$

such that the assumptions 1)-6) in Section 8.3.1 are satisfied, and so that the least biased comparison possible can be made between the three controllers.

The desired joint angles $\boldsymbol{\theta}_{\mathrm{d}}$ are designed by choosing joint angles $\boldsymbol{\theta}_{\mathrm{d}}(t_k)$ and $\boldsymbol{\theta}_{\mathrm{d}}(t_{k+1})$ at times t_k and t_{k+1}, for $k = 0, \ldots, 7$. The desired trajectory at time t is

$$\begin{aligned} \boldsymbol{\theta}_{\mathrm{d}}(t) = \Bigg[& 10 \left(\frac{t - t_k}{t_{k+1} - t_k} \right)^3 - 15 \left(\frac{t - t_k}{t_{k+1} - t_k} \right)^4 + \\ & 6 \left(\frac{t - t_k}{t_{k+1} - t_k} \right)^5 \Bigg] \left(\boldsymbol{\theta}_{\mathrm{d}}(t_{k+1}) - \boldsymbol{\theta}_{\mathrm{d}}(t_k) \right) + \boldsymbol{\theta}_{\mathrm{d}}(t_k). \end{aligned} \tag{8.54}$$

The time derivatives of (8.54) yield $\dot{\boldsymbol{\theta}}_{\mathrm{d}}$ and $\ddot{\boldsymbol{\theta}}_{\mathrm{d}}$. The trajectory is designed so that several joint positions and rates are attained by using (8.54) for various intervals. The trajectories for both joints are shown in Fig. 8.4, where

$$\begin{aligned} & t_0 = 0 \text{ s}, \ t_1 = 2.5 \text{ s}, && \boldsymbol{\theta}_{\mathrm{d}}(t_0) = \boldsymbol{\theta}_{\mathrm{d}}(t_1) = [0^\circ \ \ 0^\circ]^{\mathsf{T}}, \\ & t_2 = 5 \text{ s}, && \boldsymbol{\theta}_{\mathrm{d}}(t_2) = [-30^\circ \ \ -30^\circ]^{\mathsf{T}}, \\ & t_3 = 7.5 \text{ s}, && \boldsymbol{\theta}_{\mathrm{d}}(t_3) = [-60^\circ \ \ -30^\circ]^{\mathsf{T}}, \\ & t_4 = 10 \text{ s}, && \boldsymbol{\theta}_{\mathrm{d}}(t_4) = [60^\circ \ \ 80^\circ]^{\mathsf{T}}, \\ & t_5 = 12.5 \text{ s}, \ t_6 = 15 \text{ s}, && \boldsymbol{\theta}_{\mathrm{d}}(t_5) = \boldsymbol{\theta}_{\mathrm{d}}(t_6) = [-30^\circ \ \ -45^\circ]^{\mathsf{T}}, \\ & t_7 = 17.5 \text{ s}, \ t_8 = 20 \text{ s}, && \boldsymbol{\theta}_{\mathrm{d}}(t_7) = \boldsymbol{\theta}_{\mathrm{d}}(t_8) = [0^\circ \ \ 0^\circ]^{\mathsf{T}}. \end{aligned}$$

8.4.2 Control Structure and Synthesis

The passive map of (8.45) is joint torques to joint rates $\boldsymbol{\tau} \mapsto \dot{\boldsymbol{\theta}}$. A proportional control pre-wrap $\mathbf{K}_{\mathrm{p}}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathrm{d}})$ is added to the system, as this does not violate the

passive map [96]. The control has the structure

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{\text{ff}} - \mathbf{K}_{\text{p}}(\boldsymbol{\theta} - \boldsymbol{\theta}_{\text{d}}) - \mathbf{y}_{\text{c}}, \tag{8.55}$$

where $\mathbf{K}_{\text{p}} = \text{diag}\,(270, 8)$ N $\cdot$ m. The feedforward control is given by

$$\boldsymbol{\tau}_{\text{ff}} = \mathbf{M}_{\theta\theta}(\boldsymbol{\theta}_{\text{d}})\ddot{\boldsymbol{\theta}}_{\text{d}} - \mathbf{f}_{\text{non},\theta}(\boldsymbol{\theta}_{\text{d}}, \dot{\boldsymbol{\theta}}_{\text{d}}), \tag{8.56}$$

which attempts to cancel out the nominal rigid-body dynamics based on the desired trajectory, where $\mathbf{M}_{\theta\theta}$ is given by (8.46). The term $\mathbf{f}_{\text{non},\theta}$ is the joint angle partition of $\mathbf{f}_{\text{non}}$ evaluated along the desired trajectory. The torque $\mathbf{y}_{\text{c}}$ is the output of the chosen feedback controller $\mathbf{y}_{\text{c}} = \boldsymbol{\mathcal{G}}_{\text{c}}\mathbf{u}_{\text{c}}$, where $\mathbf{u}_{\text{c}} = \dot{\boldsymbol{\theta}} - \dot{\boldsymbol{\theta}}_{\text{d}}$. The controllers are synthesized with the plant description (8.48)-(8.53). The VSP-LPV controller (8.10)-(8.11) uses values of $\bar{\delta} = 10^{-5}$ and $\bar{\epsilon} = 0.3$ as these had the best closed-loop performance. Both the $\mathcal{H}_2$-VSP and $\mathcal{H}_\infty$-VSP controllers minimize (8.24) subject to (8.27)-(8.26) to determine $\mathbf{B}_{\text{c},i}, i = 1, \ldots, N$. In addition, the LPV controller filters the input to the plant with first order filter of 30 Hz so that the effective $\mathbf{B}_2$ matrix is constant. This filter increases the effective number of states of the controller. A constant $\mathbf{B}_2$ matrix is one of the limitations of the LPV control synthesis method. Controller synthesis and solutions to all LMIs are generated in Matlab using SDPT3 [117, 118] with the Yalmip interface [57].

The maximum singular values and minimum Hermitian parts of the controllers defined at the second set point, $\theta_2 = 45°$, are shown in Fig. 8.3. These plots serve to gain insight in the controller dynamics. Due to similar plant dynamics, all three controllers have a peak in singular values at slightly less than 10 rad/s. Since the $\mathcal{H}_2$-VSP and $\mathcal{H}_\infty$-VSP are constructed using (8.13) and (8.15), it is expected that the controller defined at a set point has a strictly positive minimum Hermitian part since (8.13) and (8.15) are similar conditions to the Kalman-Yakubovich-Popov Lemma [89] for one set point, that is when $N = 1$. Fig. 8.3 shows that the two VSP-LPV controllers are SPR-like at the set points since the minimum Hermitian parts are positive, and the LPV controller is not. However, it must be noted that "SPR" can only be said in the context when the controller is fixed at a set point. These controllers are not time-invariant and thus the VSP property of the VSP-LPV controller for all scheduling signals is important, not the SPR property at a single set point.

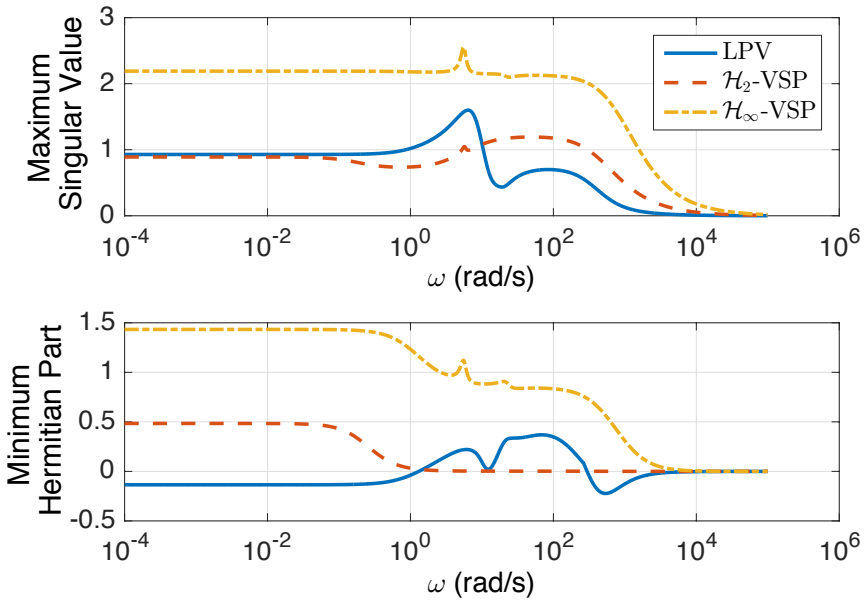


Figure 8.3: Comparison of the maximum singular values and minimum Hermitian parts of the $\mathcal{H}_2$-VSP, the $\mathcal{H}_\infty$-VSP and LPV controllers at set point 2, where $\theta_2 = 45°$.

8.4.3 Experimental Results

The results for the system (8.45) with control (8.55) with trajectory defined in (8.54) are shown in Fig. 8.4, Fig. 8.5, and Fig. 8.6. Figure 8.4 shows the desired trajectory of the manipulator and that each controller is able to follow closely. Figure 8.5 shows that each controller exhibit similar error tendencies, where the angle error is $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \boldsymbol{\theta}_\mathrm{d}$. In particular, the systems with the $\mathcal{H}_2$-VSP and $\mathcal{H}_\infty$-VSP controllers have less angle error than the LPV controller, and the $\mathcal{H}_\infty$-VSP has the least amount of angular rate error. The root mean square (RMS) angle error and error rates are tabulated in Table 8.2 for more detail. In general, all three controllers have less than 0.06 deg error for the joint angles and 3.5 deg/s angular rates. Although the $\mathcal{H}_2$-VSP controller has the greatest error for $\dot{\tilde{\theta}}_2$, the VSP-LPV controllers have less joint angle error than the self-scheduled LPV controller. The $\mathcal{H}_2$-VSP also does not have any relation to $\mathcal{H}_\infty$ theory. Comparing the two controllers that do have $\mathcal{H}_\infty$ roots, that is the LPV controller and the $\mathcal{H}_\infty$-VSP controller, the $\mathcal{H}_\infty$-VSP controller has less angle and angular rate errors, demonstrating the effectiveness of this new control technique. While we use the same weights for the all three controllers for simplicity of exposition and for fair comparison, it seems that the weights favour the $\mathcal{H}_\infty$-based synthesis methods ($\mathcal{H}_\infty$-VSP and LPV). Retuning the weighting matrices may yield decreased performance for the $\mathcal{H}_\infty$-based controllers and increased

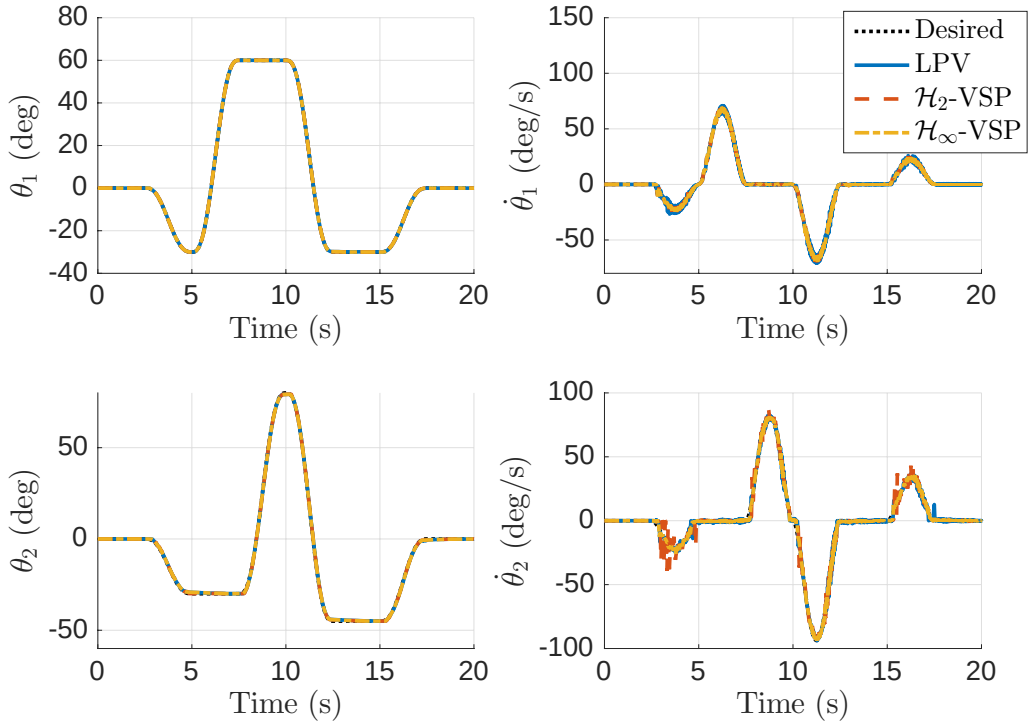


Figure 8.4: Comparison joint angles and joint rates.

Table 8.2: RMS error of angle and angle error tracking for two-link flexible joint experiment.

	RMS Angle Error (deg)		RMS Angle Rate Error (deg/s)	
Control Method	$\tilde{\theta}_1$	$\tilde{\theta}_2$	$\dot{\tilde{\theta}}_1$	$\dot{\tilde{\theta}}_2$
LPV	0.0652	0.3866	1.4417	1.8596
$\mathcal{H}_2$-VSP	0.0408	0.3463	1.0313	3.4204
$\mathcal{H}_\infty$-VSP	0.0434	0.3695	0.8871	1.3289

performance for the $\mathcal{H}_2$-VSP controller. For a specific application, we recommend testing different controller synthesis methods with different weighting matrices. The controllers we present in this chapter are alternatives to existing methods and may perform better or worse than other existing control methods. As all applications are different, it is the control engineer's responsibility to decide which implementation is best.

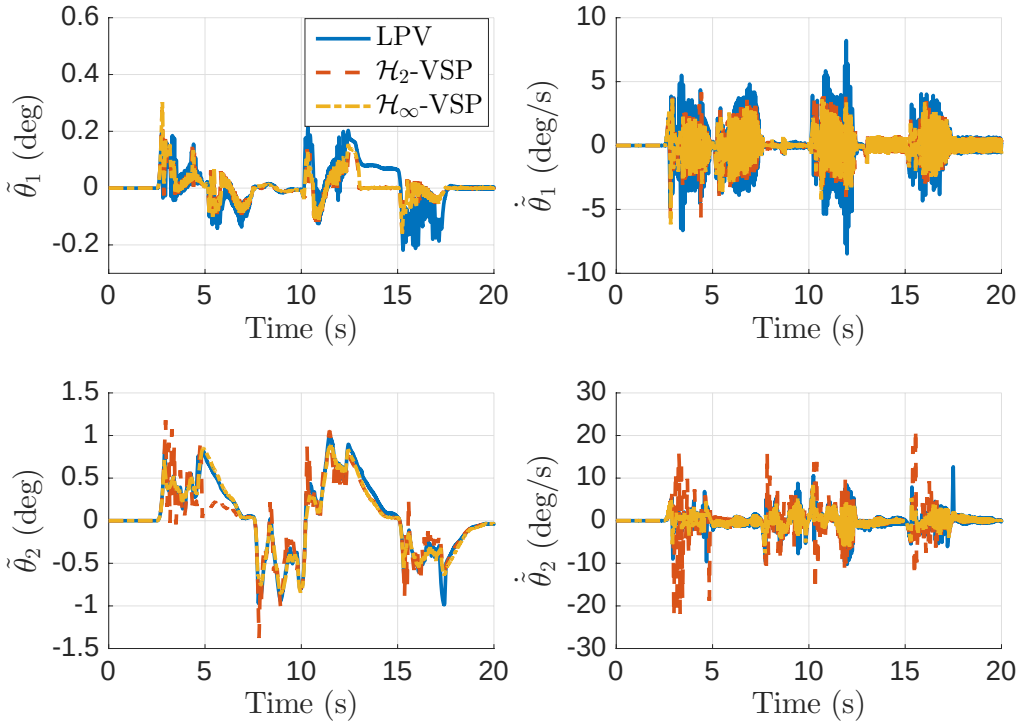


Figure 8.5: Comparison of joint angle errors and error rates.

8.5 Closing Remarks

When approximate LPV plant representations are required to make LPV controller synthesis tractable, systematic and mathematically rigorous methods to determine stability of the nonlinear system remain elusive [22]. While these issues cannot be generally solved yet, advances are made in this chapter by considering passive systems. Although an LPV approximation is used for controller synthesis, the resulting VSP-LPV controller guarantees closed-loop stability of the original nonlinear system with finite gain. In short, this chapter makes large strides to close the gap between LPV control and passive systems theory.

