5-2023

Convexity Applications in Single and Multi-Agent Control

Olli Nikodeemus Jansson
Utah State University

CONVEXITY APPLICATIONS IN SINGLE AND MULTI-AGENT CONTROL

by

Olli Nikodeemus Jansson

A dissertation submitted in partial fulfillment
of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

in

Mechanical Engineering

Approved:

Matthew W. Harris, Ph.D. Major Professor	David K. Geller, Ph.D. Committee Member
Tianyi He, Ph.D. Committee Member	Douglas Hunsaker, Ph.D. Committee Member
Greg Droge, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Vice Provost for Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2023

ABSTRACT

Convexity Applications in Single and Multi-agent Control

by

Olli Nikodeemus Jansson, Doctor of Philosophy

Utah State University, 2023

Major Professor: Matthew W. Harris, Ph.D.
Department: Mechanical and Aerospace Engineering

The focus of this dissertation is in the application of convexity for control problems; specifically, single-agent problems with linear or nonlinear dynamics and multi-agent problems with linear dynamics. A mixture of convex and non-convex constraints for optimal control problems is also considered. The main contributions of this dissertation include: 1) a convexification of single-agent problems with linear dynamics and annular control constraint, 2) a technique for controlling bounded nonlinear single-agent systems, and 3) a technique for solving multi-agent pursuit-evasion games with linear dynamics and convex control and state constraints. The first result shows that for annularly constrained linear systems, controllability is a sufficient condition for a free or fixed time problem to be solvable as a sequence of convex optimization problems. The second result shows that if a nonlinear system is bounded and "ordered", it is possible to use a convex combination of bounding linear systems to design a control for the nonlinear system. The third result takes advantage of a convex reachable set computation for each agent in solving games using a geometrical approach. Altogether, the theoretical and computational results demonstrate the significance of convex analysis in solving non-convex control problems.

(168 pages)

PUBLIC ABSTRACT

Convexity Applications in Single and Multi-agent Control

Olli Nikodeemus Jansson

The focus of this dissertation is in the application of convexity for control problems; specifically, single-agent problems with linear or nonlinear dynamics and multi-agent problems with linear dynamics. A mixture of convex and non-convex constraints for optimal control problems is also considered. The main contributions of this dissertation include: 1) a convexification of single-agent problems with linear dynamics and annular control constraint, 2) a technique for controlling bounded nonlinear single-agent systems, and 3) a technique for solving multi-agent pursuit-evasion games with linear dynamics and convex control and state constraints. The first result shows that for annularly constrained linear systems, controllability is a sufficient condition for a free or fixed time problem to be solvable as a sequence of convex optimization problems. The second result shows that if a nonlinear system is bounded and "ordered", it is possible to use a convex combination of bounding linear systems to design a control for the nonlinear system. The third result takes advantage of a convex reachable set computation for each agent in solving games using a geometrical approach. Altogether, the theoretical and computational results demonstrate the significance of convex analysis in solving non-convex control problems.

ACKNOWLEDGMENTS

First and foremost I would like to thank my major professor Dr. Matt Harris. I have been fortunate to have Dr. Harris as my adviser since the first day I stepped on USU campus. Dr. Harris is a great researcher but also a great teacher who truly cares about the students working for him. I am forever grateful for all the knowledge on control theory, optimization, as well as life in general that he has shared with me throughout my graduate studies.

I would also like to thank my family, friends, and people at the Advanced Aerospace Dynamics Lab for their support and encouragement. I could not have done this without them. Finally, I wish to acknowledge Office of Naval Research for supporting part of this research through grant N00013-22-1-2131.

Olli Jansson

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The topic of this dissertation is the application of convexity for single and multi-agent control problems. Single-agent control problems studied in this dissertation include ones with linear and nonlinear dynamics. For single-agent optimal control problems, non-convex control constraints in the presence of linear and nonlinear dynamics are also studied. For multi-agent control problems, this dissertation focuses on time-optimal pursuit-evasion games with linear dynamics and convex state and control constraints. These games may include single or multiple pursuers and evaders.

Control problems are usually categorized as either linear or nonlinear. Linear control problems tend to be easier to solve using linear algebraic techniques. Optimization problems are usually categorized as either convex or non-convex. Convex optimization problems are typically easier to solve using interior point methods. There are well-known techniques to linearize special classes of nonlinear problems [2–4] and, in recent years, significant research has been done on convexification of non-convex optimization problems [5–11].

1.1 Optimal Control Problems with Annular Control Constraints

Annular control constraints appear in many real-time control applications that involve systems with complex actuator models – particularly in the entry, descent, and landing of spacecraft with chemical thrusters [5, 12–14]. Chemical thrusters fail to operate reliably under a certain thrust level, thereby introducing a non-zero lower bound on the thrust magnitude. This lower bound is non-convex, and as such, the resulting trajectory optimization problems are difficult to solve. Existing nonlinear programming based trajectory optimization methods do not guarantee convergence to solutions (optimal or feasible) making them inappropriate for real-time applications. If, however, the problem can be written in a convex form, then powerful interior point methods can be customized to find globally optimal

solutions in polynomial-time. This appears to be the approach taken by SpaceX in their rocket landings [15].

When studying convexification of non-convex problems, the term lossless convexification comes up. The term lossless convexification refers to the two-stage process of 1) relaxing the annular, non-convex constraint to a convex form and 2) proving optimal solutions for the relaxed problem are also optimal for the original problem. These relaxation techniques have been studied extensively for annularly constrained linear systems [7], nonlinear systems [16], and linear systems with explicit state constraints and additional control inequalities [8]. Each of these proofs requires an assumption on the gradient of the final point. In general, the assumption cannot be verified a priori since it depends on the optimal solution. However, one special case where it can be verified a priori is the free final time transfer between fixed points, which is assumed directly in [16]. The proofs also have in common controllability/observability assumptions. In [7], controllability of the linear system (observability of the dual system) is assumed. In [16], controllability of the system linearized about each extremal is assumed. In [8, 17], strong controllability is assumed.

1.2 Nonlinear Single-agent Problems

Many control systems include nonlinear dynamics. The popular book on nonlinear control by Khalil [2] gives several examples in Chapter 1 including circuits, robotic systems, automotive systems, and more. In fortunate cases, an exact or approximate linearization of the dynamics may be obtained so that linear control techniques are effective in achieving the control objectives [3]. In other cases, the control engineer must design a controller using techniques of nonlinear control theory. These techniques often involve finding a Lyapunov function, solving a non-convex optimal control problem, or something else of equal difficulty. The presence of practical actuator or state constraints further increases the control design challenge. A short survey of different control approaches for nonlinear systems is given next.