The VSP-LPV controller has affine parameter dependence and two different control synthesis methods are presented. An $\mathcal{H}_2$-VSP method and a $\mathcal{H}_\infty$-VSP method are introduced that are based on $\mathcal{H}_2$ and $\mathcal{H}_\infty$ control theory. These synthesis methods have the flexibility to tune the controller at each set point, and make use of LMIs, which are numerically efficient to solve. Both these controllers are able to provide joint-angle tracking for a two-link flexible-joint manipulator, with performance that improves upon polytopic LPV control.

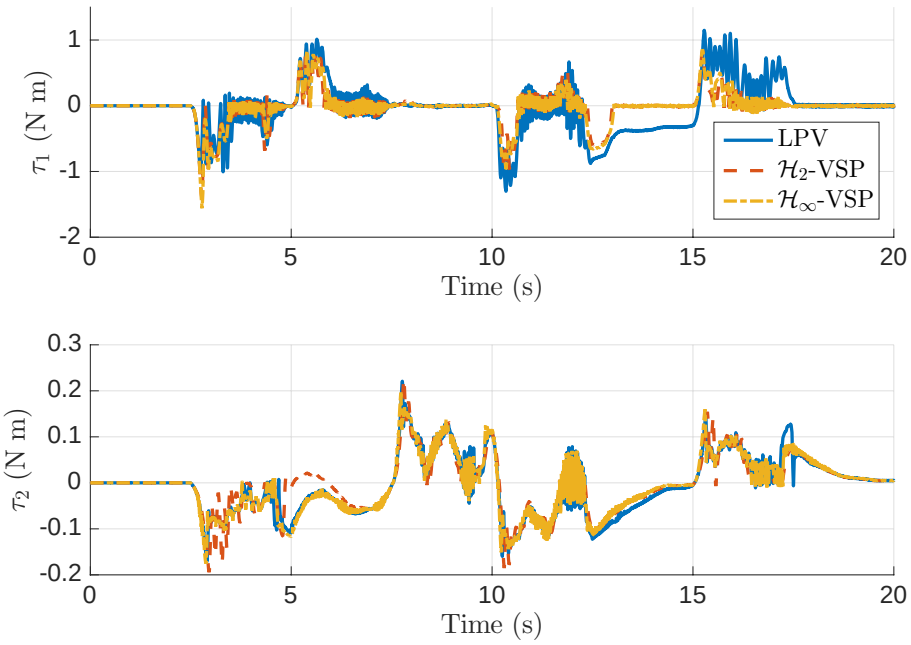


Figure 8.6: Comparison of joint torques from each controller.

Several avenues for future research exist. For example, the control synthesis method used for the $\mathcal{H}_\infty$-VSP controller minimizes the $\mathcal{H}_2$ norm between the nominal $\mathcal{H}_\infty$ controllers at each set point and a controller that satisfies Theorem 8.1. Although this synthesis method may retain some features of the nominal $\mathcal{H}_\infty$ controller of the set point, no real performance guarantees for the VSP-LPV controller exist (only stability guarantees). A controller synthesis method that determines an upper bound on a closed-loop performance criteria is desirable. Additionally, a common Lyapunov matrix $\mathbf{P}_c$ is used for controller synthesis and for the proof that the controller is indeed VSP. A common Lyapunov matrix approach is known to increase the conservatism of LPV controllers, and research into extensions of Theorem 8.1 to accommodate a parameter-dependent Lyapunov matrix may be possible.

Chapter 9

Interior Conic Polytopic Systems

9.1 Introduction

Input-output theories have found widespread usage in robust control. The Small Gain Theorem is at the heart of robust linear parameter varying (LPV) control. Specifically, the Bounded Real Lemma was extended for application to parameter varying systems, enabling the statement of sufficient conditions to derive stabilizing LPV controllers [26, p. 15]. Two examples that are particularly relevant to this chapter are [102, 119], where polytopic controller synthesis is discussed. In [102], subcontrollers at each vertex are synthesized while simultaneously satisfying a global constraint for stability, whereas in [119], subcontrollers at each vertex are first synthesized, and then a global constraint for stability is imposed in an effort to reduce computational complexities.

Initial attempts at deriving LPV controllers had shortcomings such as limits on parameter-variation rates and synthesis complexity, but recent efforts have greatly improved LPV control effectiveness and synthesis tractability [22, 24, 25, 26]. Nonetheless, LPV plant descriptions are often approximations of nonlinear systems, and a controller synthesized to stabilize one of these approximations is not guaranteed to stabilize the original nonlinear system [22]. For passive systems, the Passivity Theorem is used to circumvent this problem [3, 98, 97]. Strictly passive controllers can guarantee closed-loop input-output stability of passive plants, even if an approximate model is used for controller synthesis.

Despite advances in control theory and controller synthesis rooted in the Passivity and Small Gain Theorems, relying on these two theorems introduces limitations. For instance, relying on the Small Gain Theorem may lead to conservative results owing to the fact that the analysis relies on the supremum of the operator in question.

Weighting transfer functions can be used to emphasize certain frequency bands, but the conservatism still remains. When stabilizing systems using the Passivity Theorem, plants that are nominally considered passive can have so-called passivity violations due to discretization, time-delay, and sensor noise [120, 18]. With any one or a combination of these violations, results obtained using the Passivity Theorem may be void. As it turns out, the Conic Sector Theorem is a more general input-output stability theorem, of which small gain and passivity are special cases [90]. Synthesizing a controller with the Conic Sector Theorem can help avoid the conservative nature of controllers designed using the Small Gain Theorem, and provide a remedy for situations where the plant features a passivity violation and the Passivity Theorem is not applicable.

The design of conic controllers has been studied for linear systems [91, 18, 19]. The design approach considered in [91, 18, 19] begins by determining the conic bounds of the plant using the Conic Sector Lemma [92], then choosing appropriate conic bounds of the controller using the Conic Sector Theorem [90], and finally synthesizing a linear controller that satisfies these conic bounds. A similar approach is undertaken in this chapter specific to polytopic systems, which are a type of LPV systems. The challenges to overcome include the fact that currently, there are no LMI conditions to determine conic bounds for polytopic systems, and no synthesis methods for conic polytopic controllers.

In short, the two main contributions of this chapter are providing a means to assess the conic bounds of polytopic systems using an LMI as stated in Theorem 9.2, and a method to design polytopic conic controllers in Section 9.3. A third contribution is the demonstration of the conic polytopic controller on a heat exchanger. The derivation of conic bounds for polytopic systems are in Section 9.2, the numerical example is in Section 9.4, and closing remarks are in Section 9.5.

9.2 Conic Polytopic Systems

9.2.1 System Architecture

Recall a matrix polytope is defined as the convex hull of a finite number of matrices $\mathbf{H}_i$, defined as

$$\mathrm{Co}\{\mathbf{H}_i, i = 1, \ldots, N\} = \left\{ \sum_{i=1}^{N} s_i \mathbf{H}_i \;\middle|\; s_i \geq 0,\ \sum_{i=1}^{N} s_i = 1 \right\}.$$

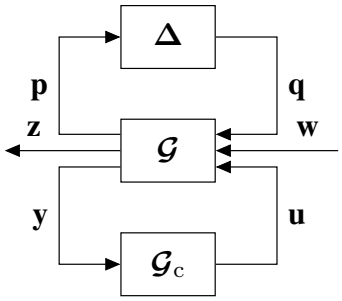
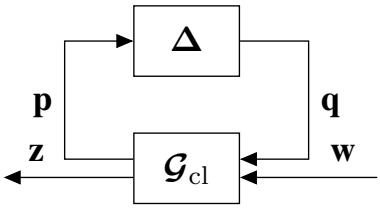
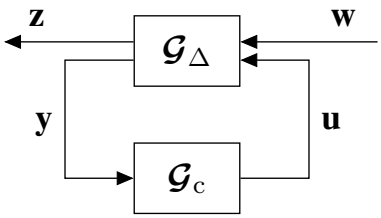


Figure 9.1: Standard problem block diagram with an uncertainty block.



(a) Closed-loop control framework relying on the Small Gain Theorem.



(b) Closed-loop control framework relying on the Conic Sector Theorem.

Figure 9.2: Upper and lower LFTs for controller synthesis.

In this chapter, only polytopic systems are examined. Consider the polytopic system shown in Fig. 9.1, given by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_1\mathbf{w} + \mathbf{B}_2\mathbf{u} + \mathbf{B}_3\mathbf{q}, \tag{9.1}$$

$$\mathbf{z} = \mathbf{C}_1\mathbf{x} + \mathbf{D}_{11}\mathbf{w} + \mathbf{D}_{12}\mathbf{u} + \mathbf{D}_{13}\mathbf{q}, \tag{9.2}$$

$$\mathbf{y} = \mathbf{C}_2\mathbf{x} + \mathbf{D}_{21}\mathbf{w}, \tag{9.3}$$

$$\mathbf{p} = \mathbf{C}_3\mathbf{x} + \mathbf{D}_{31}\mathbf{w} + \mathbf{D}_{32}\mathbf{u} + \mathbf{D}_{33}\mathbf{q}, \tag{9.4}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the system state, $\mathbf{z} \in \mathbb{R}^{n_z}$ is the performance variable, $\mathbf{w} \in \mathbb{R}^{n_w}$ is the exogenous signal that can include noise, $\mathbf{y} \in \mathbb{R}^m$ is the measurement variable, $\mathbf{p} \in \mathbb{R}^{n_p}$ is the input to the uncertainty block, $\mathbf{q} \in \mathbb{R}^{n_q}$ is the output of the uncertainty block, and $\mathbf{u} \in \mathbb{R}^m$ is the system input. Each system matrix is a function of a scheduling signal s_i, such that

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 \\ \mathbf{C}_1 & \mathbf{D}_{11} & \mathbf{D}_{12} & \mathbf{D}_{13} \\ \mathbf{C}_2 & \mathbf{D}_{21} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_3 & \mathbf{D}_{31} & \mathbf{D}_{32} & \mathbf{D}_{33} \end{bmatrix} = \sum_{i=1}^{N} s_i(\boldsymbol{\sigma}, \mathbf{x}, t) \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_{1,i} & \mathbf{B}_{2,i} & \mathbf{B}_{3,i} \\ \mathbf{C}_{1,i} & \mathbf{D}_{11,i} & \mathbf{D}_{12,i} & \mathbf{D}_{13,i} \\ \mathbf{C}_{2,i} & \mathbf{D}_{21,i} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_{3,i} & \mathbf{D}_{31,i} & \mathbf{D}_{32,i} & \mathbf{D}_{33,i} \end{bmatrix}, \tag{9.5}$$

where

$$0 \leq s_i(\boldsymbol{\sigma}, \mathbf{x}, t) \leq 1, \quad \sum_{i=1}^{N} s_i(\boldsymbol{\sigma}, \mathbf{x}, t) = 1. \tag{9.6}$$

The scheduling signals $s_i(\boldsymbol{\sigma}, \mathbf{x}, t)$ can be a function of time $t \in \mathbb{R}_{\geq 0}$, a function of state $\mathbf{x} \in \mathbb{R}^n$, or a function of an exogenous signal $\boldsymbol{\sigma} \in \mathbb{R}^{n_\sigma}$.

For robust control design using $\mathcal{H}_\infty$-based techniques, the $\mathcal{H}_\infty$ norm of the closed-loop system of the plant $\boldsymbol{\mathcal{G}}$ and the controller $\boldsymbol{\mathcal{G}}_\mathrm{c}$, denoted $\boldsymbol{\mathcal{G}}_\mathrm{cl}$, is determined. Closed-loop input-output stability with $\boldsymbol{\Delta}$ is guaranteed via Theorem 7.3 when $\|\boldsymbol{\Delta}\|_\infty \|\boldsymbol{\mathcal{G}}_\mathrm{cl}\|_\infty < 1$. The block diagram for this synthesis is described by Fig. 9.2a. Controller synthesis methods using the Small Gain Theorem are extensively covered in the literature. See [27] and [102] for general reference on control synthesis using LMIs and polytopic systems, respectively.

For robust control design rooted in the conic-systems framework, which is the focus of this chapter, the uncertainty and the plant are lumped together as $\boldsymbol{\mathcal{G}}_\Delta$, as shown in Fig. 9.2b. The conic bounds of the system $\boldsymbol{\mathcal{G}}_\Delta$ are first determined for the mapping $\mathbf{u} \mapsto \mathbf{y}$. Theorem 9.2 in Section 9.2.2 provides sufficient conditions to determine conic bounds for polytopic systems. Appropriate controller bounds are then determined via Theorem 7.10 to ensure closed-loop input-output stability. Theorem 9.2 is used once again when deriving a polytopic controller.

9.2.2 Conic Bounds for Polytopic Systems

Conic bounds for a system are defined for a specific input-output pair. For the plant (9.1)-(9.4), the input-output pair for conic bounds is $\mathbf{u} \mapsto \mathbf{y}$, with minimal state space realization

$$\boldsymbol{\mathcal{G}} : \begin{cases} \dot{\mathbf{x}} = \mathbf{A}(\mathbf{s})\mathbf{x} + \mathbf{B}(\mathbf{s})\mathbf{u} \\ \mathbf{y} = \mathbf{C}(\mathbf{s})\mathbf{x} \end{cases} \tag{9.7}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{u} \in \mathbb{R}^m$, $\mathbf{y} \in \mathbb{R}^m$, $\mathbf{s} = [s_1 \ \cdots \ s_N]^\mathsf{T}$, and

$$\begin{bmatrix} \mathbf{A}(\mathbf{s}) & \mathbf{B}(\mathbf{s}) \\ \mathbf{C}(\mathbf{s}) & \mathbf{0} \end{bmatrix} = \sum_{i=1}^{N} s_i(\boldsymbol{\sigma}, \mathbf{x}, t) \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{C}_i & \mathbf{0} \end{bmatrix}, \tag{9.8}$$

where $\mathbf{s}$ satisfies (9.6). The feedthrough matrix, $\mathbf{D}_{22}$, is assumed to be zero as physical systems exhibit some sort of roll-off in gain at higher frequency. If a model does contain a $\mathbf{D}_{22}$ matrix, the measurement $\mathbf{y}$ can be filtered to remove the feedthrough term.

Lemma 9.1. *Given* $\mathbf{y}(t) = \sum_{i=1}^{N} s_i(\boldsymbol{\sigma}, \mathbf{x}, t)\mathbf{y}_i(t)$, *where* $\mathbf{y}(t) \in \mathcal{L}_{2e}$, $\mathbf{y}_i = \mathbf{C}_i\mathbf{x}(t)$, *and where scheduling signals satisfy* (9.6), *then*

$$-\|\mathbf{y}(t)\|_{2T}^2 \geq \sum_{i=1}^{N} -\left\|\sqrt{s_i(\boldsymbol{\sigma}, \mathbf{x}, t)}\mathbf{y}_i(t)\right\|_{2T}^2. \tag{9.9}$$

Proof. See Appendix B.4. □

Theorem 9.2. *Consider the system* $\mathbf{y} = \boldsymbol{\mathcal{G}}\mathbf{u}$ *described by* (9.7), *where* $\mathbf{A}(\mathbf{s})$, $\mathbf{B}(\mathbf{s})$, *and* $\mathbf{C}(\mathbf{s})$ *satisfy* (9.6) *and* (9.8). *For* $a, b \in \mathbb{R}$, $a < 0 < b$, *if the LMI in* $\mathbf{P}$

$$\begin{bmatrix} \mathbf{P}\mathbf{A}_i + \mathbf{A}_i^{\mathsf{T}}\mathbf{P} + \mathbf{C}_i^{\mathsf{T}}\mathbf{C}_i & \mathbf{P}\mathbf{B}_i - \frac{a+b}{2}\mathbf{C}_i^{\mathsf{T}} \\ \star & ab\mathbf{1} \end{bmatrix} = -\begin{bmatrix} \mathbf{L}_i^{\mathsf{T}} \\ \mathbf{W}_i^{\mathsf{T}} \end{bmatrix}\begin{bmatrix} \mathbf{L}_i & \mathbf{W}_i \end{bmatrix} \leq 0 \tag{9.10}$$

is satisfied for some real matrices $\mathbf{P} = \mathbf{P}^{\mathsf{T}} > 0$, $\mathbf{L}_i$ *and* $\mathbf{W}_i$ *for* $i = 1, \ldots, N$, *then the system* $\boldsymbol{\mathcal{G}}$ *is in* $\text{cone}[a, b]$.

Proof. In this proof, the argument $\mathbf{s}$ associated with $(\mathbf{A}(\mathbf{s}), \mathbf{B}(\mathbf{s}), \mathbf{C}(\mathbf{s}))$ is dropped for brevity. First note that (9.10) implies

$$\begin{aligned} \mathbf{P}\mathbf{A}_i + \mathbf{A}_i^{\mathsf{T}}\mathbf{P} + \mathbf{C}_i^{\mathsf{T}}\mathbf{C}_i &= -\mathbf{L}_i^{\mathsf{T}}\mathbf{L}_i, \\ \mathbf{P}\mathbf{B}_i - \frac{a+b}{2}\mathbf{C}_i^{\mathsf{T}} = -\mathbf{L}_i^{\mathsf{T}}\mathbf{W}_i, \qquad ab\mathbf{1} &= -\mathbf{W}_i^{\mathsf{T}}\mathbf{W}_i. \end{aligned} \tag{9.11}$$

Consider the time derivative of the Lyapunov-like function $\mathcal{V} = \frac{1}{2}\mathbf{x}^{\mathsf{T}}\mathbf{P}\mathbf{x}$,

$$\dot{\mathcal{V}} = \frac{1}{2}\mathbf{x}^{\mathsf{T}}(\mathbf{P}\mathbf{A} + \mathbf{A}^{\mathsf{T}}\mathbf{P})\mathbf{x} + \mathbf{x}^{\mathsf{T}}\mathbf{P}\mathbf{B}\mathbf{u}. \tag{9.12}$$

Substituting (9.11) in to (9.12) yields

$$\begin{aligned} \dot{\mathcal{V}} &= \sum_{i=1}^{N} s_i\Bigg[-\frac{1}{2}\mathbf{x}^{\mathsf{T}}\mathbf{C}_i^{\mathsf{T}}\mathbf{C}_i\mathbf{x} + \mathbf{x}^{\mathsf{T}}\left(\frac{a+b}{2}\right)\mathbf{C}_i\mathbf{u} - \frac{1}{2}\mathbf{x}^{\mathsf{T}}\mathbf{L}_i^{\mathsf{T}}\mathbf{L}_i\mathbf{x} - \mathbf{x}^{\mathsf{T}}\mathbf{L}_i^{\mathsf{T}}\mathbf{W}_i\mathbf{u} \\ &\quad - \frac{1}{2}\mathbf{u}^{\mathsf{T}}\mathbf{W}_i^{\mathsf{T}}\mathbf{W}_i\mathbf{u} + \frac{1}{2}\mathbf{u}^{\mathsf{T}}\mathbf{W}_i^{\mathsf{T}}\mathbf{W}_i\mathbf{u}\Bigg] \\ &= \sum_{i=1}^{N} s_i\left[-\frac{1}{2}\mathbf{y}_i^{\mathsf{T}}\mathbf{y}_i - \frac{1}{2}(\mathbf{L}_i\mathbf{x} + \mathbf{W}_i\mathbf{u})^{\mathsf{T}}(\mathbf{L}_i\mathbf{x} + \mathbf{W}_i\mathbf{u})\right] - \frac{1}{2}ab\mathbf{u}^{\mathsf{T}}\mathbf{u} + \left(\frac{a+b}{2}\right)\mathbf{y}^{\mathsf{T}}\mathbf{u}. \end{aligned} \tag{9.13}$$

Integrating (9.13) in time from 0 to $0 < T < \infty$ gives

$$\mathcal{V}(T) - \mathcal{V}(0) = \sum_{i=1}^{N} \left[-\frac{1}{2} \|\sqrt{s_i}\mathbf{y}_i\|_{2T}^2 - \frac{1}{2} \|\sqrt{s_i}\left(\mathbf{L}_i\mathbf{x} + \mathbf{W}_i\mathbf{u}\right)\|_{2T}^2 \right] - \frac{ab}{2} \|\mathbf{u}\|_{2T}^2 + \left(\frac{a+b}{2}\right) \langle \mathbf{y}, \mathbf{u} \rangle_T .$$

Rearranging yields

$$-\sum_{i=1}^{N} \|\sqrt{s_i}\mathbf{y}_i\|_{2T}^2 + (a+b) \langle \mathbf{y}, \mathbf{u} \rangle_T - ab \|\mathbf{u}\|_{2T}^2 = +2(\mathcal{V}(T) - \mathcal{V}(0)) + \sum_{i=1}^{N} \|\sqrt{s_i}\left(\mathbf{L}_i\mathbf{x} + \mathbf{W}_i\mathbf{u}\right)\|_{2T}^2 . \tag{9.14}$$

Since $2(\mathcal{V}(T) - \mathcal{V}(0)) + \sum_{i=1}^{N} \left\|\sqrt{s_i}\left(\mathbf{L}_i\mathbf{x} + \mathbf{W}_i\mathbf{u}\right)\right\|_{2T}^2 \geq -2\mathcal{V}(0)$, using Lemma 9.1 reduces (9.14) to

$$-\|\mathbf{y}\|_{2T}^2 + (a+b) \langle \mathbf{y}, \mathbf{u} \rangle_T - ab \|\mathbf{u}\|_{2T}^2 \geq \beta, \tag{9.15}$$

where $\beta = -2\mathcal{V}(0)$ only depends on initial conditions. Comparing (9.15) to (7.6) proves Theorem 9.2. □

Corollary 9.3. *Consider the system* $\mathbf{y} = \boldsymbol{\mathcal{G}}\mathbf{u}$ *described by* (9.7). *For* $a, b \in \mathbb{R}$, $a < 0 < b$, *if the LMI in* $\tilde{\mathbf{P}}$

$$\begin{bmatrix} \tilde{\mathbf{P}}\mathbf{A}_i + \mathbf{A}_i^\mathsf{T}\tilde{\mathbf{P}} + \frac{1}{b}\mathbf{C}_i^\mathsf{T}\mathbf{C}_i & \tilde{\mathbf{P}}\mathbf{B}_i - \frac{1}{2}\left(\frac{a}{b}+1\right)\mathbf{C}_i^\mathsf{T} \\ \star & a\mathbf{1} \end{bmatrix} \leq 0 \tag{9.16}$$

is satisfied for $i = 1, \ldots, N$, *and* $\tilde{\mathbf{P}} > 0$, *then the system* $\boldsymbol{\mathcal{G}}$ *is in* $\text{cone}[a, b]$.

Proof. Multiplying (9.10) by $\frac{1}{b} > 0$ results in

$$\begin{bmatrix} \frac{1}{b}\mathbf{P}\mathbf{A}_i + \mathbf{A}_i^\mathsf{T}\mathbf{P}\frac{1}{b} + \frac{1}{b}\mathbf{C}_i^\mathsf{T}\mathbf{C}_i & \frac{1}{b}\mathbf{P}\mathbf{B}_i - \frac{1}{2}\left(\frac{a}{b}+1\right)\mathbf{C}_i^\mathsf{T} \\ \star & -a\mathbf{1} \end{bmatrix} \leq 0.$$

Using the change of variable $\tilde{\mathbf{P}} = \frac{1}{b}\mathbf{P} > 0$ yields (9.16). Since change of variable and multiplication are reversible, (9.16) also implies (9.10). Thus, if $\boldsymbol{\mathcal{G}}$ satisfies Corollary 9.3, then $\boldsymbol{\mathcal{G}}$ also satisfies Theorem 9.2. □

Both Theorem 9.2 and Corollary 9.3 are sufficient conditions for $\boldsymbol{\mathcal{G}} \in \text{cone}[a, b]$. Corollary 9.3 is a reformulation of Theorem 9.2, and may be useful in situations with

a large b. A large b can cause (9.10) to become ill-conditioned, whereas a large b does not cause problems for (9.16).

9.2.3 Determining Conic Bounds

The matrix inequality (9.10) is nonlinear in a and b, and thus solving for an a and b directly given the polytopic plant $\boldsymbol{\mathcal{G}}$ is not possible. Three methods of determining tight bounds a and b exists. They consist of

1. fixing $a = -\infty$, minimizing b, then fixing b, and maximizing a,
2. fixing $b = \infty$, maximizing a, then fixing a, and minimizing b, and
3. defining the conic radius $r = \frac{b-a}{2}$ and conic centre, $c = \frac{a+b}{2}$, and then minimizing the conic radius.

As [18] shows for the LTI case, each method yields different conic bounds. In the polytopic case, the same holds true. Fixing $b = \infty$, and then maximizing a is preferred to determine plant conic bounds because this results in a controller with the least conservative gain. To understand why, note that the controller conic bound $b_c = -\frac{1}{a}$ is related to controller gain, and thus an a closest to zero as possible is desired. When b is set to ∞, Corollary 9.3 is used.

To determine conic bounds using the radius and centre, r and c respectively, consider that

$$\begin{bmatrix} \mathbf{P}\mathbf{A}_i + \mathbf{A}_i^\mathsf{T}\mathbf{P} + \mathbf{C}_i^\mathsf{T}\mathbf{C}_i & \mathbf{P}\mathbf{B}_i - c\mathbf{C}_i^\mathsf{T} \\ \star & -\kappa\mathbf{1} \end{bmatrix} \leq 0 \tag{9.17}$$

is equivalent to (9.10), where $\kappa = -ab$. Note that

$$r^2 = \frac{(a-b)^2}{4} = \frac{(a+b)^2 - 4ab}{4} = c^2 + \kappa, \tag{9.18}$$

and thus minimizing (9.18) results in the minimum conic radius. Since (9.18) is quadratic, it can be transformed to a linear objective along with an LMI constraint by introducing a variable z, and then minimizing z subject to the constraint

$$\begin{bmatrix} z - \kappa & c \\ \star & 1 \end{bmatrix} \geq 0, \tag{9.19}$$

where (9.19) is derived from $z \geq c^2 + \kappa$ using the Schur complement.

9.3 Design of Polytopic Conic Controllers

Passivity-based synthesis for affine controllers is discussed in Chapter 8. The conic-polytopic controller synthesis considered in this chapter is an adaptation of the synthesis methods presented in [3, 19, 18]. The goal is to determine a set of controller matrices,

$$\begin{bmatrix} \mathbf{A}_{\mathrm{c}} & \mathbf{B}_{\mathrm{c}} \\ \mathbf{C}_{\mathrm{c}} & \mathbf{0} \end{bmatrix} = \sum_{i=1}^{N} s_i(\boldsymbol{\sigma}, \mathbf{x}, t) \begin{bmatrix} \mathbf{A}_{\mathrm{c},i} & \mathbf{B}_{\mathrm{c},i} \\ \mathbf{C}_{\mathrm{c},i} & \mathbf{0} \end{bmatrix}, \tag{9.20}$$

where the signals s_i satisfy (9.6). In particular, an $\mathcal{H}_\infty$ controller is synthesized at each vertex of the polytope, and then the closest controller in an $\mathcal{H}_2$ sense that satisfies conic constraints is determined. Associated assumptions are

1. $(\mathbf{A}_i, \mathbf{B}_{2,i}, \mathbf{C}_{2,i})$ is stabilizable and detectable, and
2. $\mathbf{D}_{22} = \mathbf{0}$.

The resulting state space matrices from the $\mathcal{H}_\infty$ synthesis are

$$\begin{bmatrix} \mathbf{A}_{\mathrm{c},i} & \mathbf{L}_i \\ \mathbf{K}_i & \mathbf{0} \end{bmatrix},$$

where $\mathbf{A}_{\mathrm{c},i}$ from the $\mathcal{H}_\infty$ synthesis is also the $\mathbf{A}_{\mathrm{c},i}$ of the conic controller given in (9.20). The matrix $\mathbf{C}_{\mathrm{c},i}$ in (9.20) is set to $\mathbf{C}_{\mathrm{c},i} = \mathbf{K}_i$. The $\mathbf{B}_{\mathrm{c},i}$ matrix is determined by minimizing the $\mathcal{H}_2$-norm of the difference between the $\mathcal{H}_\infty$ controller and a controller that satisfies (9.10). Specifically, this optimization problem is given by minimizing

$$\mathcal{J}(\boldsymbol{\Pi}_{\mathrm{c}}, \mathbf{B}_{\mathrm{c},1}, \ldots, \mathbf{B}_{\mathrm{c},N}) = \sum_{i=1}^{N} \operatorname{tr}(\mathbf{B}_{\mathrm{c},i} - \mathbf{L}_i)^{\mathsf{T}} \mathbf{W}_i (\mathbf{B}_{\mathrm{c},i} - \mathbf{L}_i), \tag{9.21}$$

subject to

$$\begin{bmatrix} \boldsymbol{\Pi}_{\mathrm{c}} \mathbf{A}_{\mathrm{c},i}^{\mathsf{T}} + \mathbf{A}_{\mathrm{c},i} \boldsymbol{\Pi}_{\mathrm{c}} & \mathbf{B}_{\mathrm{c},i} & \mathbf{C}_{\mathrm{c},i} \boldsymbol{\Pi}_{\mathrm{c}} \\ \star & -\frac{(a_{\mathrm{c}} - b_{\mathrm{c}})^2}{4 b_{\mathrm{c}}} \mathbf{1} & -\frac{a_{\mathrm{c}} + b_{\mathrm{c}}}{2} \mathbf{1} \\ \star & \star & -b_{\mathrm{c}} \mathbf{1} \end{bmatrix} < 0, \tag{9.22}$$

for $i = 1, \ldots, N$, where (9.22) is derived from Corollary 9.3. See Appendix B.5 for the derivation. Equation (9.16) is bilinear in $\mathbf{B}_{\mathrm{c},i}$ and $\mathbf{P}$, and thus cannot be used for conic synthesis. The matrix $\mathbf{W}_i$ is the observability Grammian that satisfies

$$\mathbf{A}_{\mathrm{c},i}^{\mathsf{T}} \mathbf{W}_i + \mathbf{W}_i \mathbf{A}_{\mathrm{c},i} + \mathbf{C}_{\mathrm{c},i}^{\mathsf{T}} \mathbf{C}_{\mathrm{c},i} = \mathbf{0}. \tag{9.23}$$

The objective function (9.21) is chosen because when the difference between $\mathbf{B}_{\mathrm{c},i}$ and

$\mathbf{L}_i$ subject to the weight $\mathbf{W}_i$ is minimized, the difference between the $\mathcal{H}_\infty$ controller and the controller that satisfies (9.10) is minimized in an $\mathcal{H}_2$ sense.

The quadratic function (9.21) can be transformed to a linear function and an LMI constraint given by

$$\hat{\mathcal{J}}(\nu, \mathbf{Z}, \mathbf{\Pi}_\mathrm{c}, \mathbf{B}_{\mathrm{c},1}, \ldots, \mathbf{B}_{\mathrm{c},N}) = \nu, \tag{9.24}$$

constrained by

$$\nu \geq \mathrm{tr}(\mathbf{Z}), \tag{9.25}$$

and

$$\begin{bmatrix} \mathbf{Z} & (\mathbf{B}_{\mathrm{c},1} - \mathbf{L}_1)^\mathsf{T} & \cdots & (\mathbf{B}_{\mathrm{c},N} - \mathbf{L}_N)^\mathsf{T} \\ \star & \mathbf{W}_1^{-1} & \mathbf{0} & \mathbf{0} \\ \vdots & \star & \ddots & \vdots \\ \star & \star & \star & \mathbf{W}_N^{-1} \end{bmatrix} \geq 0. \tag{9.26}$$

Determining $\mathbf{B}_{\mathrm{c},i}$ is summarized by Problem 9.4.

Problem 9.4. The problem to determine $\mathbf{B}_{\mathrm{c},i}$ for $i = 1, \ldots, N$ is given by minimizing (9.24), subject to (9.25), (9.26), and (9.22). □

The controller scheduling signals and the plant scheduling signals can be different and arbitrarily chosen. The reason is that the plant will always lie within the plant conic sector, and the controller will always lie within the controller conic sector for all scheduling signals that satisfy (9.6). However, since the controller is designed using the plant, for performance it would only make sense to use the same scheduling signals for both the plant and controller.