1.2.1 Linearization Techniques

Beyond standard linearization by computing the Jacobian, recent research has advanced the idea of approximating the nonlinear system by a higher-order linear one. Techniques include approximation of the Koopman operator, data-driven modeling, and the use of machine-learning tools [18]. Whereas a standard linearization retains the same dimension as the original system, these recent techniques *lift* the state through nonlinear maps to a potentially much larger dimension. In certain cases, this may result in an exact linearization [19]. This is the ideal case. However, it is well-known that some nonlinear systems exhibit purely nonlinear phenomena such as finite-time escape and limit-cycles [2], in which case an exact finite-dimensional linearization is beyond reach. A linear parameter varying (LPV) or quasi-linear parameter varying (q-LPV) techniques may also be used to linearize the system such that the varying parameter is bounded by a convex polytope [20–24]. This allows the use of linear matrix inequalities (LMIs) to design controllers for parameters at each vertex of the polytope offline. A convex combination of these controllers is then used online based on the current parameter value.

1.2.2 Nonlinear Control Techniques

There exists a large set of techniques from nonlinear control theory. These include Lyapunov analysis, backstepping, passivity-based control, sliding mode control and other robust techniques, adaptive control, feedback linearization, and more [2]. Feedback linearization allows linear control techniques to be used in conjunction with a nonlinear feedback law. When applicable, this can simplify the analysis, though it is known that such laws are not robust. Moreover, even systems that are feedback linearizable may not be so on the domain of interest. In this case, the control magnitude may become unbounded.

Most of these techniques assume the control is unconstrained or subject only to saturation. Recent work using control Lyapunov functions has analyzed nonlinear control affine systems with polytopic control bounds [25]. Their algorithm minimizes the control magnitude pointwise in time; and as such, requires the solution of a quadratic program (QP) pointwise in time.

1.2.3 Non-convex Optimization

Optimal control theory provides a useful framework for formulating control problems with nonlinear dynamics, actuator constraints, objectives, and various boundary conditions [26, 27]. Solving such problems, however, remains a challenging task. One approach is the "indirect" approach wherein the state and costate equations are solved in continuous time through a shooting algorithm or something similar. The key challenge in this approach is obtaining reasonable estimates for the costates so that convergence is obtained. An alternative "direct" approach is to discretize the continuous problem into a nonlinear programming problem (NLP) that can be solved using a generic NLP solver [28]. The key challenge in this approach is obtaining reasonable estimates for the control values at the discrete times so that convergence is obtained. Recent work in sequential convex programming (SCP) has apparently led to significant improvements in computation time, though theoretical convergence guarantees remain weak [29]. Yet another approach is model predictive control (MPC) or receding horizon control (RHC). Standard MPC/RHC solves an optimization problem with the nonlinear dynamics or its linearization, advances time forward, and resolves the optimization problem [30].

For special cases in which the dynamics are linear (or can be exactly linearized) but other constraints are non-convex, a number of relaxations have been proposed leading to polynomial-time (fast and provable) algorithms for their solution [7–9, 31–33]. These relaxations exploit system properties such as controllability [34], strong observability [35], and normality [36], which are more difficult to synthesize for nonlinear systems [37]. They use interior-point methods for numerical solution [38, 39]. However, they only apply to linear dynamical systems.

1.3 Pursuit-Evasion Games

Pursuit-evasion games are commonplace in engineering as well as other fields such as economics. There are a variety of techniques available to solve pursuit-evasion games. Generally speaking, however, their solution is more difficult to solve than those of an individual pursuer's or evader's optimal control probelms would be, which can be difficult in their own

right. One approach is the variational approach wherein the necessary conditions of optimal control are written for both players and solved simultaneously. This is briefly described by Bryson and Ho for a special set of linear-quadratic games and time-optimal games with scalar control [40].

Due to the complex nature of differential games, an analytical solution using the variational approach may not always be reached. This may happen, for example, when the problem has nonlinear dynamics or control constraints. Instead of solving the problem by calculus of variation techniques, numerical techniques have also been used successfully to solve games using neural networks [41] and genetic algorithms [42–44] leading to so-called semi-direct methods. A purely direct method known as the iterative or relaxation approach has also been used and offers a convergence guarantee when both sub-problems are convex [45, 46].

A different approach is geometric in nature and requires the calculation of each player's reachable set. This is motivated by the work of Mizukami [47] where it was shown that the solution to a single-pursuer and single-evader pursuit-evasion game lies on the boundary of both pursuer's and evader's reachable sets. An analytical solution for capture time and optimal trajectory was derived when control inputs of the players are restricted by an integral constraint. Reachable sets have also been used for control constrained problems in dynamic flowfields [48], robotics, development of trade studies, and more [49–52].

1.4 Objectives

This dissertation addresses the following research topics.

1. For linear systems with annular control constraints and fixed final time:
   (a) Investigate methods to solve these problems in real-time.
   (b) Determine conditions for such method to work.
2. For bounded nonlinear systems:
   (a) Investigate ways to bound nonlinear systems with linear systems.

(b) Determine conditions on when the nonlinear system can be controlled by bounding linear systems.

(c) Investigate adding convex and non-convex control constraints into the original nonlinear system.

3. For linear multi-agent systems:

(a) Explore solving single-pursuer, single-evader pursuit-evasion games with control and state constraints using reachable sets.

(b) Examine generalizing this to multi-pursuer and multi-evader games.

1.5 Outline

An overview of the remaining chapters is now given.

Chapter 2: Background

Chapter 2 introduces some mathematical and system theory background that will be used during the remainder of the work. Topics covered in this chapter include convexity, linear systems, and discretization.

Chapter 3: Linear Single-agent Problems with Annular Control Constraints

This chapter includes work that has been done for linear single-agent problems with annular control constraints as well as a literature review for this subject. The work in this chapter has been published in [32]. The chapter provides a sufficient condition for the standard convexification to hold for both free and fixed final time problems, a sufficient condition for the standard convexification to hold for all final times between the minimum feasible time and the optimal time, and a perturbation technique to solve the general fixed time problem as a sequence of convex programs when the final time is greater than the optimal time.

Chapter 4: Control of Bounded Nonlinear Systems

This chapter discusses single-agent systems with bounded additive nonlinearities. Systems with scalar nonlinearity are considered first and a sufficient condition to control such systems with bounding linear ones is given in the presence of control constraints and convex state constraints. The sufficient condition is then generalized to systems with multi-dimensional nonlinearities in the presence of convex control and state constraints. The sufficient conditions are applied to multiple numerical examples throughout the chapter as well to demonstrate their capabilities in solving trajectory design and control problems. This work has been published in [53–55].

Chapter 5: Linear Pursuit-Evasion Games

The work related to linear pursuit-evasion games is shown in this chapter. Linear pursuit-evasion games with 1) single pursuer and single evader, 2) multiple pursuers and single evader, and 3) single pursuer and multiple evaders are considered. The games are solved using a geometrical method that utilizes approximations of the agents' reachable sets. The agents' motion is constrained by convex control input and state constraints. This work has been published in [56, 57].

Chapter 6: Conclusions and Summary

This chapter concludes and summarizes the dissertation. Some open questions related to the research are also discussed.

CHAPTER 2
BACKGROUND

This chapter provides mathematical background on convexity, dynamical systems, and discretization that is useful in subsequent chapters.

2.1 Convexity

First a definition of convexity is given for a set as well as for a function.

Definition 2.1. *[58] A set $\mathcal{S}$ is said to be convex if for every $x_1, x_2 \in \mathcal{S}$ and for every $\theta \in [0, 1]$, it is true that*

$$\theta x_1 + (1 - \theta)x_2 \in \mathcal{S} \tag{2.1}$$

In $\mathbb{R}^n$ this means that a line segment can be drawn between any two points in the set and any point on the line segment is also in the set. Figure 2.1 shows examples of convex and non-convex sets in $\mathbb{R}^2$. The set on the left is convex as any two points in the set can be connected with a line segment and any point on the line segment is in the set as well. The set on the right however is non-convex. As seen in the figure, the line segment connecting the points passes through a region that is not in the set.

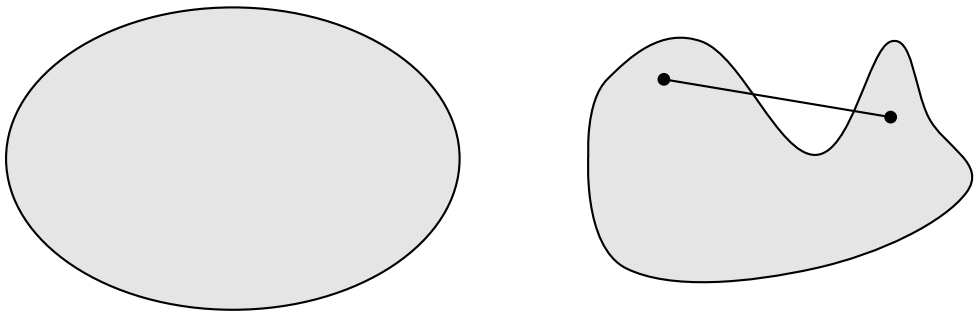


Fig. 2.1: Examples of convex and non-convex sets in $\mathbb{R}^2$.

A point that is given as $\theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_k x_k$, where $\sum_{i=1}^{k} \theta_i = 1$ and $\theta_i \geq 0,\ i = 1, 2, \ldots k$ is called a convex combination of points $x_1, x_2, \ldots, x_k$. It can be shown that a point that is given as a convex combination of points which lie in a convex set is in the convex set as well [58].

Definition 2.2. *[59] A function $f : \mathbb{R} \to \mathbb{R}$ is convex on the open set $\mathcal{D}$ if for every $x, y \in \mathcal{D}$ and for every $\theta \in [0, 1]$, it is true that*

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta) f(y) \tag{2.2}$$

The function is called strictly convex if equality holds only for $\theta = 0$ or $\theta = 1$.

This means that the line segment connecting any two points on the function is above the function. This is illustrated in Figure 2.2.

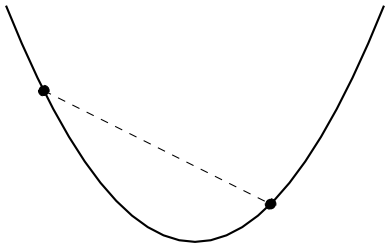


Fig. 2.2: Example of a convex function in $\mathbb{R}^2$.

It should be noted that an affine transformation of a function or set preserves its convexity [58].

2.2 Linear Dynamical Systems

2.2.1 Continuous-time

The most general representation of a continuous-time linear dynamical system used in this dissertation is

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + E(t)\delta(t) \tag{2.3}$$

where $x(t) \in \mathbb{R}^n$ is a system state vector, $u(t) \in \mathbb{R}^m$ is a control input vector, and $\delta(t) \in \mathbb{R}^q$ is a bias vector. The state function x is assumed to be continuous, and the control and bias functions are assumed to be piecewise continuous. The matrix-valued functions $A(t) \in \mathbb{R}^{n\times n}, B(t) \in \mathbb{R}^{n\times m}$, and $E(t) \in \mathbb{R}^{n\times q}$ are assumed to be continuous. The dynamical system is assumed to be globally Lipschitz. This system is also called a continuous-time linear time-varying (CLTV) system or often times just a linear time-varying (LTV) system. A unique solution to this system is given in Theorem 2.1 with initial state vector $x_0 \in \mathbb{R}^n$ so that at initial time $t_0 \in \mathbb{R}$, $x(t_0) = x_0$.

Theorem 2.1. *[60] The solution to* (2.3) *is given by*

$$x(t) = \Phi(t, t_0)x_0 + \int_{t_0}^{t} \Phi(t, \tau)\Big(B(\tau)u(\tau) + E(\tau)\delta(\tau)\Big)d\tau \tag{2.4}$$

where $\Phi(t_2, t_1)$ *is the state transition matrix.*

Proof. Assume (2.4) is the solution to (2.3). Take the derivative of (2.4) with respect to time to get

$$\frac{d}{dt}x(t) = \dot{x}(t) = \frac{d}{dt}\Phi(t, t_0)x_0 + \frac{d}{dt}\int_{t_0}^{t} \Phi(t, \tau)\Big(B(\tau)u(\tau) + E(\tau)\delta(\tau)\Big)d\tau \tag{2.5}$$

Now, recall the Leibniz integral rule [34] and notice that at $t = t_0$ the integral in the above equation disappears.

$$\begin{aligned}
\dot{x}(t) = {} & A(t)\Phi(t, t_0)x_0 + \Phi(t, t)\Big(B(t)u(t) + E(t)\delta(t)\Big) + \\
& \int_{t_0}^{t_0} \frac{\partial}{\partial t}\Phi(t, \tau)\Big(B(\tau)u(\tau) + E(\tau)\delta(\tau)\Big)d\tau + \\
& \int_{t_0}^{t} \frac{\partial}{\partial t}\Phi(t, \tau)\Big(B(\tau)u(\tau) + E(\tau)\delta(\tau)\Big)d\tau \\
= {} & A(t)\Phi(t, t_0)x_0 + \Phi(t, t)\Big(B(t)u(t) + E(t)\delta(t)\Big) + \\
& \int_{t_0}^{t} \frac{\partial}{\partial t}\Phi(t, \tau)\Big(B(\tau)u(\tau) + E(\tau)\delta(\tau)\Big)d\tau
\end{aligned} \tag{2.6}$$