9.4 Numerical Example

9.4.1 System Description

Consider the linearized model of a heat exchanger described by [121, pp. 55–60], shown in Fig. 9.3, with properties listed in Table 9.1. The nonhomogeneous differential equation describing the dynamics of the heat exchanger is given by

$$\dot{\mathbf{T}} = \mathbf{A}_\mathrm{p}\mathbf{T} + \mathbf{B}_\mathrm{p}T_h^i + \mathbf{W}T_c^i, \tag{9.27}$$

Table 9.1: Heat exchanger properties

Parameter	Unit	Cold Fluid	Hot Fluid
U	$\frac{\text{J}}{\text{s}\cdot\text{m}^2\cdot^\circ\text{C}}$	2411.8	2411.8
A	m^2	48.4	48.4
$v_{c,\text{i}}$, $v_{h,\text{i}}$	m^3/s	0.04	0.10
$v_{c,\text{f}}$, $v_{h,\text{f}}$	m^3/s	0.02	0.06
ρ_c, ρ_h	$\text{kg}/\text{m}^3 \cdot 10^3$	3.50	3.72
c_{pc}, c_{ph}	$\frac{\text{J}}{\text{kg}\cdot^\circ\text{C}}$	481.8	499.0
V_c, V_h	$\text{m}^3 \cdot 10^{-2}$	15.8	57.8
$T^o_{c,\text{i}}$	$^\circ\text{C}$	9.3	–
$T^o_{c,\text{f}}$	$^\circ\text{C}$	25	–

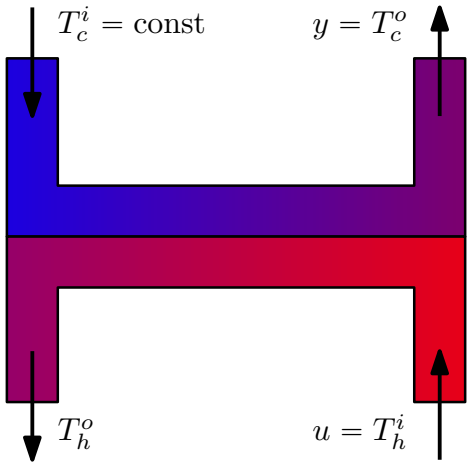


Figure 9.3: Diagram of a counter flow heat exchanger.

where

$$\mathbf{A}_\text{p} = \begin{bmatrix} -\frac{v_c(t)}{V_c} - \frac{UA}{c_{pc}\rho_c V_c} & \frac{UA}{c_{pc}\rho_c V_c} \\ \frac{UA}{c_{ph}\rho_h V_h} & -\frac{v_h(t)}{V_h} - \frac{UA}{c_{ph}\rho_h V_h} \end{bmatrix}, \tag{9.28}$$

$$\mathbf{B}_\text{p} = \begin{bmatrix} 0 \\ \frac{-v_h(t)}{V_h} \end{bmatrix}, \ \mathbf{W} = \begin{bmatrix} \frac{-v_c(t)}{V_c} \\ 0 \end{bmatrix}, \ \mathbf{T} = \begin{bmatrix} T_c^o(t) \\ T_h^o(t) \end{bmatrix}. \tag{9.29}$$

In this model, the input cold stream T_c^i is constant, and the outlet temperature T_c^o is to be regulated. The hot inlet temperature T_h^i is the control input, and the flow rates of the hot and cold fluids, $v_c(t)$ and $v_h(t)$ respectively, are time varying. In particular, cold and hot stream flow rates are $v_c(t) = \phi(t, v_{c,\text{i}}, v_{c,\text{f}})$ and $v_h(t) = \phi(t, v_{h,\text{i}}, v_{h,\text{f}})$,

where $t_\mathrm{f} = 20$ s, and where

$$\phi(t, x_\mathrm{i}, x_\mathrm{f}) = \begin{cases} x_\mathrm{i}, & t \leq 0, \\ x_\mathrm{i} + (x_\mathrm{f} - x_\mathrm{i}) \left[3 \left(\frac{t}{t_\mathrm{f}} \right)^2 - 2 \left(\frac{t}{t_\mathrm{f}} \right)^3 \right], & 0 \leq t \leq t_\mathrm{f}, \\ x_\mathrm{f}, & t_\mathrm{f} \leq t. \end{cases} \tag{9.30}$$

The desired cold output temperature is given by $T^o_{c,\mathrm{d}}(t) = \phi(t, T^o_{c,\mathrm{i}}, T^o_{c,\mathrm{f}})$ To realize a linear system with desired output $y = 0$, the system (9.27) is approximated by using the change of variables

$$\begin{aligned} \mathbf{x} &= \mathbf{T} + \boldsymbol{\eta} + \begin{bmatrix} \nu \\ \nu \end{bmatrix}, \\ u &= T^i_h + \nu, \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} (\mathbf{T} + \boldsymbol{\eta}) + \nu, \end{aligned}$$

where

$$\begin{aligned} \boldsymbol{\eta} &= (\mathbf{A}_\mathrm{p} + \mathbf{W} \begin{bmatrix} 1 & 0 \end{bmatrix})^{-1} \mathbf{W} (T^i_c - T^o_{c,\mathrm{f}}), \\ \nu &= \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{\eta} - T^o_{c,\mathrm{f}}. \end{aligned}$$

When fluid-flow is constant, $\dot{\mathbf{x}} = \dot{\mathbf{T}}$. The resulting state-space realization is determined using (9.30). The matrices $\mathbf{A}$ and $\mathbf{B}_2$ are

$$\mathbf{A} = \sum_{i=1}^{2} s_i(\boldsymbol{\sigma}, \mathbf{x}, t) \mathbf{A}_i, \quad \mathbf{B}_2 = \sum_{i=1}^{2} s_i(\boldsymbol{\sigma}, \mathbf{x}, t) \mathbf{B}_{2,i},$$

where $s_1 = \phi(t, 1, 0)$, and $s_2 = 1 - s_1 = \phi(t, 0, 1)$. The matrices $\mathbf{A}_1$ and $\mathbf{B}_{2,1}$ are defined by evaluating $\mathbf{A}_\mathrm{p}$ and $\mathbf{B}_\mathrm{p}$ in (9.28)-(9.29) at $v_{c,\mathrm{i}}$ and $v_{h,\mathrm{i}}$. The matrices $\mathbf{A}_2$ and $\mathbf{B}_{2,2}$ are defined by evaluating $\mathbf{A}_\mathrm{p}$ and $\mathbf{B}_\mathrm{p}$ in (9.28)-(9.29) at $v_{c,\mathrm{f}}$ and $v_{h,\mathrm{f}}$. The matrix $\mathbf{C}_2$ is constant, where $\mathbf{C}_2 = [1 \;\; 0]$.

9.4.2 Robust Control Design

The matrix physical parameters in $\mathbf{A}_\mathrm{p}$ may be uncertain. For example, U may be uncertain due to $CaCO_3$ formation [121, p. 31]. The uncertain $\bar{\mathbf{A}}$ matrix is modelled

Table 9.2: Conic bounds of the uncertain heat exchanger model.

Plant	Conic Max a	Conic Min r	$\|\boldsymbol{\Delta}\|_\infty$
$\boldsymbol{\mathcal{G}}$	cone$[-0.06, 98.9]$	cone$[-0.14, 0.38]$	0
$\boldsymbol{\mathcal{G}}_\Delta,\ \delta = 0.5$	cone$[-0.04, 97.4]$	cone$[-0.09, 0.24]$	0.057
$\boldsymbol{\mathcal{G}}_\Delta,\ \delta = -1$	cone$[-0.08, 99.4]$	cone$[-0.19, 0.52]$	0.23

as $\bar{\mathbf{A}} = \mathbf{A} + \mathbf{A}_\delta$, where

$$\mathbf{A}_\delta = \delta \begin{bmatrix} \frac{UA}{c_{pc}\rho_c V_c} & -\frac{UA}{c_{pc}\rho_c V_c} \\ -\frac{UA}{c_{ph}\rho_h V_h} & \frac{UA}{c_{ph}\rho_h V_h} \end{bmatrix}.$$

The system for robust control design is

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}_2\mathbf{u} + \mathbf{q}, \tag{9.31}$$

$$\mathbf{y} = \mathbf{C}_2\mathbf{x}, \tag{9.32}$$

$$\mathbf{p} = \mathbf{x}, \tag{9.33}$$

where the uncertainty block is given by $\mathbf{q} = \mathbf{A}_\delta\mathbf{p}$, and thus $\boldsymbol{\Delta} = \mathbf{A}_\delta$. The weighting matrices for robust control design using (9.31)-(9.33), are

$$\mathbf{B}_3 = \mathbf{1}, \quad \mathbf{C}_3 = \mathbf{1}, \quad \mathbf{D}_{23} = \mathbf{0}, \quad \mathbf{D}_{32} = \mathbf{0}. \tag{9.34}$$

Other weighting combinations are possible, such as by factoring $\mathbf{A}_\delta$, with factorization $\mathbf{B}_3\mathbf{C}_3 = \mathbf{A}_\delta$ or $\mathbf{B}_3\mathbf{C}_3 = \delta^{-1}\mathbf{A}_\delta$, and then by setting $\boldsymbol{\Delta} = \mathbf{1}$ or $\boldsymbol{\Delta} = \delta\mathbf{1}$. In some cases, this may provide a numerically simpler controller synthesis, but in the heat exchanger example, no significant controller improvements were found. Table 9.2 shows the heat exchanger's conic sectors and $||\boldsymbol{\Delta}||_\infty$ for various values of δ. A value of $\delta = 0.5$ is equivalent to halving the heat exchanger's area, such as during $CaCO_3$ buildup. A value of $\delta = -1$ doubles the heat exchanger's area.

Using the design method for an LPV controller from [102], the input must be filtered to obtain a constant $\mathbf{B}_2$ matrix. A first-order filter with a cutoff frequency of 2 rad/s is implemented on the input of the system, which adds system states. Designing the polytopic LPV controller using the weights from (9.34) results in $||\boldsymbol{\mathcal{G}}_{\text{cl}}||_\infty = 16.67$. Recall that for robust closed-loop stability via the Small Gain Theorem, $||\boldsymbol{\mathcal{G}}_{\text{cl}}||_\infty < 1/||\boldsymbol{\Delta}||_\infty$. This means that stability can be guaranteed via the Small Gain Theorem for $\delta = 0.5$, but cannot be guaranteed for $\delta = -1$.

The conic bounds of the heat exchanger are found using Theorem 9.2. The Nyquist

plots of $\boldsymbol{\mathcal{G}}$ at the vertices of the polytope are shown in Fig. 9.4, as well as a circle that denotes the conic bounds. Notice that the plant lies well within the conic bounds determined by Theorem 9.2. The Nyquist plot of the vertices is shown in Fig. 9.5 when the conic radius is minimized. In this plot, the black dashed circle is much tighter than when maximizing a. This would seem like tighter conic bounds and thus improved controller performance, but the more negative a leads to a decreased b_c, and thus yields a controller with more conservative gain and inferior performance. To robustly stabilize the plant, the largest cone that captures both the nominal and perturbed plant must be considered, thus cone for controller design is cone$[-0.19, 0.52]$ for minimizing r and cone$[-0.08, 99.4]$ for maximizing a.

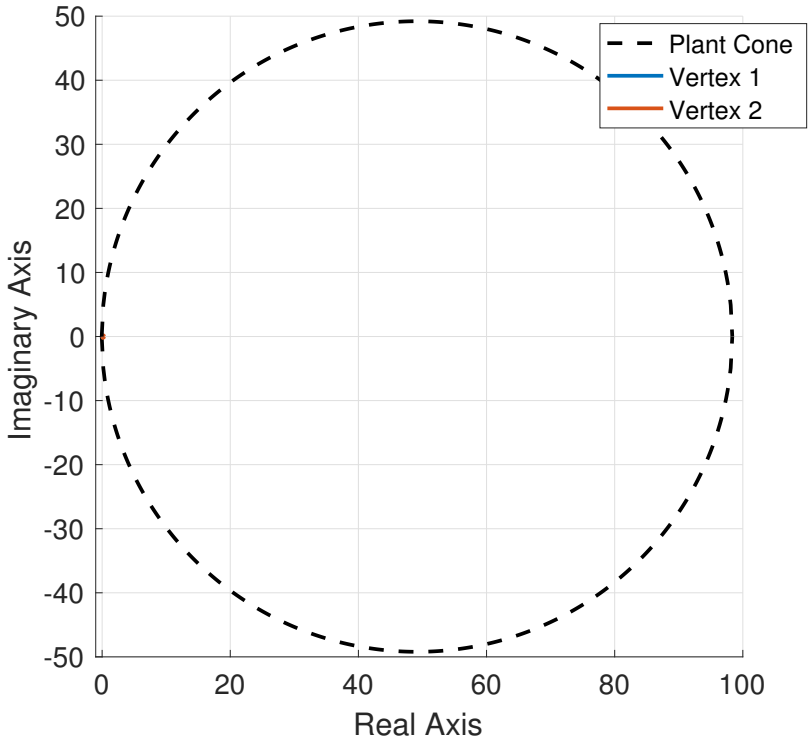
The weights (9.34) are used to design the $\mathcal{H}_\infty$ controllers at each vertex. These controllers are used for controller synthesis, as discussed in Section 9.3. To analyze the utility of forcing the $\mathcal{H}_\infty$ controllers at each vertex to satisfy Theorem 7.10, the $\mathcal{H}_\infty$ controllers at each vertex are linearly interpolated as is to form an $\mathcal{H}_\infty$ gain-scheduled controller. This controller has no stability or performance guarantees when in closed-loop with the heat exchanger.

The Nyquist plots of the $\mathcal{H}_\infty$ controller at each vertex and the conic controller at each vertex, synthesized using maximum plant a, are shown in Fig. 9.6. The gain of the $\mathcal{H}_\infty$ controllers is much larger than the gain of the conic controllers. When synthesizing the conic controller, the gain of the controller is decreased so that the resulting controller fits in the b_c bound. The gain at both vertices are not the same. The gain at vertex 1 of the $\mathcal{H}_\infty$ controller is smaller than at vertex 2, and this is reversed for the conic case. This may indicate that the controller is more effective at one vertex compared to the other vertex, but this has not been tested due to the way the temperature is commanded as the fluid flow rate changes.

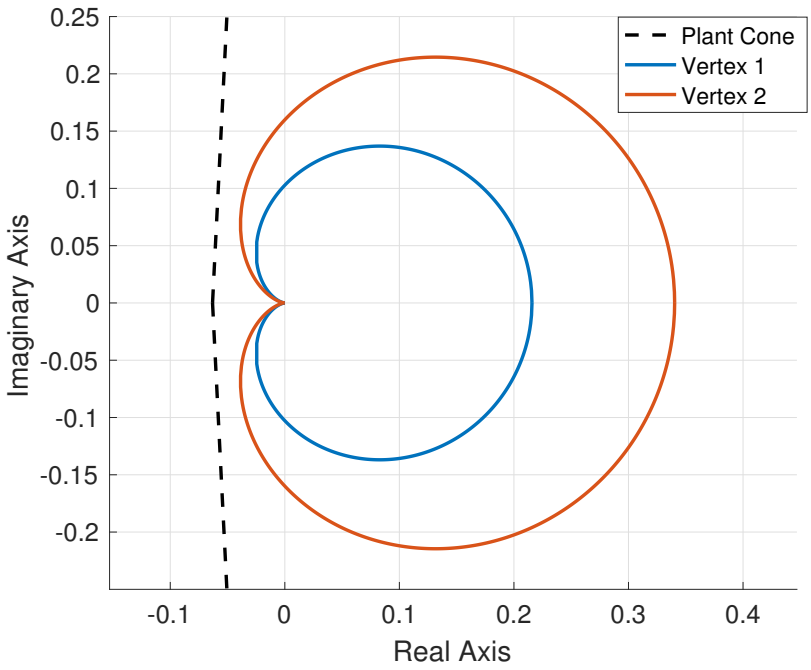
Fig. 9.7 shows the conic bounds for the controller synthesized with maximum plant a and minimum plant r. The b_c bound, which is related to the upper limit on controller gain, is approximately 2.5 times larger when using the bounds determined by maximum a than by minimum r. It is expected that the controller with the larger design freedom exhibits superior closed-loop performance, that is the RMS error of T_c^o should be less for the max a controller than the min r controller.

9.4.3 Numerical Results

Numerical results are shown in Fig. 9.8, and the case $\delta = -1$ is omitted for brevity. The root-mean square (RMS) values of the error of the cold outlet temperature are shown in Table 9.3. At $\delta = 0.5$, the area for heat exchange halves. The performance



(a) Plant conic boundary and plant Nyquist plot.



(b) Zoomed-in plot of plant Nyquist plot.