Using the following two properties of the state transition matrix: $\frac{d}{dt}\phi(t,\tau) = A(t)\phi(t,\tau)$ and $\phi(t,t) = I$ [60], the above simplifies to

$$\begin{aligned}\dot{x}(t) = &A(t)\Phi(t,t_0)x_0 + B(t)u(t) + E(t)\delta(t)+\\ &A(t)\int_{t_0}^{t}\Phi(t,\tau)\Big(B(\tau)u(\tau) + E(\tau)\delta(\tau)\Big)d\tau\\ = &A(t)\left(\Phi(t,t_0)x_0 + \int_{t_0}^{t}\Phi(t,\tau)\left(B(\tau)u(\tau) + E(\tau)\delta(\tau)\right)d\tau\right) +\\ &B(t)u(t) + E(t)\delta(t)\end{aligned} \tag{2.7}$$

Substituting (2.4) into the above gives

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + E(t)\delta(t) \tag{2.8}$$

which is (2.3). □

The state transition matrix is defined in Definition 2.3 using a Peano-Baker series.

Definition 2.3. *[34] For a CLTV system as described in* (2.3) *the state transition matrix is defined as*

$$\begin{aligned}\Phi(t,t_0) = &I + \int_{t_0}^{t}A(s_1)ds_1 + \int_{t_0}^{t}A(s_1)\int_{t_0}^{s_1}A(s_2)ds_2ds_1+\\ &\int_{t_0}^{t}A(s_1)\int_{t_0}^{s_1}A(s_2)\int_{t_0}^{s_2}A(s_3)ds_3ds_2ds_1 + \dots\end{aligned} \tag{2.9}$$

If the system matrix A is not time-varying, the above equation simplifies to

$$\begin{aligned}\Phi(t,t_0) = &I + A\int_{t_0}^{t}ds_1 + A^2\int_{t_0}^{t}\int_{t_0}^{s_1}ds_2ds_1+\\ &A^3\int_{t_0}^{t}\int_{t_0}^{s_1}\int_{t_0}^{s_2}ds_3ds_2ds_1 + \dots\\ = &\sum_{k=0}^{\infty}\frac{1}{k!}A^k(t-t_0)^k\\ = &e^{A(t-t_0)}\end{aligned} \tag{2.10}$$

2.2.2 Discrete-time

A discrete-time linear dynamical system is defined as

$$x[k+1] = A[k]x[k] + B[k]u[k] + E[k]\delta[k] \tag{2.11}$$

where $k \in \mathbb{N}$. This system is referred to as a discrete-time linear time-varying (DLTV) system with the solution to it given in Theorem 2.2 with a definition of a discrete-time state transition matrix given in Definition 2.4.

Theorem 2.2. *[60] The solution to* (2.11) *is given by*

$$x[k] = \Phi[k, k_0]x_0 + \sum_{i=k_0}^{k-1} \Phi[k, i+1]\Big(B[i]u[i] + E[i]\delta[i]\Big) \tag{2.12}$$

where $\Phi[k_2, k_1]$ *is the discrete-time state transition matrix.*

Definition 2.4. *[60] For a DLTV system as described in* (2.11) *the discrete-time state transition matrix is defined as*

$$\Phi[k_2, k_1] = \begin{cases} I & k_2 = k_1 \\ A[k_2 - 1]A[k_2 - 2]\dots A[k_1 + 1]A[k_1] & k_2 > k_1 \end{cases} \tag{2.13}$$

If the discrete system matrix is not time-varying, the above simplifies to

$$\Phi[k_2, k_1] = A^{k_2 - k_1} \tag{2.14}$$

2.3 Discretization

Different discretization methods are described and compared in this section. Discretization methods are covered here because optimization tools are later used in this dissertation and require a finite-dimensional system. The continuous-time system considered in this section is

$$\dot{x}(t) = Ax(t) + Bu(t) + E\eta(x(t)) \tag{2.15}$$

where $\eta : \mathbb{R}^n \to \mathbb{R}^q$ is a nonlinear function and the other variables are the same ones as described in (2.3) except that the matrices A, B, and E are constant and not time-varying like they were in (2.3). The dynamics in (2.15) have a linear time-invarying (LTI) portion given by $Ax(t)+Bu(t)$ with an added nonlinearity from $E\eta(x(t))$. The discretization methods considered here include discretization using state transition matrix, Euler method, and 4th order Runge-Kutta method. Other discretization methods exist as well such as pseudo-spectral ones [61, 62] but are not covered here.

2.3.1 State Transition Matrix Method

This method is also often referred as exact discretization as it does discretize the LTI portion exactly as long as an assumption for a zero-order hold on control input vector $u(t)$ is valid. However, for exact discretization using this method, a zero-order hold in the nonlinear term would be needed as well which is not in general valid for this system. Because of this, the discretization is in general not exact. Using this method, the system is discretized as follows. Upon defining the state transitionl matrix $\Phi(t_{k+1}, t_k) = e^{A(t_{k+1}-t_k)}$, it can be shown by direct differentiation that a solution to (2.15) on the interval $[t_k, t_{k+1}]$ is

$$x(t_{k+1}) = \Phi(t_{k+1}, t_k)x(t_k) + \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)[Bu(\tau) + E\eta(x(\tau))]d\tau. \tag{2.16}$$

For small time steps $\Delta t = t_{k+1} - t_k$, evaluation of the integral may be approximated by fixing u and $\eta(x)$ at their initial values $u(t_k)$ and $\eta(x(t_k))$ (zero-order hold), respectively. Upon defining

$$A_d = \Phi(t_{k+1}, t_k), \quad B_d = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)Bd\tau, \quad E_d = \int_{t_k}^{t_{k+1}} \Phi(t_{k+1}, \tau)Ed\tau, \tag{2.17}$$

a discrete-time approximation of the continuous-time system is

$$x[k+1] = A_d x[k] + B_d u[k] + E_d \eta(x[k]), \tag{2.18}$$

which is similar form to (2.11). For linear systems this method is an exact discretization method as long as the zero-order hold for the control input is a valid assumption.