Figure 9.4: Nyquist plot of plant and conic bounds.

of all controllers improve with $\delta = 0.5$, which implies that not all uncertainty degrades performance. When $\delta = -1$, the opposite is true and the performance of all controllers

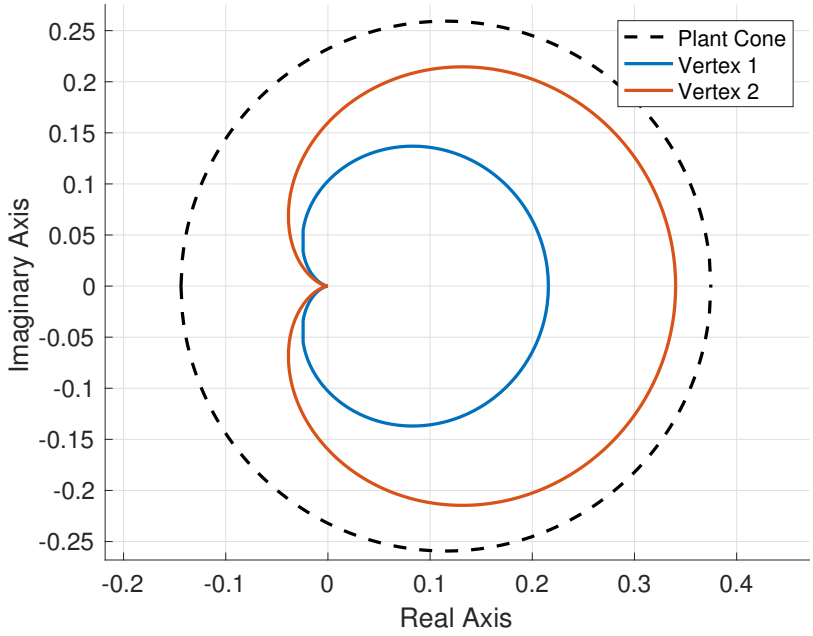


Figure 9.5: Plant conic boundary and plant Nyquist plot with minimum conic radius.

deteriorates.

For $\delta = 0$ and $\delta = 0.5$, the $\mathcal{H}_\infty$ controller actually performs better than the LPV controller. This may be explained by the fact that for polytopic controller design, a common Lyapunov matrix is required, which is a source of conservatism. The interpolated $\mathcal{H}_\infty$ is synthesized at the vertices, and even though it has no guarantee to stabilize the closed-loop system, it still worked. However, with a different set of weighting matrices, in simulation, the interpolated $\mathcal{H}_\infty$ controller lead to poor closed-loop performance, and even closed-loop instability.

This numerical example highlights the benefits of conic-sector-based control. The largest conic bounds for the three plants are chosen for conic design, and the conic controller is guaranteed to stabilize the plant in each case. In addition, the sensitivity to plant uncertainty is minimal. The standard deviations of the RMS temperature error of each plant for each controller are given in Table 9.3. The standard deviation between the RMS errors is seven times less for the conic controller design than with the $\mathcal{H}_\infty$-based LPV design. However, this is partly expected because conic controllers exhibit a similar level of robustness for linear control design [18].

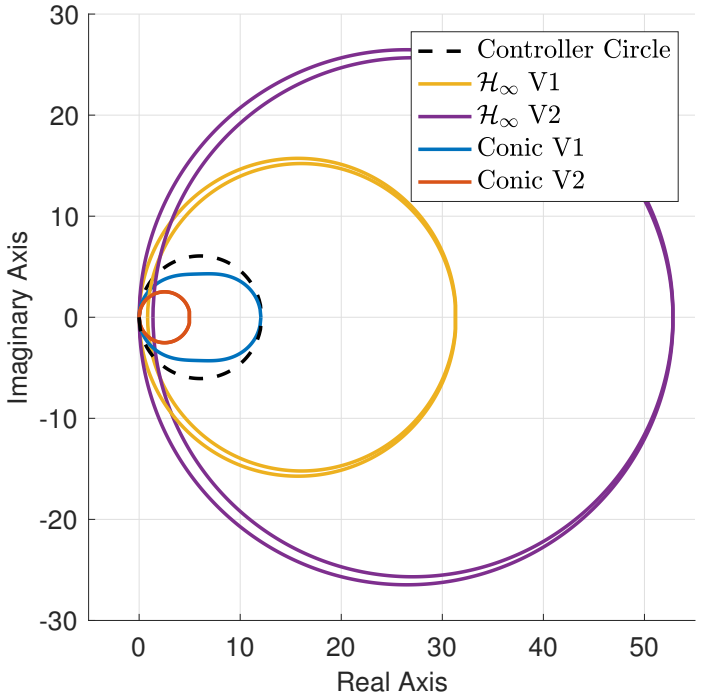
9.5 Closing Remarks

This chapter provides LMI conditions to determine conic bounds for interior-conic, polytopic systems. This chapter also provides a method that synthesizes a polytopic

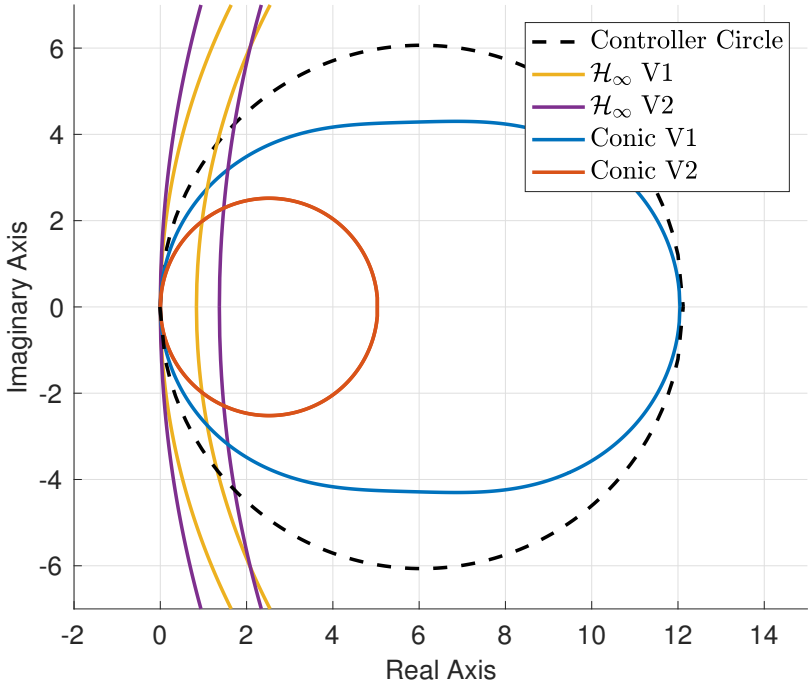
Table 9.3: RMS error of $T_c^o - T_{c,\mathrm{d}}^o$, $^\circ C$

Plant	$\mathcal{H}_\infty$	Conic Max a	Conic Min r	LPV
$\boldsymbol{\mathcal{G}}$	1.13	0.612	0.912	1.19
$\boldsymbol{\mathcal{G}}_\Delta,\ \delta = 0.5$	0.782	0.606	0.850	0.832
$\boldsymbol{\mathcal{G}}_\Delta,\ \delta = -1$	1.72	0.721	1.047	1.68
std. dev.	0.477	0.065	0.100	0.424

controller subject to conic bounds. A common Lyapunov matrix is required for both these methods, which is a source of conservatism. This may provide unnecessarily large conic bounds for the plant, which would lead to more conservatives controllers. Additionally, even without conservative controller conic bounds, a common Lyapunov matrix may result in a conservative controller anyway. That being said, the effects and methods to mitigate the common Lyapunov matrix is an area of future research.



(a) Zoomed-out plot.



(b) Zoomed-in plot.

Figure 9.6: Nyquist plot of $\mathcal{H}_\infty$ controller and the conic controller at each vertex of the polytope. The controller circle plots the conic bounds of the controller.

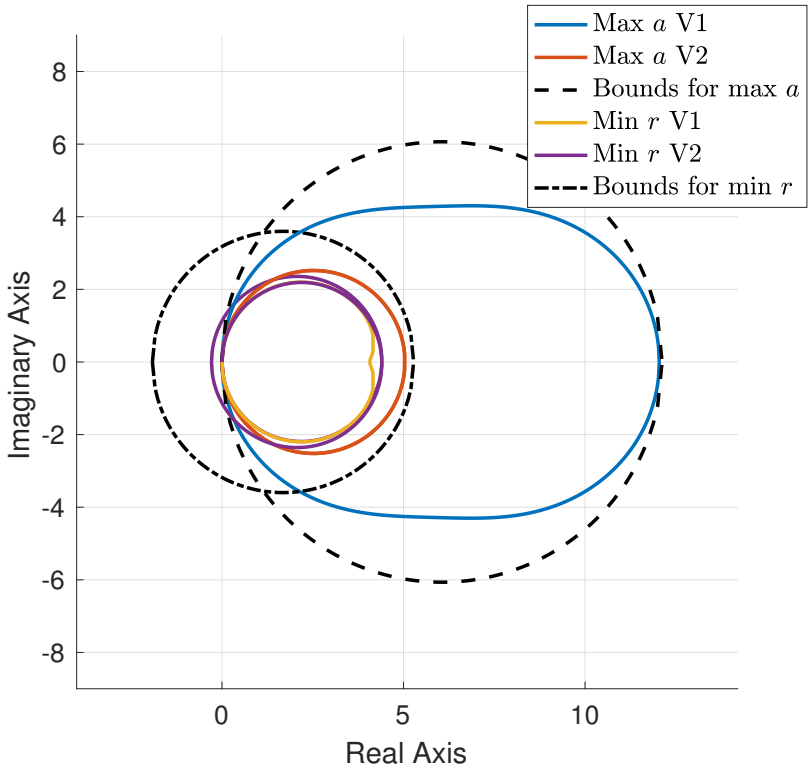
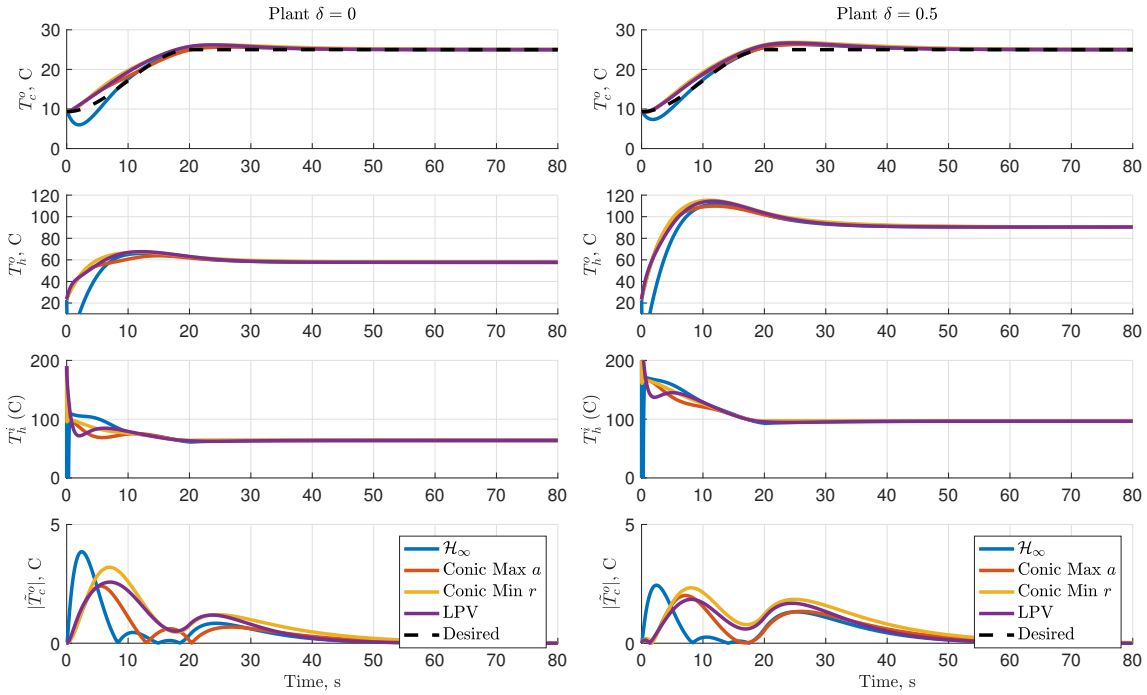


Figure 9.7: Nyquist plot of controllers at vertices of polytope synthesized using maximum plant a and minimum plant r conic bounds.



(a) Nominal plant with $\delta = 0$.

(b) Modified plant with $\delta = 0.5$.

Figure 9.8: Simulation results to track T_c^o, with input T_h^i of the heat exchanger.

Part IV

Conclusion

Chapter 10

Closing Remarks and Future Work

This dissertation considers guidance and control of nonlinear systems. First, guidance for position and attitude is discussed. The guidance laws extremize objective functions using SDPs in Chapter 4 and gradient-based optimization methods for extremum-seeking in Chapters 5 and 6. Gradient-based extremum-seeking guidance necessitates estimation of the gradient, which is enabled by a Kalman filter. The Kalman filter is modified so that it can handle any type of linear or LMI constraint, and is able to filter measurement noise to better estimate the gradient of the objective function. Although the presented extremum-seeking guidance algorithms are effective, they are gradient-based methods, which suffer from local maxima and minima when the optimization space is nonconvex. Extremum-seeking guidance for nonconvex problems is an area for future research.

A particular contribution that is emphasized in this dissertation is the use of the DCM, an element of $SO(3)$, for attitude parameterization. The DCM parameterizes attitude globally and uniquely, but familiar tools such as matrix addition and subtraction cannot be used with the DCM. This dissertation presents extremum-seeking for attitude using the DCM directly and, by doing so, effectively overcomes these challenges.

Guidance in Part II is demonstrated on the formation flight of two aircraft and a spacecraft with three reaction wheels. Problems that consider swarms of aircraft or spacecraft that work in unison to accomplish a common goal may require extremum-seeking guidance to be extended to $\mathbb{R}^n$, $\mathbb{R}^{n\times n}$, $SO(n)$, or $SE(n)$. The matrix Lie group $SE(n)$ is not discussed in this dissertation. An element of $SE(3)$ represents a body's pose, that is its position and attitude relative to a particular point and reference frame. The set $SE(n)$ is the generalization of $SE(3)$. An extremum-seeking guidance law on $SE(3)$ can simultaneously provide guidance for position and attitude, rather

than separately, as is the case in this dissertation. In the short term, extremum-seeking guidance on $SE(3)$ can be derived by using Chapters 5 and 6 as a starting point.

Part III moves from guidance to control. LPV control based on the Passivity and Conic Sector Theorems finds its origins from gain-scheduling of SPR subcontrollers [96, 95]. Initial attempts at gain-scheduling resulted in ISP and VSP controllers with a constant dynamics matrix, where the poles of the controller were invariant [97, 98]. Only the $\mathbf{B}_c$ and $\mathbf{C}_c$ matrices of the controller changed as a function of scheduling signal. This dissertation extends [97, 98], which resulted in [3], and is detailed in Chapter 8. Chapter 8 presents a control architecture that is VSP and allows all controller state-space matrices to be parameter dependent. This new controller is much closer in structure to a traditional LPV system than previous passivity-based gain-scheduled control attempts. However, stability results from the Passivity Theorem are based on phase. Thus, unlike traditional LPV controllers where stability results depend on the Small Gain Theorem, gain of uncertainty is not needed in the passivity framework. The VSP controller is able to stabilize any passive plant where uncertainty does not violate the passive input-output map. Robustness of the controller is inherent due to the input-output theory that is used.

Since passive systems may exhibit passivity violations, the Conic Sector Theorem can be used as an alternative to the Passivity Theorem. The Conic Sector Theorem does not require plants, uncertainty, or controllers to be passive. Furthermore, uncertainty that results in a large b bound on the plant may not result in a conservative controller compared to controllers designed with the Small Gain Theorem. For an explanation, recall that $b_c < -1/a$, and the b and b_c bounds are related to plant and controller gain respectively. The b_c bound is not related to the b bound via the Conic Sector Theorem. Using the Small Gain Theorem, a large b bound would mean an equally large a bound (since $a = -b$ in this case), which would dramatically decrease controller gain. In this case, $b_c = -1/a = 1/b$, indicating b and b_c are related. Conic-sector-based control circumvents this problem. That being said, this dissertation does not provide a truly direct comparison between these two design methods. For conic-sector-based robust control, the controller is forced to satisfy conic bounds to ensure closed-loop stability with $\boldsymbol{\mathcal{G}}_\Delta$. For traditional robust control, the controller is designed such that $\boldsymbol{\mathcal{G}}_{cl}$ satisfies the small gain condition with respect to the uncertainty block, $\boldsymbol{\Delta}$. In the future, it is of interest to design $\boldsymbol{\mathcal{G}}_{cl}$ to satisfy conic bounds, or to have $\boldsymbol{\mathcal{G}}_c$ satisfy the small gain condition with respect to $\boldsymbol{\mathcal{G}}_\Delta$.