2.3.2 Euler Method

The next discretization method considered is Euler method which assumes zero-order hold in states, control inputs, as well as in the nonlinear term. Considering the system in (2.15), the method gives the state after a short time step $\Delta t = t_{k+1} - t_k$ as

$$x(t_{k+1}) = x(t_k) + \Big(Ax(t_k) + Bu(t_k) + E\eta(x(t_k)) \Big) \Delta t \tag{2.19}$$

With the definition of

$$A_d = (I + A)\Delta t, \quad B_d = B\Delta t, \quad E_d = E\Delta t, \tag{2.20}$$

a discrete system is given as

$$x[k+1] = A_d x[k] + B_d u[k] + E_d \eta(x[k]) \tag{2.21}$$

which again is in the similar format as (2.11).

2.3.3 4th Order Runge-Kutta Method

The last discretization method considered is the 4th order Runge-Kutta method. A zero-order hold is assumed for the control input and the nonlinear term once again. The discretization is given as

$$x[k+1] = x[k] + \frac{1}{6} \left(\psi_1 + 2\psi_2 + 2\psi_3 + \psi_4 \right) \Delta t \tag{2.22}$$

where

$$\begin{aligned}
\psi_1 &= Ax[k] + Bu[k] + E\eta\left(x[k]\right) \\
\psi_2 &= A\left(x[k] + \psi_1\frac{\Delta t}{2}\right) + Bu[k] + E\eta\left(x[k]\right) \\
\psi_3 &= A\left(x[k] + \psi_2\frac{\Delta t}{2}\right) + Bu[k] + E\eta\left(x[k]\right) \\
\psi_4 &= A\left(x[k] + \psi_3\Delta t\right) + Bu[k] + E\eta\left(x[k]\right)
\end{aligned} \tag{2.23}$$

These equations can be combined and expressed more compactly as

$$x[k+1] = \sum_{i=0}^{4}\frac{(A\Delta t)^i}{i!}x[x] + \sum_{i=1}^{4}\frac{A^{i-1}\Delta t^i}{i!}\left(Bu[k] + E\eta(x[k])\right) \tag{2.24}$$

Comparing this to (2.10) reveals that the above is simply a 4th order approximation of the state transition matrix method given in Section 2.3.1.

2.3.4 Numerical Examples

Numerical examples of discretization of both linear and nonlinear systems are shown in this section. First, consider a homogeneous LTI system given as

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -0.5 & -0.1 \end{bmatrix} x \tag{2.25}$$

The above equation is a damped harmonic oscillator where the first state is displacement from reference position and the second state is velocity. An exact solution for this can be found by using Theorem 2.1 with a known initial state. Let the initial state be $x(0) = x_0 = [1 \ -0.5]^\top$ and final time be $t_f = 5$. The discretization time step used is $\Delta t = 0.1$. Figures 2.3 and 2.4 show the magnitude of error between the exact solution and solutions gotten using the described discretization methods. From the figures it is evident that for this example, the state transition matrix method provides the best discretization results as it should provide exact discretization. It is assumed that the error associated with the state transition matrix method is caused by numerical error.

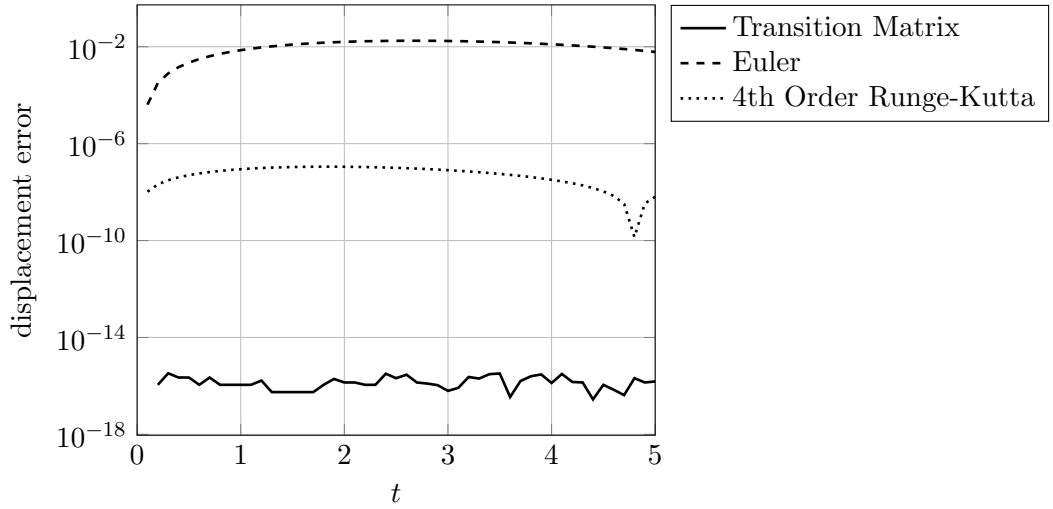


Fig. 2.3: Errors in displacement between exact solution and discretized versions for damped harmonic oscillator.

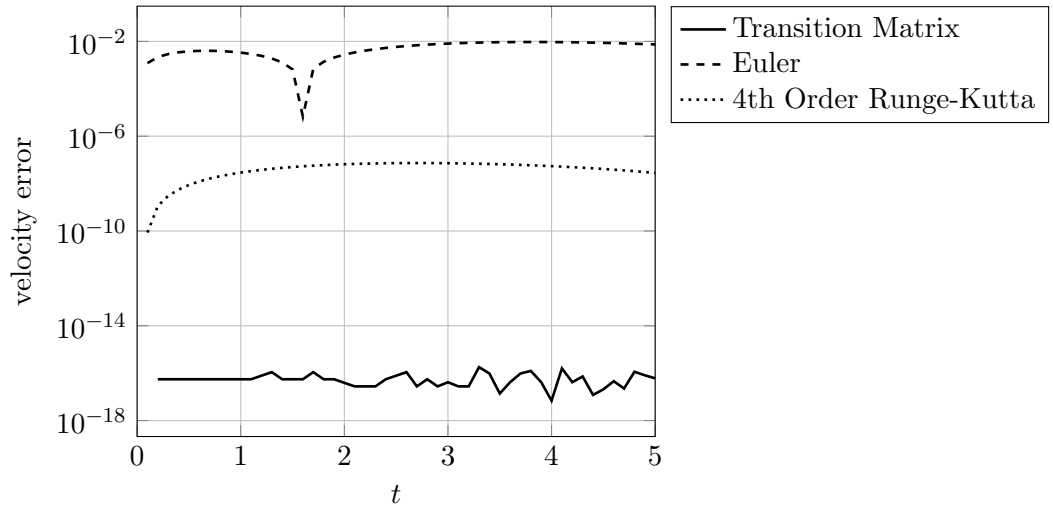


Fig. 2.4: Errors in velocity between exact solution and discretized versions for damped harmonic oscillator.