In Chapter 8, nonlinear systems can be shown to be passive, and thus VSP con-

trollers can be used to stabilize the passive system even if an affine plant approximation is used for controller synthesis. However, for a general nonlinear system, there is no systematic method to determine conic bounds for an input-output map. As such, control techniques from Chapter 9 may be limited to polytopic systems because Chapter 9, via Theorem 9.2, introduces a method to determine bounds that is limited to polytopic systems. However, Theorem 9.2 may also be used for an affine system by transforming the affine system to a polytopic system. This transformation is possible if bounds on the parameters or scheduling signals are known, which is often the case [22].

In Part III, controller synthesis requires a common Lyapunov matrix in their formulations, which is a source of conservatism. In the traditional LPV framework, a parameter dependent Lyapunov matrix is possible, but controller synthesis often results in gridding. It is possible to extend the presented framework to allow gridding, but this injects numerical complexity to synthesis problems. The use of dilated LMIs may be useful to reduce conservatism from a common Lyapunov matrix, but only preliminary work has been conducted using dilation, and is thus not included in this dissertation.

In summary, although there are still challenges associated with conic-sector-based control, this new approach to LPV control has many advantages. Conic-sector-based control does not have the drawbacks associated with the strict conditions of passivity-based control or the conservativeness associated with small-gain-based control. Chapter 9 further reveals that controllers based on the Conic Sector Theorem are not very sensitive to plant uncertainty and can provide superior closed-loop performance compared to traditional LPV controllers.

Appendices

Appendix A

Polytopic Conic Design with $\mathbf{C}_c$ Matrix

The alternative to conic design with the $\mathbf{B}_{c,i}$ matrix is design with $\mathbf{C}_{c,i}$. The resulting state space matrices from the $\mathcal{H}_\infty$ synthesis are

$$\begin{bmatrix} \mathbf{A}_{c,i} & \mathbf{L}_i \\ \mathbf{K}_i & \mathbf{0} \end{bmatrix}.$$

The matrix $\mathbf{B}_{c,i}$ is set to $\mathbf{B}_{c,i} = \mathbf{L}_i$. The $\mathbf{C}_{c,i}$ matrix is determined by minimizing the $\mathcal{H}_2$-norm of the difference between the $\mathcal{H}_\infty$ controller and a controller that satisfies (9.10). Specifically, this optimization problem is given by minimizing

$$\mathcal{J}(\mathbf{P}_c, \mathbf{C}_{c,1}, \ldots, \mathbf{C}_{c,N}) = \sum_{i=1}^{N} \operatorname{tr}(\mathbf{C}_{c,i} - \mathbf{K}_i)\mathbf{V}_i(\mathbf{C}_{c,i} - \mathbf{K}_i)^\mathsf{T}, \tag{A.1}$$

subject to

$$\begin{bmatrix} \mathbf{A}_{c,i}^\mathsf{T}\mathbf{P}_c + \mathbf{P}_c\mathbf{A}_{c,i} & \mathbf{P}_c\mathbf{B}_{c,i} & \mathbf{C}_{c,i} \\ \star & -\frac{(a_c - b_c)^2}{4b_c}\mathbf{1} & -\frac{a_c + b_c}{2}\mathbf{1} \\ \star & \star & -b_c\mathbf{1} \end{bmatrix} < 0, \tag{A.2}$$

for $i = 1, \ldots, N$, where (A.2) is a modification of (9.10). The matrix $\mathbf{V}_i$ is the controllability Grammarian that satisfies

$$\mathbf{V}_i\mathbf{A}_{c,i}^\mathsf{T} + \mathbf{A}_{c,i}\mathbf{V}_i + \mathbf{B}_{c,i}\mathbf{B}_{c,i}^\mathsf{T} = \mathbf{0}. \tag{A.3}$$

The objective function (A.1) is chosen because when the difference between $\mathbf{C}_{c_i}$ and $\mathbf{K}_i$ subject to the weight $\mathbf{W}_i$ is minimized, the difference between the $\mathcal{H}_\infty$ controller and the controller that satisfies (9.10) in an $\mathcal{H}_2$ sense.

The quadratic function (A.1) can be transformed to a linear function and an LMI constraint given by

$$\mathcal{J}'(\nu, \mathbf{Z}, \mathbf{P}_{\mathrm{c}}, \mathbf{C}_{\mathrm{c},1}, \ldots, \mathbf{C}_{\mathrm{c},N}) = \nu, \tag{A.4}$$

constrained by

$$\nu \geq \mathrm{tr}(\mathbf{Z}), \tag{A.5}$$

and

$$\begin{bmatrix} \mathbf{Z} & (\mathbf{C}_{\mathrm{c},1} - \mathbf{K}_1) & \cdots & (\mathbf{C}_{\mathrm{c},N} - \mathbf{K}_N) \\ \star & \mathbf{V}_1^{-1} & \mathbf{0} & \mathbf{0} \\ \vdots & \star & \ddots & \vdots \\ \star & \star & \star & \mathbf{V}_N^{-1} \end{bmatrix} \geq 0. \tag{A.6}$$

Determining $\mathbf{C}_{\mathrm{c},i}$ is summarized by Problem A.1.

Problem A.1. The problem to determine $\mathbf{B}_{\mathrm{c},i}$ for $i = 1, \ldots, N$ is given by minimizing (A.4), subject to (A.5), (A.6), and (A.2). □

Appendix B

Miscellaneous Derivations

B.1 Newton-Euler Derivation of Spacecraft with Three Reaction Wheels

In this section, only the rotational equation of motion for the spacecraft is derived. The Newton-Euler equation of motion for a rigid-body $\mathcal{B}$ is given by

$$\underrightarrow{h}^{\mathcal{B}c/a\,\bullet\,a} = \underrightarrow{m}^{\mathcal{B}c}, \tag{B.1}$$

where $\underrightarrow{m}^{\mathcal{B}_c}$ is the total moment acting on the body relative to point c, the centre of mass, and $\underrightarrow{h}^{\mathcal{B}c/a}$ is the angular momentum of body $\mathcal{B}$ relative to point c with respect to $\mathcal{F}_a$ [35]. The total angular momentum is

$$\underrightarrow{h}^{\mathcal{B}c/a} = \underrightarrow{h}^{\mathcal{P}c/a} + \sum_{k=1}^{N} \underrightarrow{h}^{\mathcal{W}_k c/a}, \tag{B.2}$$

where the angular momentum of a reaction wheel is

$$\underrightarrow{h}^{\mathcal{W}_k c/a} = \underrightarrow{h}^{\mathcal{W}_k d_k/a} + \underrightarrow{r}^{d_k c} \times m_{\mathcal{W}_k} \cancelto{\underrightarrow{0}}{\underrightarrow{r}^{d_k c\,\bullet\,a}} = \underrightarrow{J}^{\mathcal{W}_k d_k} \cdot \underrightarrow{\omega}^{w_k a} = \underrightarrow{J}^{\mathcal{W}_k d_k} \cdot \left(\underrightarrow{\omega}^{w_k p} + \underrightarrow{\omega}^{pa} \right), \tag{B.3}$$

and the angular momentum of the spacecraft bus is $\underrightarrow{h}^{\mathcal{P}c/a} = \underrightarrow{J}^{\mathcal{P}c} \cdot \underrightarrow{\omega}^{pa}$. Substituting $\underrightarrow{h}^{\mathcal{P}c/a}$ and Eq. (B.3) into Eq. (B.2) yields

$$\underrightarrow{h}^{\mathcal{B}c/a} = \underrightarrow{J}^{\mathcal{P}c} \cdot \underrightarrow{\omega}^{pa} + \sum_{k=1}^{N} \underrightarrow{J}^{\mathcal{W}_k d_k} \cdot \left(\underrightarrow{\omega}^{w_k p} + \underrightarrow{\omega}^{pa} \right) = \underrightarrow{J}^{\mathcal{B}c} \cdot \underrightarrow{\omega}^{pa} + \sum_{k=1}^{N} \underrightarrow{J}^{\mathcal{W}_k d_k} \cdot \underrightarrow{\omega}^{w_k p}. \tag{B.4}$$

Assume $\underrightarrow{\omega}^{w_k p}$ is a along the spin axis of $\underrightarrow{J}^{\mathcal{W}_k d_k}$ and that $\underrightarrow{J}^{\mathcal{B}c}$ is constant. Taking the time derivative of Eq. (B.4) with respect to $\mathcal{F}_a$ and substituting the result into Eq. (B.1) yields

$$\begin{aligned}
\underrightarrow{m}^{\mathcal{B}c} &= \underrightarrow{J}^{\mathcal{B}c} \cdot \underrightarrow{\omega}^{pa^{\bullet_a}} + \sum_{k=1}^{N} \underrightarrow{J}^{\mathcal{W}_k d_k} \cdot \underrightarrow{\omega}^{w_k p^{\bullet_a}} \\
&= \underrightarrow{J}^{\mathcal{B}c} \cdot \underrightarrow{\omega}^{pa^{\bullet_p}} + \underrightarrow{\omega}^{pa} \times (\underrightarrow{J}^{\mathcal{B}c} \cdot \underrightarrow{\omega}^{pa}) + \sum_{k=1}^{N} \left(\underrightarrow{\omega}^{pa} \times \underrightarrow{J}^{\mathcal{W}_k d_k} \cdot \underrightarrow{\omega}^{w_k p} + \underrightarrow{J}^{\mathcal{W}_k d_k} \cdot \underrightarrow{\omega}^{w_k p^{\bullet_p}} \right) \\
&= \underrightarrow{J}^{\mathcal{B}c} \cdot \underrightarrow{\omega}^{pa^{\bullet_p}} + \underrightarrow{\omega}^{pa} \times \left(\underrightarrow{J}^{\mathcal{B}c} \cdot \underrightarrow{\omega}^{pa} + \sum_{k=1}^{N} \underrightarrow{J}^{\mathcal{W}_k d_k} \cdot \underrightarrow{\omega}^{w_k p} \right) + \sum_{k=1}^{N} \underrightarrow{J}^{\mathcal{W}_k d_k} \cdot \underrightarrow{\omega}^{w_k p^{\bullet_p}}.
\end{aligned} \tag{B.5}$$

Equation (B.5) is the equation of motion of the spacecraft derived via a Newton-Euler approach. To resolve Eq. (B.5) in $\mathcal{F}_p$, the relations

$$\begin{aligned}
&\underrightarrow{J}^{\mathcal{B}c} = \underrightarrow{\boldsymbol{\mathcal{F}}}_p^{\mathsf{T}} \mathbf{J}_p^{\mathcal{B}c} \underrightarrow{\boldsymbol{\mathcal{F}}}_p, \quad \underrightarrow{J}^{\mathcal{W}_k d_k} = \underrightarrow{\boldsymbol{\mathcal{F}}}_{w_k}^{\mathsf{T}} \mathbf{J}_{w_k}^{\mathcal{W}_k d_k} \underrightarrow{\boldsymbol{\mathcal{F}}}_{w_k}, \quad \underrightarrow{\omega}^{pa} = \underrightarrow{\boldsymbol{\mathcal{F}}}_p^{\mathsf{T}} \boldsymbol{\omega}_p^{pa}, \quad \underrightarrow{\omega}^{pa^{\bullet_p}} = \underrightarrow{\boldsymbol{\mathcal{F}}}_p^{\mathsf{T}} \dot{\boldsymbol{\omega}}_p^{pa}, \\
&\underrightarrow{\omega}^{w_k p} = \underrightarrow{\boldsymbol{\mathcal{F}}}_{w_k}^{\mathsf{T}} \boldsymbol{\omega}_{w_k}^{w_k p}, \quad \underrightarrow{\omega}^{w_k p^{\bullet_p}} = \underrightarrow{\boldsymbol{\mathcal{F}}}_{w_k}^{\mathsf{T}} \dot{\boldsymbol{\omega}}_{w_k}^{w_k p}, \qquad \underrightarrow{m}^{\mathcal{B}_c} = \underrightarrow{\boldsymbol{\mathcal{F}}}_p^{\mathsf{T}} \boldsymbol{\tau}_p^{\text{ext}},
\end{aligned}$$

are substituted into Eq. (B.5) and the result is left-multiplied by $\underrightarrow{\boldsymbol{\mathcal{F}}}_p$ yielding

$$\mathbf{J}_p^{\mathcal{B}c} \dot{\boldsymbol{\omega}}_p^{pa} + \boldsymbol{\omega}_p^{pa\times} \left(\mathbf{J}_p^{\mathcal{B}c} \boldsymbol{\omega}_p^{pa} + \sum_{k=1}^{N} \mathbf{C}_{w_k p}^{\mathsf{T}} \mathbf{J}_{w_k}^{\mathcal{W}_k d_k} \boldsymbol{\omega}_{w_k}^{w_k p} \right) + \sum_{k=1}^{N} \mathbf{C}_{w_k p}^{\mathsf{T}} \mathbf{J}_{w_k}^{\mathcal{W}_k d_k} \dot{\boldsymbol{\omega}}_{w_k}^{w_k p} = \boldsymbol{\tau}_p^{\text{ext}}. \tag{B.6}$$

B.2 Proof of Proposition 6.3.

The derivatives of a function $f(\mathbf{X}) : \mathbb{R}^{n\times n} \to \mathbb{R}$ and a function $\mathbf{X}(y) : \mathbb{R} \to \mathbb{R}^{n\times n}$ are given by

$$\mathbf{A} = \frac{\mathrm{d}f}{\mathrm{d}\mathbf{X}}, \quad \mathbf{B} = \frac{\mathrm{d}\mathbf{X}}{\mathrm{d}y}, \tag{B.7}$$

where the elements of $\mathbf{A}$ and $\mathbf{B}$ are given by

$$a_{ij} = \frac{\mathrm{d}f(\mathbf{X})}{\mathrm{d}x_{ji}}, \quad b_{ji} = \frac{\mathrm{d}x_{ji}(y)}{\mathrm{d}y}, \tag{B.8}$$

where x_{ji} are elements of $\mathbf{X}$ (notice the order of the indices). In addition, the chain rule for the function $f(\mathbf{X}(y))$ is defined as

$$\frac{\mathrm{d}f}{\mathrm{d}y} = \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\mathrm{d}f(\mathbf{X})}{\mathrm{d}x_{ji}} \frac{\mathrm{d}x_{ji}(y)}{\mathrm{d}y}. \tag{B.9}$$

Since the trace of two matrices $\mathbf{A}$ and $\mathbf{B}$ is $\mathrm{tr}(\mathbf{AB}) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{ji}$, using (B.7)-(B.9), yields

$$\begin{aligned}
\frac{\mathrm{d}f}{\mathrm{d}y} &= \sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\mathrm{d}f(\mathbf{X})}{\mathrm{d}x_{ji}} \frac{\mathrm{d}x_{ji}(y)}{\mathrm{d}y} \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{ji} \\
&= \mathrm{tr}(\mathbf{AB}) \\
&= \mathrm{tr}\left[\left(\frac{\mathrm{d}f(\mathbf{X})}{\mathrm{d}\mathbf{X}}\right)\left(\frac{\mathrm{d}\mathbf{X}(y)}{\mathrm{d}y}\right)\right].
\end{aligned}$$

B.3 Kalman Filter Derivation

Consider the objective function given by

$$\mathcal{J}_k(\mathbf{K}_k) = \frac{1}{2}\left[(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{b}}_k)^{\mathsf{T}}\mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{b}}_k) + (\hat{\mathbf{b}}_k - \hat{\mathbf{b}}_k^-)^{\mathsf{T}}(\mathbf{P}_k^-)^{-1}(\hat{\mathbf{b}}_k - \hat{\mathbf{b}}_k^-)\right],$$

which is inspired by the recursive least-squares [83, p. 205–207] and maximum likelihood approach to deriving the Kalman filter [84]. This objective function weighs the term $\mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{b}}_k$ with respect to the measurement covariance $\mathbf{R}_k$, and weighs the residual between the prior and posterior estimates with respect to the prior covariance $\mathbf{P}_k^-$.