As a second example, a nonlinear system is considered. The system is given as

$$\dot{x} = \begin{bmatrix} -0.1 & 0 \\ 0 & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ -1 \end{bmatrix} x_1^2 \tag{2.26}$$

To get an exact solution for this, the system is linearized exactly using Koopman theory and defining a new state vector as $y = [x_1\ x_2\ x_1^2]^\top$ [19]. The nonlinear equations are used in the discretization. The initial state is taken as $x(0) = x_0 = [1\ \ -0.5]^\top$ and final time as $t_f = 5$. The discretization time step is $\Delta t = 0.1$ in this example as well. The errors in states between the exact solution and the solutions gotten from the discretized systems are shown in Figures 2.5 and 2.6. As the first state of this system has only linear behavior, the results shown in Figure 2.5 closely follow those of Figures 2.3 and 2.4. However, the second state is affected by the nonlinear term x_1^2 and the error between the exact solution and the discrete approximates is a lot larger. All three discretization methods seem to perform very close to each other with the state transition matrix and the 4th order Runge-Kutta methods performing close to identically with their error trajectories being on top of each other in Figure 2.6.

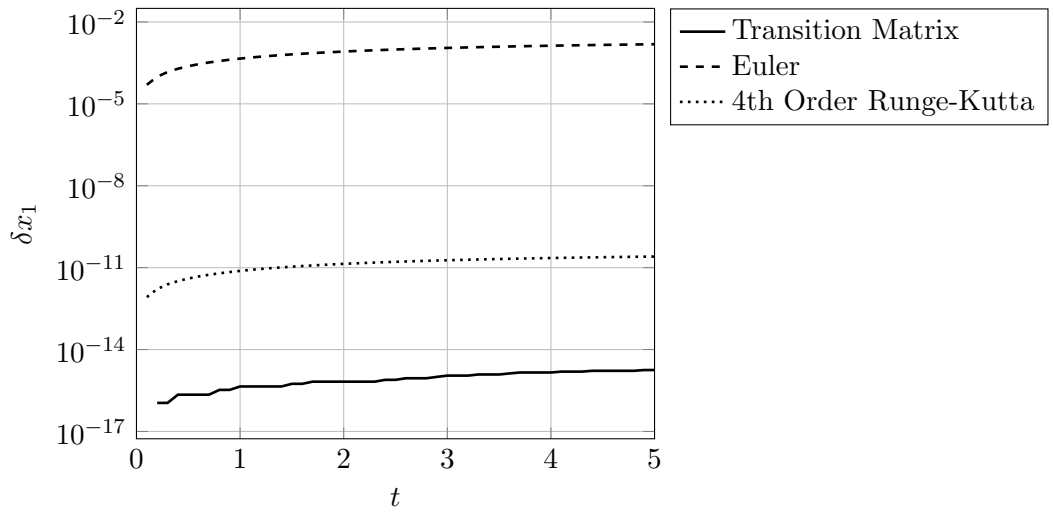


Fig. 2.5: Errors in the first state between exact solution and discretized versions for the nonlinear system.

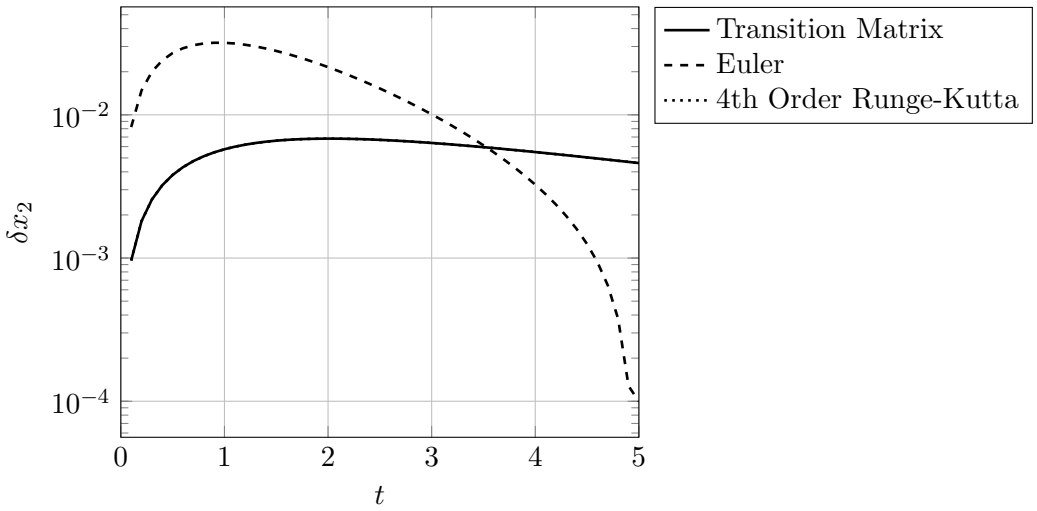


Fig. 2.6: Errors in the second state between exact solution and discretized versions for the nonlinear system.

CHAPTER 3
LINEAR SINGLE-AGENT PROBLEMS WITH ANNULAR CONTROL CONSTRAINT

3.1 Introduction

This chapter analyzes a class of optimal control problems with an annular control constraint. As noted before, the primary motivation for the work in this chapter is real-time optimization. Within the domain of spaceflight, the real-time calculation of trajectories is called guidance. Traditional guidance algorithms, dating back to the 1950s, are simple and analytical in nature. Examples include Apollo lunar descent guidance [63–66], the Iterative Guidance Mode (IGM) for Saturn V ascent [67,68], and Powered Explicit Guidance (PEG) for Shuttle [69, 70]. However, with improved computing power and algorithms, computational guidance laws (especially ones based on convex optimization) are becoming popular [71–77]. Notably, in the last few decades, the computational speed-up due to algorithmic advances exceeds that due to hardware improvements by a factor of two for some problem classes [78]. Within a computational guidance framework, more difficult problems involving systems with practical limitations or constraints can be solved such as the powered rocket landing problem with velocity bounds [31], problems with linear state constraints [79], problems with quadratic state constraints [80], and problems with integer-type control constraints [9, 10].

Even when the problem can be rendered convex, customization is required to accelerate solve times [81–83] to levels sufficient for use onboard radiation-hardened flight processors. These customizations exploit problem structure, sparsity, and memory allocation. The efficacy of this relax and customize approach to the problem of powered rocket landing has been successfully demonstrated through bench testing on a flight computer [84], flight tests [85, 85, 86], and the SpaceX landings [15]. Regarding the flight computer tests [84],

tests were carried out on a 200 MHz RAD750 in the flight software testbed at NASA JPL. Solution times of milliseconds on a 3.4 GHz desktop processor corresponded to solution times of approximately one second on the flight computer with customized code, which is considered appropriate for a real-time guidance, navigation, and control system. During testing and simulation for the above demonstrations, it was observed that the relaxations work even at times that are not the fuel optimal times. This observation and the removal of the assumption on the gradient of the final point motivate the results of this chapter.