Using the correction of the state estimate $\hat{\mathbf{b}}_k = \hat{\mathbf{b}}_k^- + \mathbf{K}_k\mathbf{r}_k$, where $\mathbf{r}_k = \mathbf{y}_k - \mathbf{H}_k\hat{\mathbf{b}}_k^-$

is the innovation term, the objective function is

$$\begin{aligned}\mathcal{J}_k(\mathbf{K}_k) &= \frac{1}{2}\big[(\mathbf{y}_k - \mathbf{H}_k(\hat{\mathbf{b}}_k^- + \mathbf{K}_k\mathbf{r}_k))^\mathsf{T}\mathbf{R}_k^{-1}(\mathbf{y}_k - \mathbf{H}_k(\hat{\mathbf{b}}_k^- + \mathbf{K}_k\mathbf{r}_k)) \\ &\quad + ((\hat{\mathbf{b}}_k^- + \mathbf{K}_k\mathbf{r}_k) - \hat{\mathbf{b}}_k^-)^\mathsf{T}(\mathbf{P}_k^-)^{-1}((\hat{\mathbf{b}}_k^- + \mathbf{K}_k\mathbf{r}_k) - \hat{\mathbf{b}}_k^-)\big] \\ &= \frac{1}{2}\big[(\mathbf{r}_k - \mathbf{H}_k\mathbf{K}_k\mathbf{r}_k)^\mathsf{T}\mathbf{R}_k^{-1}(\mathbf{r}_k - \mathbf{H}_k\mathbf{K}_k\mathbf{r}_k) + (\mathbf{K}_k\mathbf{r}_k)^\mathsf{T}(\mathbf{P}_k^-)^{-1}(\mathbf{K}_k\mathbf{r}_k)\big] \\ &= \frac{1}{2}\mathbf{r}_k^\mathsf{T}\big[(\mathbf{1} - \mathbf{H}_k\mathbf{K}_k)^\mathsf{T}\mathbf{R}_k^{-1}(\mathbf{1} - \mathbf{H}_k\mathbf{K}_k) + \mathbf{K}_k^\mathsf{T}(\mathbf{P}_k^-)^{-1}\mathbf{K}_k\big]\mathbf{r}_k. \end{aligned} \tag{B.10}$$

The objective function (B.10) can be rewritten as

$$\hat{\mathcal{J}}_k(\mathbf{K}_k) = \mathrm{tr}\big[(\mathbf{1} - \mathbf{H}_k\mathbf{K}_k)^\mathsf{T}\mathbf{R}_k^{-1}(\mathbf{1} - \mathbf{H}_k\mathbf{K}_k) + \mathbf{K}_k^\mathsf{T}(\mathbf{P}_k^-)^{-1}\mathbf{K}_k\big],$$

since we must find the minimum of (B.10) for any $\mathbf{r}_k$. Taking the derivative of (B.10) with respect to $\mathbf{K}_k$ and setting the result to zero yields

$$\frac{\partial\hat{\mathcal{J}}_k(\mathbf{K}_k)}{\partial\mathbf{K}_k} = -(\mathbf{1} - \mathbf{H}_k\mathbf{K}_k)^\mathsf{T}\mathbf{R}_k^{-1}\mathbf{H}_k + \mathbf{K}_k^\mathsf{T}(\mathbf{P}_k^-)^{-1} = \mathbf{0}.$$

Simplifying the above results in

$$\begin{aligned} -(\mathbf{1} - \mathbf{H}_k\mathbf{K}_k)^\mathsf{T}\mathbf{R}_k^{-1}\mathbf{H}_k + \mathbf{K}_k^\mathsf{T}(\mathbf{P}_k^-)^{-1} &= \mathbf{0} \\ -\mathbf{H}_k^\mathsf{T}\mathbf{R}_k^{-1}(\mathbf{1} - \mathbf{H}_k\mathbf{K}_k) + (\mathbf{P}_k^-)^{-1}\mathbf{K}_k &= \mathbf{0} \\ -\mathbf{H}_k^\mathsf{T}\mathbf{R}_k^{-1} + \mathbf{H}_k^\mathsf{T}\mathbf{R}_k^{-1}\mathbf{H}_k\mathbf{K}_k + (\mathbf{P}_k^-)^{-1}\mathbf{K}_k &= \mathbf{0} \\ ((\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^\mathsf{T}\mathbf{R}_k^{-1}\mathbf{H}_k)\mathbf{K}_k &= \mathbf{H}_k^\mathsf{T}\mathbf{R}_k^{-1} \\ \mathbf{K}_k &= ((\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^\mathsf{T}\mathbf{R}_k^{-1}\mathbf{H}_k)^{-1}\mathbf{H}_k^\mathsf{T}\mathbf{R}_k^{-1}. \end{aligned} \tag{B.11}$$

Applying the identity given by (13) in [122] to (B.11) directly yields

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^\mathsf{T}(\mathbf{R}_k + \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^\mathsf{T})^{-1}. \tag{B.12}$$

B.4 Proof of Lemma 9.1

This Appendix presents the proof of Lemma 9.1. Each element of $\mathbf{y}$ is given by

$$y_j(t) = \sum_{i=1}^{N} s_i(\boldsymbol{\sigma}, \mathbf{x}, t)y_{ij}(t). \tag{B.13}$$

The square of (B.13) results in

$$\begin{aligned} y_j^2(t) &= \left| \sum_{i=1}^{N} s_i(\boldsymbol{\sigma}, \mathbf{x}, t) y_{ij}(t) \right|^2, \\ &= \left| \sum_{i=1}^{N} \sqrt{s_i(\boldsymbol{\sigma}, \mathbf{x}, t)} \left(\sqrt{s_i(\boldsymbol{\sigma}, \mathbf{x}, t)} y_{ij}(t) \right) \right|^2. \end{aligned}$$

Using the Cauchy-Schwartz Inequality yields

$$\begin{aligned} y_j^2(t) &\leq \left(\sum_{i=1}^{N} \left| \sqrt{s_i(\boldsymbol{\sigma}, \mathbf{x}, t)} \right|^2 \right) \left(\sum_{i=1}^{N} \left| \sqrt{s_i(\boldsymbol{\sigma}, \mathbf{x}, t)} y_{ij}(t) \right|^2 \right) \\ &= \left(\sum_{i=1}^{N} s_i(\boldsymbol{\sigma}, \mathbf{x}, t) \right) \left(\sum_{i=1}^{N} (\sqrt{s_i(\boldsymbol{\sigma}, \mathbf{x}, t)} y_{ij}(t))^2 \right) \\ &= \left(\sum_{i=1}^{N} s_i(\boldsymbol{\sigma}, \mathbf{x}, t) y_{ij}^2(t) \right). \end{aligned}$$

Taking the sum from $j = 1$ to m, and then rearranging yields

$$\begin{aligned} \sum_{j=1}^{m} y_j^2(t) &\leq \sum_{j=1}^{m} \left(\sum_{i=1}^{N} s_i(\boldsymbol{\sigma}, \mathbf{x}, t) y_{ij}^2(t) \right) \\ \mathbf{y}^{\mathsf{T}}(t)\mathbf{y}(t) &\leq \sum_{i=1}^{N} s_i(\boldsymbol{\sigma}, \mathbf{x}, t) \sum_{j=1}^{m} y_{ij}^2(t) \\ &= \sum_{i=1}^{N} s_i(\boldsymbol{\sigma}, \mathbf{x}, t) \mathbf{y}_i^{\mathsf{T}}(t)\mathbf{y}_i(t). \end{aligned}$$

Integrating both sides in time from 0 to T in time yields

$$\begin{aligned} \int_0^T \mathbf{y}^{\mathsf{T}}(t)\mathbf{y}(t)\, \mathrm{d}t &\leq \sum_{i=1}^{N} \int_0^T s_i(\boldsymbol{\sigma}, \mathbf{x}, t) \mathbf{y}_i^{\mathsf{T}}(t)\mathbf{y}_i(t)\, \mathrm{d}t \\ \|\mathbf{y}(t)\|_{2T}^2 &\leq \sum_{i=1}^{N} \left\| \sqrt{s_i(\boldsymbol{\sigma}, \mathbf{x}, t)} \mathbf{y}_i(t) \right\|_{2T}^2. \end{aligned} \tag{B.14}$$

Multiplying both sides of (B.14) by -1 yields (9.9).

B.5 Equivalence of (9.10) and (9.22)

Let $\mathbf{\Pi} = \tilde{\mathbf{P}}^{-1}$, and pre and post multiply (9.16) by

$$\begin{bmatrix} \mathbf{\Pi} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix}$$

to obtain

$$\begin{bmatrix} \mathbf{A}_i\mathbf{\Pi} + \mathbf{\Pi}\mathbf{A}_i^\mathsf{T} + \frac{1}{b}\mathbf{\Pi}\mathbf{C}_i^\mathsf{T}\mathbf{C}_i\mathbf{\Pi} & \mathbf{B}_i - \frac{a+b}{2b}\mathbf{\Pi}\mathbf{C}_i^\mathsf{T} \\ \star & a\mathbf{1} \end{bmatrix} \leq 0,$$

which is equal to

$$\begin{aligned} 0 \geq & \begin{bmatrix} \mathbf{A}_i\mathbf{\Pi} + \mathbf{\Pi}\mathbf{A}_i^\mathsf{T} & \mathbf{B}_i \\ \star & a - \frac{(a+b)^2}{4b}\mathbf{1} \end{bmatrix} + \begin{bmatrix} \mathbf{\Pi}\mathbf{C}_i^\mathsf{T}\mathbf{C}_i\mathbf{\Pi} & -\frac{a+b}{2b}\mathbf{\Pi}\mathbf{C}_i^\mathsf{T} \\ \star & \frac{(a+b)^2}{4b}\mathbf{1} \end{bmatrix} \\ = & \begin{bmatrix} \mathbf{A}_i\mathbf{\Pi} + \mathbf{\Pi}\mathbf{A}_i^\mathsf{T} & \mathbf{B}_i \\ \star & -\frac{(a-b)^2}{4b}\mathbf{1} \end{bmatrix} + \begin{bmatrix} \mathbf{\Pi}\mathbf{C}_i^\mathsf{T} \\ -\frac{a+b}{2}\mathbf{1} \end{bmatrix} (-\frac{1}{b}\mathbf{1}) \begin{bmatrix} \mathbf{C}_i\mathbf{\Pi} & -\frac{a+b}{2}\mathbf{1} \end{bmatrix}. \end{aligned} \tag{B.15}$$

Taking the Schur complement of (B.15) yields

$$\begin{bmatrix} \mathbf{\Pi}\mathbf{A}_i^\mathsf{T} + \mathbf{A}_i\mathbf{\Pi} & \mathbf{B}_i & \mathbf{C}_i\mathbf{\Pi} \\ \star & -\frac{(a-b)^2}{4b}\mathbf{1} & -\frac{a+b}{2}\mathbf{1} \\ \star & \star & -b\mathbf{1} \end{bmatrix} \leq 0. \tag{B.16}$$

Equation (B.16) and (9.22) are equivalent when the controller matrices and controller conic bounds are used.

Bibliography

Bibliography

[1] A. Walsh, J. J. Ryan, and J. R. Forbes, "Constrained extremum-seeking guidance using a constrained kalman filter," in *Proc. 2017 American Control Conference*, (Seattle, WA), pp. 1290–1295, 2017.

[2] A. Walsh, J. R. Forbes, S. A. Chee, and J. J. Ryan, "Kalman-filter-based unconstrained and constrained extremum-seeking guidance on SO(3)," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 9, pp. 2260–2271, 2017.

[3] A. Walsh and J. R. Forbes, "Very strictly passive controller synthesis with affine parameter dependence," *IEEE Transactions on Automatic Control*, 2017.

[4] C. Thomas, "EC01-0328-4," 2001. [Online] Available https://images.nasa.gov/details-EC01-0328-4.html. [Accessed: 14-Nov-2017].

[5] K. F. Lee and K. M. Luk, *Microstrip Patch Antennas*. London, UK: Imperial College Press, 2011.

[6] D. Popovic, M. Jankovic, S. Magner, and A. R. Teel, "Extremum seeking methods for optimization of variable cam timing engine operation," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 3, pp. 398–407, 2006.

[7] S. Drakunov, U. Ozguner, P. Dix, and B. Ashrafi, "ABS control using optimum search via sliding modes," *IEEE Transactions on Control Systems Technology*, vol. 3, no. 1, pp. 79–85, 1995.

[8] C. Zhang and R. Ordóñez, *Extremum-Seeking Control and Applications: A Numerical Optimization-Based Approach*. Advances in Industrial Control, London, United Kingdom: Springer-Verlag, 2012.

[9] A. Ghaffari, M. Krstić, and S. Seshagiri, "Power optimization and control in wind energy conversion systems using extremum seeking," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1684–1695, 2014.

[10] M. Ye and G. Hu, "Distributed extremum seeking for constrained networked optimization and its application to energy consumption control in smart grid," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 6, pp. 2048–2058, 2016.

[11] M. Guay, D. Dochain, M. Perrier, and N. Hudon, "Flatness-based extremum-seeking control over periodic orbits," *IEEE Transactions on Automatic Control*, vol. 52, no. 10, pp. 2005–2012, 2007.

[12] K. B. Ariyur and M. Kristić, *Real-Time Optimization by Extremum-Seeking Control*. Hoboken, New Jersey: John Wiley & Sons, Inc., 2003.

[13] J. J. Ryan and J. L. Speyer, "Peak-seeking control using gradient and Hessian estimates," in *Proc. 2010 American Control Conference*, (Baltimore, MD), 2010.

[14] P. C. Hughes, *Spacecraft Attitude Dynamics*. Mineola, N.Y.: Dover Publications, 2004.

[15] R. Ortega, J. Loría Perez, P. Nicklasson, and H. Sira-Ramirez, *Passivity-Based Control of Euler-Lagrange Systems: Mechanical, Electrical, and Electromechanical Applications*. London; New York: Springer, 1998.

[16] E. Lavretsky and K. A. Wise, *Robust and Adaptvie Control: With Aerospace Applications*. London, United Kingdom: Spring-Verlag, 2013.

[17] B. M. Wen, *Robust and H_∞ Control*. London, United Kingdom: Spring-Verlag, 2000.

[18] L. J. Bridgeman and J. R. Forbes, "Conic-sector-based control to circumvent passivity violations," *International Journal of Control*, vol. 87, no. 8, pp. 1467–1477, 2014.

[19] L. J. Bridgeman, R. J. Caverly, and J. R. Forbes, "Conic-sector-based controller synthesis: Theory and experiments," in *Proc. 2014 American Control Conference (ACC)*, (Portland, Oregon, USA), 2014.

[20] L. J. Bridgeman and J. R. Forbes, "The extended conic sector theorem," *IEEE Transactions on Automatic Control*, vol. 61, no. 7, pp. 1931–1937, 2016.

[21] L. J. Bridgeman and J. R. Forbes, "A comparative study of input-output stability results," *IEEE Transactions on Automatic Control*, 2017.

[22] C. Hoffmann and H. Werner, "A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, 2014.

[23] J. S. Shamma and M. A. Athans, "Guaranteed properties of gain scheduled control for linear parameter-varying plants," *Automatica*, vol. 27, no. 3, pp. 559–564, 1991.

[24] P. Apkarian and R. Adams, "Advanced gain-scheduling techniques for uncertain systems," *IEEE Transactions on Control Systems Technology*, vol. 6, no. 1, pp. 21–32, 1996.

[25] W. J. Rugh and J. S. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, pp. 1401–1425, 2000.

[26] J. Mohammadpour and C. W. Scherer, *Control of Linear Parameter Varying Systems with Applications*. Boston, MA: Springer, 2012.

[27] G. E. Dellerud and F. G. Paganini, *A Course in Robust Control Theory: A Convex Approach*. New York, NY: Springer-Verlag, 2000.

[28] C. A. Desoer and M. Vidyasagar, *Feedback Systems: Input-Output Properties*. New York, NY: Academic Press., 1975.

[29] K. Lange, *Optimization*. New York, NY: Springer, 2013.

[30] D. G. Luenberger, *Optimization by Vector Space Methods*. New York, NY: John Wiley & Sons, Inc., 1969.

[31] S. P. Boyd, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia: Society for Industrial and Applied Mathematics, 1994.

[32] M. Wolfe and R. Osten, "JWST Primer v 3.0," tech. rep., Space Telescope Science Institute, 2014.

[33] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch, "Rigid-body attitude control," *IEEE Control Systems Magazine*, vol. 31, no. 3, pp. 30–51, 2011.