To summarize, the three most significant theoretical contributions to lossless convexification involve a controllability assumption and a second assumption at the final point [7, 8, 16]. This assumption precludes convexification for fixed final time problems between fixed points. A key result of this chapter is that for time-invariant systems the second assumption is not needed. The main contributions of the chapter are:

1. Conditions for the standard convexification to hold that are applicable to both free and fixed final time problems (see Theorem 3.1, Corollary 3.2, and Theorem 3.2).

2. A sufficient condition for the standard convexification to hold for all final times between the minimum feasible time and optimal time (see Theorem 3.3).

3. Establishment of controllability as a sufficient condition for solving the general fixed time problem as a sequence of convex programs (see Theorem 3.4).

The remainder of the chapter is organized as follows. Section 3.2 introduces the mathematical notation used in this chapter. Section 3.3 describes the problem of interest, which is an optimal control problem with an annular control constraint. Section 3.4 provides the mathematical analysis of the problem. The main result of the chapter is presented in Section 3.5, which proves controllability to be a sufficient condition for solving fixed final time and fixed end point problems. The section also provides a brief background on reachable sets and minimum time problems. Section 3.6 describes the algorithm used to solve the example problems. Section 3.7 illustrates the application of the new convexification results

to a standard problem in optimal control. A Mars powered descent guidance example is also presented. Section 3.8 concludes the chapter.

3.2 Nomenclature for the Chapter

The following is a partial list of notation used; a function $f \in C^n$ if its first n derivatives exist and are continuous; $\mathbb{R}$ is the set of real numbers; $\mathbb{R}^n$ is the set of real n-tuples; $||v||$ is the 2-norm of v; a condition is said to hold almost everywhere in the interval $[a, b]$, a.e. $[a, b]$, if the set of points in $[a, b]$ where this condition fails to hold is measure zero; the time derivative of a function is denoted with an over-dot, i.e. $dx(t)/dt = \dot{x}(t)$; the boundary of a set S is denoted by ∂S, and the interior of the set by int S. The open ball centered at p with radius r is $B_r(p)$.

3.3 Problem Description

The primary problem of interest is that of minimizing the control effort required to transfer a linear time-invariant system between fixed points subject to an annular control constraint. The problem is described mathematically below in (P3.0).

$$\begin{aligned} \min \quad & J = \textstyle\int_{t_0}^{t_f} \ell(g(u(t)))\, dt \\ \text{subj. to} \quad & \dot{x}(t) = Ax(t) + Bu(t) \\ & x(t_0) = x_0, \;\; x(t_f) = x_f \\ & u(t) \in U_0 = \{\omega : 0 < \rho_1 \leq g(\omega) \leq \rho_2\} \end{aligned} \tag{P3.0}$$

The initial time is t_0. The final time is t_f. The system state $x : [t_0, t_f] \to \mathbb{R}^n$ belongs to the set of absolutely continuous functions. The control $u : [t_0, t_f] \to U_0 \subset \mathbb{R}^m$ belongs to the set of bounded measurable functions with $\rho_1, \rho_2 \in \mathbb{R}$. The objective is to minimize the control effort given by $\ell : [\rho_1, \rho_2] \to \mathbb{R}$ where $g : \mathbb{R}^m \to \mathbb{R}$. The system dynamics are described by the linear differential equations where A and B are constant matrices. It is assumed that g is convex and that ℓ is convex, strictly increasing, and positive on its domain. The objective function is then convex. For the moment, the final time can be free or fixed. The least

final time for which the problem is feasible is called the minimum feasible time. The final time for which (P3.0) achieves a global minimum is called the optimal time. The primary challenge in solving (P3.0) is the non-convex control constraint U_0.

The problem (P3.1) below is the standard relaxation of (P3.0). It is obtained by introducing a new variable Γ and reformulating the control constraint.

$$\begin{aligned} \min \quad & J = \textstyle\int_{t_0}^{t_f} \ell(\Gamma(t))\, dt \\ \text{subj. to} \quad & \dot{x}(t) = Ax(t) + Bu(t) \\ & x(t_0) = x_0, \;\; x(t_f) = x_f \\ & (u(t), \Gamma(t)) \in U_1 = \{(\omega, \Omega) : \rho_1 \leq \Omega \leq \rho_2, \dots \\ & \qquad\qquad \text{and } g(\omega) \leq \Omega\}. \end{aligned} \tag{P3.1}$$

The control set U_1 is a convex relaxation of U_0, but because it is a relaxation, solutions of (P3.1) may not be feasible for (P3.0). If it so happens that $g(u(t)) = \Gamma(t)$ almost everywhere, then solutions of (P3.1) are solutions of (P3.0). This leads to the definition of *lossless convexification*.

Definition 3.1. (P3.1) *is a lossless convexification of* (P3.0) *if for every* (u, Γ) *solving* (P3.1) *it follows that* $g(u(t)) = \Gamma(t)$ *almost everywhere.*

From this definition and inspection of the problems follows a necessary and sufficient condition, albeit one that is not checkable a priori.

Lemma 3.1. *Let* (u, Γ) *be a solution of* (P3.1)*. Lossless convexification holds if and only if* $g(u(t)) \geq \rho_1$ *almost everywhere. Equivalently, it fails if and only if* $g(u(t)) < \rho_1$ *on a set of positive measure.*

Proof. To prove the forward implication, suppose convexification holds. Then $g(u(t)) = \Gamma(t) \geq \rho_1$. To prove the backward implication, suppose $g(u(t)) \geq \rho_1$. Then the objective is minimized by picking $\Gamma(t)$ as small as possible, i.e., $\Gamma(t) = g(u(t))$. □

This leads to the definition of the *hairline case*.

Definition 3.2. *Let* (u, Γ) *be a solution of* (P3.1). *The hairline case is when* $g(u(t)) = \rho_1$ *almost everywhere.*

Lossless convexification holds in the hairline case according to the above theorem. However, any change in problem data that causes $g(u(t))$ to decrease on a set of positive measure will make convexification fail.