[34] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*. New York, NY: Springer, 2014.

[35] A. Weiss, I. Kolmanovsky, D. S. Bernstein, and A. Sanyal, "Inertia-free spacecraft attitude control using reaction wheels," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 5, pp. 1425–1439, 2013.

[36] S. Weng, D. Yue, and T. Yang, "Coordinated attitude motion control of multiple rigid bodies on manifold SO(3)," *IET Control Theory & Applications*, vol. 7, no. 16, pp. 1984–1991, 2013.

[37] J. R. Forbes, "Passivity-based attitude control on the special orthogonal group of rigid-body rotations," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 6, pp. 1596–1605, 2013.

[38] J. R. Forbes, "Direction-cosine-matrix-based attitude control subject to actuator saturation," *IET Control Theory & Applications*, vol. 9, no. 11, pp. 1653–1661, 2015.

[39] T. Lee, "Global exponential attitude tracking controls on SO(3)," *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2837–2842, 2015.

[40] R. Mahony, T. Hamel, and J. M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[41] D. Choukroun, H. Weiss, I. Y. Bar-Itzhack, and Y. Oshman, "Direction cosine matrix estimation from vector observations using a matrix Kalman filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 1, pp. 61–79, 2010.

[42] D. E. Zlotnik and J. R. Forbes, "Exponential convergence of a nonlinear attitude estimator," *Automatica*, vol. 72, pp. 11–18, 2016.

[43] D. E. Zlotnik and J. R. Forbes, "Nonlinear estimator design on the special orthogonal group using vector measurements directly," *IEEE Transactions on Automatic Control*, vol. 62, no. 1, pp. 149–160, 2017.

[44] T. Lee, M. Leok, and N. H. McClamroch, "Optimal attitude control of a rigid body using geometrically exact computations on SO(3)," *Journal of Dynamical and Control Systems*, vol. 14, no. 4, pp. 465–487, 2008.

[45] A. Saccon, J. Hauser, and A. P. Aguiar, "Optimal control on lie groups: The projection operator approach," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2230–2245, 2013.

[46] U. Kalabić, R. Gupta, S. D. Cairano, A. Bloch, and I. Kolmanovsky, "Constrained spacecraft attitude control on SO(3) using reference governors and nonlinear model predictive control," *2014 American Control Conference*, pp. 5586–5593, 2014.

[47] R. Dai and C. Sun, "Path planning of spatial rigid motion with constrained attitude," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 8, pp. 1356–1365, 2015.

[48] H. C. Kjellberg and E. G. Lightsey, "Discretized quaternion constrained attitude pathfinding," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 3, pp. 713–718, 2015.

[49] H. C. Kjellberg and E. G. Lightsey, "Discretized constrained attitude pathfinding and control for satellites," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 5, pp. 1301–1309, 2013.

[50] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[51] Y. Kim and M. Mesbahi, "Quadratically constrained attitude control via semidefinite programming," *IEEE Transactions on Automatic Control*, vol. 49, no. 5, pp. 731–735, 2004.

[52] Y. Kim, M. Mesbahi, G. Singh, and F. Y. Hadaegh, "On the convex parameterization of constrained spacecraft reorientation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 3, pp. 1097–1109, 2010.

[53] X. Liu and P. Lu, "Solving nonconvex optimal control problems by convex optimization," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 3, pp. 750–765, 2014.

[54] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*. New York, NY: Springer-Verlag, 2014.

[55] A. Walsh, D. E. Zlotnik, and J. R. Forbes, "Modelling of spacecraft with N reaction wheels using arbitrary attitude parameterizations," *Advances in the Astronautical Sciences*, vol. 157, pp. 945–956, 2016.

[56] H. Frenk, K. Roos, T. Terlaky, S. Zhang, E. Andersen, and K. Andersen, *High Performance Optimization*, vol. 33 of *Applied Optimization*, ch. The MOSEK Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm, pp. 197–232. Springer, 2000.

[57] J. Löfberg, "Yalmip : a toolbox for modeling and optimization in matlab," in *IEEE International Symposium on Computer Aided Control Systems Design*, pp. 284–289, 2004.

[58] Y. Tan, Y. Li, and I. M. Y. Mareels, "Extremum seeking for constrained inputs," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2405–2410, 2013.

[59] M. Guay, E. Moshksar, and D. Dochain, "A constrained extremum-seeking control approach," *International Journal of Robust and Nonlinear Control*, vol. 25, no. 16, pp. 3132–3153, 2015.

[60] D. DeHaan and M. Guay, "Extremum-seeking control of state-constrained nonlinear systems," *Automatica*, vol. 41, pp. 1567–1574, 9 2005.

[61] G. Mills and M. Krstic, "Constrained extremum seeking in 1 dimension," in *Proc. 53rd IEEE Conference on Decision and Control*, (Los Angeles, CA), pp. 2654–2659, 2014.

[62] V. Koropouli, A. Gusrialdi, and D. Lee, "An adaptive dynamic inversion-extremum seeking control approach for constrained robotic motion tasks," in *Proceedings of the 2015 European Control Conference*, (Linz, Austria), pp. 2786–2791, 2015.

[63] S. Ko and R. R. Bitmead, "State estimation for linear systems with state equality constraints," *Automatica*, vol. 43, no. 8, pp. 1363–1368, 2007.

[64] B. Teixeira, J. Chandrasekar, H. J. Palanthandalam-Madapusi, L. Torres, L. A. Aguirre, and D. S. Bernstein, "Gain-constrained Kalman filtering for linear and nonlinear systems," *IEEE Transactions on Signal Processing*, vol. 56, no. 9, pp. 4113–4123, 2008.

[65] D. Simon, "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms," *IET Control Theory & Applications*, vol. 4, no. 8, pp. 1303–1318, 2010.

[66] D. F. Chichka, J. L. Speyer, C. Fanti, and C. G. Park, "Peak-seeking control for drag reduction in formation flight," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 5, pp. 1221–1230, 2006.

[67] A. Chakraborty, P. Seiler, and G. J. Balas, "Applications of linear and nonlinear robustness analysis techniques to the F/A-18 flight control laws," in *Proc. AIAA Guidance, Navigation, and Control Conference*, (Chicago, Illinois), 10-13 Aug. 2009.

[68] J. L. Hansen and B. R. Cobleigh, "Induced moment effects of formation flight using two F/A-18 aircraft," in *Proc. Flight Mechanics Conference and Exhibit*, (Monterey, CA), 2002.

[69] M. J. Vachon, R. J. Ray, K. R. Walsh, and K. Ennix, "F/A-18 aircraft performance benefits measured during the autonomous formation flight project," in *Proc. AIAA Flight Mechanics Conference and Exhibit*, (Monterey, CA), 2002.

[70] D. Simon and T. L. Chia, "Kalman filtering with state equality constraints," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 1, pp. 128–136, 2002.

[71] D. J. Bugajski and D. F. Enns, "Nonlinear control law with application to high angle-of-attack flight," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 3, pp. 761–767, 1992.

[72] F. Coito, J. M. Lemos, and S. S. Alves, "Stochastic extremum seeking in the presence of constraints," in *16th IFAC World Congress*, vol. 38, pp. 276–281, 2005.

[73] F. Taringoo, D. Nešić, Y. Tan, and P. M. Dower, "Extremum seeking control for nonlinear systems on compact riemannian manifolds," in *53rd IEEE Conference on Decision and Control*, (Los Angeles, CA), pp. 2667–2672, 2014.

[74] H.-B. Dürr, M. Stanković, K. H. Johansson, and C. Ebenbauer, "Extremum seeking on submanifolds in the euclidian space," *Automatica*, vol. 50, no. 10, pp. 2591–2596, 2014.

[75] M. Guay and D. Dochain, "A minmax extremum-seeking controller design technique," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1874–1886, 2014.

[76] F. Yang and Y. Li, "Set-membership filtering with state constraints," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 4, pp. 1619–1629, 2009.

[77] F. Yang and Y. Li, "Set-membership filtering for discrete-time systems with nonlinear equality constraints," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2480–2486, 2009.

[78] A. Weiss, F. Leve, M. Baldwin, J. R. Forbes, and I. Kolmanovsky, "Spacecraft constrained attitude control using positively invariant constraint admissible sets on $\mathrm{SO}(3)\times\mathbb{R}^3$," in *Proceedings of the 2014 American Control Conference*, (Portland, OR), pp. 4955–4960, 2014.

[79] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press, 2008.

[80] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, United Kingdom: Cambridge University Press, 2004.

[81] D. S. Bemstein, *Matrix Mathematics: Theory, Facts, and Formulas*. Princeton, NJ: Princeton University Press, 2 ed., 2009.

[82] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York, NY: Springer, 2 ed., 2006.

[83] A. H. Jazwinski, *Stochastic Processes and Filtering Theory*. New York, NY: Academic Press, Inc., 1970.

[84] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[85] J. Wertz and W. J. Richard. Larson, eds., *Space Mission Analysis and Design*. Hawthorne, CA: Microcosm Press, 3rd ed ed., 1999.

[86] F. M. Sobolic and D. S. Bernstein, "Kalman-filter-based time-varying parameter estimation via retrospective optimization of the process noise covariance," in *Proceedings of the 2016 American Control Conference*, (Boston, MA), pp. 4545–4550, 2016.

[87] A. J. v. d. Schaft, *L_2-gain and passivity techniques in nonlinear control*. London; New York: Springer, 1996.

[88] B. Brogliato, R. Lozano, B. Maschke, and O. Egeland, *Dissipative Systems Analysis and Control Theory and Applications*. London: Springer, 2 ed., 2007.

[89] J. T. Wen, "Time domain and frequency domain conditions for strict positive realness," *IEEE Transactions on Automatic Control*, vol. 33, no. 10, pp. 988–992, 1988.

[90] G. Zames, "On the input-output stability of time-varying nonlinear feedback systems part I & II," *IEEE Transactions on Automatic Control*, vol. 11, no. 2, pp. 229–231, 1966.

[91] S. Joshi and A. Kelkar, "Design of norm-bounded and sector-bounded lqg controllers for uncertain systems," *Journal of Optimization Theory and Applications*, vol. 113, no. 2, pp. 269–282, 2002.

[92] S. Gupta and S. M. Joshi, "Some properties and stability results for sector-bounded lti systems," in *Proc. 33rd Conference on Decision and Control*, vol. 3, (Lake Buena Vista, FL), pp. 2973–2978, 1994.

[93] D. J. Leith and W. E. Leithead, "Survey of gain-scheduling analysis and design," *International Journal of Control*, vol. 73, no. 11, pp. 1001–1025, 2000.

[94] C. Mracek, J. Clontier, and C. D'Souza, "A new technique for nonlinear estimation," in *Proc. 1996 American Control Conference (ACC)*, (Dearborn, MI), pp. 338–343, 1996.

[95] J. R. Forbes and C. J. Damaren, "Design of gain-scheduled strictly positive real controllers using numerical optimization for flexible robotic systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 132, no. 3, 2010.

[96] C. J. Damaren, "Gain scheduled SPR controllers for nonlinear flexible systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 118, no. 4, pp. 698–703, 1996.

[97] A. Walsh and J. R. Forbes, "Analysis and synthesis of input strictly passive gain-scheduled controllers," *Journal of the Franklin Institute*, 2016, doi:10.1016/j.jfranklin.2016.03.014.

[98] A. Walsh and J. R. Forbes, "A very strictly passive gain-scheduled controller: Theory and experiments," *IEEE/ASME Transactions on Mechatronics*, 2016.

[99] A. Megretski and A. Rantzer, "System analysis via integral quadratic constraints," *IEEE Transactions on Automatic Control*, vol. 42, no. 6, pp. 819–830, 1997.

[100] C. W. Scherer, "LPV control and full block multipliers," *Automatica*, vol. 37, no. 3, pp. 361–375, 2001.

[101] B. P. Rasmussen and A. G. Alleyne, "Gain scheduled control of an air conditioning system using the youla parameterization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 5, p. 1216, 2010.

[102] P. Apkarian, P. Gahinet, and G. Becker, "Self-scheduled $\mathcal{H}_\infty$ control of linear parameter-varying systems: a design example," *Automatica*, vol. 31, pp. 1251–1261, 9 1995.

[103] R. Shorten, K. S. Narendra, and O. Mason, "A result on common quadratic Lyapunov functions," *IEEE Transactions on Automatic Control*, vol. 48, no. 1, pp. 110–113, 2003.

[104] T. J. Laffey and H. Šmigoc, "Tensor conditions for the existence of a common solution to the Lyapunov equation," *Linear Algebra and its Applications*, vol. 420, pp. 672–685, 1 2007.

[105] W. M. Haddad, D. S. Bemstein, and Y. W. Wang, "Dissipative $\mathcal{H}_2/\mathcal{H}_\infty$ controller synthesis," *IEEE Transactions on Automatic Control*, vol. 39, no. 4, pp. 827–831, 1994.

[106] J. C. Geromel and P. B. Gapski, "Synthesis of positive real $\mathcal{H}_2$ controllers," *IEEE Transactions on Automatic Control*, vol. 42, pp. 988–992, July 1997.

[107] T. Shimomura and S. P. Pullen, "Strictly positive real $\mathcal{H}_2$ controller synthesis via iterative algorithms for convex optimization," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 6, pp. 1003–1011, 2002.

[108] J. R. Forbes, "Dual approaches to strictly positive real controller synthesis with a $\mathcal{H}_2$ performance using linear matrix inequalities," *International Journal of Robust and Nonlinear Control*, vol. 23, pp. 903–918, 2013.

[109] M. S. Sadabadi and A. Karimi, "Fixed-order $\mathcal{H}_\infty$ and $\mathcal{H}_2$ controller design for continuous-time polytopic systems: An lmi-based approach," in *Proc. 2013 European Control Conference (ECC)*, pp. 1132–1137, 2013.

[110] M. S. Sadabadi and A. Karimi, "Fixed-order control of LTI systems subject to polytopic uncertainty via the concept of strictly positive realness," in *Proc. 2015 American Control Conference (ACC)*, (Chicago, Illinois, USA), pp. 2882–2887, 2015.

[111] Z. Emedi and A. Karimi, "Fixed-structure LPV discrete-time controller design with induced $\mathcal{L}_2$-norm and performance," *International Journal of Control*, 2015.

[112] C. Scherer, P. Gahinet, and M. Chilali, "Multiobjective output-feedback control via LMI optimization," *IEEE Transactions on Automatic ControlAutomatic Control*, vol. 42, no. 7, pp. 896–911, 1997.

[113] R. Benhabib, R. Iwens, and R. Jackson, "Stability of large space structure control systems using positivity concepts," *Journal of Guidance and Control*, vol. 4, no. 5, pp. 487–494, 1981.

[114] P. Gahinet and P. Apkarian, "A linear matrix inequality approach to $\mathcal{H}_\infty$ control," *International Journal of Robust and Nonlinear Control*, vol. 4, no. 4, pp. 421–448, 1994.

[115] G. Becker, A. Packard, D. Philbrick, and G. Balas, "Control of parametrically-dependent linear systems: A single quadratic lyapunov approach," in *Proc. 1993 American Control Conference (ACC)*, (San Francisco, California, USA), pp. 2795–2799, 2-4 June 1993.

[116] Quanser, *2-DOF Serial Flexible Joint Robot User Manual*. Markham, Ontario, 2013.

[117] K. Toh, M. Todd, and R. Tutuncu, "SDPT3 — a matlab software package for semidefinite programming," *Optimization Methods and Software*, vol. 11, pp. 545–581, 1999.

[118] R. Tutuncu, K. Toh, and M. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3," *Mathematical Programming Ser. B*, vol. 95, pp. 189–217, 2003.

[119] F. D. Bianchi and R. S. P. na, "Interpolation for gain-scheduled control with guarantees," *Automatica*, vol. 47, pp. 239–243, 2011.

[120] J. R. Forbes and C. J. Damaren, "Synthesis of optimal finite-frequency controllers able to accommodate passivity violations," *IEEE Transactions on Control Systems Technology*, vol. PP, no. 99, pp. 1–1, 2012.

[121] K. M. Hangos, J. Bokor, and G. Szederkényi, *Analysis and Control of Nonlinear Process Systems*. London, United Kingdom: Springer, 2004.

[122] H. V. Henderson and S. R. Searle, "On deriving the inverse of a sum of matrices," *SIAM Review*, vol. 23, no. 1, pp. 53–60, 1981.