3.4 Mathematical Results

Problem (P3.1) is now analyzed mathematically leading to some new results. The necessary conditions of Pontryagin [26] state that if (x, u, Γ) is optimal for (P3.1) then there exist a scalar $p_0 \in \{0, 1\}$, a function $p : [t_0, t_f] \to \mathbb{R}^n$, and a scalar ν such that the following hold.

$$0 \neq (p_0, p) \tag{3.1}$$

$$H(t) = p_0 \ell(\Gamma(t)) + p(t)^T (Ax(t) + Bu(t)) = -\nu \tag{3.2}$$

$$\dot{p}(t) = -A^T p(t) \tag{3.3}$$

$$u(t) \in \arg\min_{\omega} p^T B\omega \quad \text{s.t. } g(\omega) \leq \Gamma \tag{3.4}$$

$$\Gamma(t) \in \arg\min_{\Omega} p_0 \ell(\Omega) \ \text{ s.t. } \rho_1 \leq \Omega \leq \rho_2 \ , \ g(u) \leq \Omega \tag{3.5}$$

The first condition is called the non-triviality condition. The second condition states that the Hamiltonian H is a constant. It is zero when the final time is free and possibly non-zero when the final time is fixed. The third condition is the adjoint system. The fourth and fifth conditions are the pointwise minimum conditions.

A sufficient condition for lossless convexification is stated below. In general, it cannot be checked a priori.

Lemma 3.2. *If* $p^T(t)B$ *is non-zero almost everywhere, then lossless convexification holds.*

Proof. If $p^T(t)B$ is non-zero, then the optimal point is on the boundary of the feasible set in (3.4), i.e., $g(u(t)) = \Gamma(t)$ almost everywhere. □

In other words, convexification holds if the solution is non-singular. Conversely, if convexification fails, then the solution is singular. However, if the solution is singular, convexification may or may not hold. This leads to another result for free final time problems.

Corollary 3.1. *If (A, B) is controllable and the final time is free, then lossless convexification holds.*

Proof. Suppose convexification fails such that $p^T(t)B = 0$ on a set of positive measure. Because p is analytic and (A, B) is controllable, $p(t) = 0$ on this set [7]. Because the final time is free, the Hamiltonian must be zero, which can only happen if $p_0 = 0$ since ℓ is positive. This violates the non-triviality condition. Therefore, the problem is non-singular and lossless convexification holds. □

The following sufficient condition is true for all cases except the hairline case. This condition and its corollary are new results.

Theorem 3.1. *Let (u, Γ) be a solution of* (P3.1)*. If (A, B) is controllable and $g(u(t)) > \rho_1$ on a set of positive measure, then convexification holds.*

Proof. There are two cases.

Case 1) Suppose the final time is free. Then the above corollary applies and lossless convexification holds.

Case 2) Suppose the final time is fixed and convexification fails. Then on a set of positive measure, $p^T(t)B = 0$ and $p(t) = 0$ because of controllability. The Hamiltonian reduces to $H(t) = \ell(\Gamma(t)) = \textit{constant}$ almost everywhere. From Lemma 3.1, it is known that there is some time where $g(u(t)) < \rho_1$. Minimizing the Hamiltonian implies $\Gamma(t) = \rho_1$ almost everywhere since ℓ is strictly increasing. Thus, $g(u(t)) \leq \rho_1$ almost everywhere, which is a contradiction. □

This proof leads to a very useful corollary.

Corollary 3.2. *Let (u, Γ) be a solution of* (P3.1)*. Suppose (A, B) is controllable. Lossless convexification fails if and only if $g(u(t)) \leq \rho_1$ almost everywhere and $g(u(t)) < \rho_1$ on a set of positive measure.*

For free final time problems, there is an a priori checkable sufficient condition: controllability. For fixed final time problems, another checkable condition is now derived from this corollary for the common situation where $g(u(t)) = ||u(t)||$. Recall that the state equation can be written as

$$x_f - \Phi(t_f, t_0)x_0 = \int_{t_0}^{t_f} \Phi(t_f, t)Bu(t)dt, \tag{3.6}$$

where Φ is the state transition matrix, or matrix exponential. Norming each side and assuming convexification fails, i.e., $||u(t)|| \leq \rho_1$, then

$$\begin{aligned} ||x_f - \Phi(t_f, t_0)x_0|| &= || \int_{t_0}^{t_f} \Phi(t_f, t)Bu(t)dt|| && (3.7)\\ &\leq \int_{t_0}^{t_f} ||\Phi(t_f, t)Bu(t)||dt && (3.8)\\ &\leq \int_{t_0}^{t_f} ||\Phi(t_f, t)B||\, ||u(t)||dt && (3.9)\\ &\leq \rho_1 \int_{t_0}^{t_f} ||\Phi(t_f, t)B||dt. && (3.10) \end{aligned}$$

Define the following

$$\begin{aligned} F(t_f) &= ||x_f - \Phi(t_f, t_0)x_0||,\\ G(t_f) &= \rho_1 \int_{t_0}^{t_f} ||\Phi(t_f, t)B||dt, \end{aligned} \tag{3.11}$$

so that a checkable condition for the fixed final time problem can be stated.

Theorem 3.2. *Suppose that (A, B) is controllable and $g(u(t)) = ||u(t)||$. If $F(t_f) > G(t_f)$, then convexification holds.*

Proof. The proof follows from the analysis above and Corollary 3.2. □

Conceptually, convexification is guaranteed when the boundary conditions belong to the open, non-convex set whose complement is the closed ellipsoid defined by F and G. As an example, consider problems that terminate at the origin. The ellipsoid is then centered at the origin and given by

$$x_0^T \Phi^T(t_f, t_0)\Phi(t_f, t_0)x_0 \leq \left(\rho_1 \int_{t_0}^{t_f} ||\Phi(t_f, t)B||dt\right)^2, \tag{3.12}$$

which can easily be calculated for given problem data.

It is now proven that for fixed final time problems, convexification holds for all final times between the minimum feasible time and the optimal time.

Theorem 3.3. *Suppose (A, B) is controllable. Let t_1 be the minimum feasible time and t_2 be the optimal time. If the optimal cost $J(t_f) = \int_{t_0}^{t_f} \ell(g(u(t)))dt$ decreases strictly on $[t_1, t_2]$, then convexification holds for any $t_f \in [t_1, t_2]$.*

Proof. Because the objective is strictly decreasing, one can deduce that $\nu \geq 0$ and that the Hamiltonian is a non-positive constant [87]. Suppose convexification fails such that $p^T(t)B$ is zero on a set of positive measure. Because (A, B) is controllable, $p(t) = 0$ on this set. The Hamiltonian then reduces to

$$p_0 \ell(\Gamma) + \nu = 0. \tag{3.13}$$

The scalar p_0 is non-negative and $\ell(\Gamma)$ is positive such that the both terms are non-negative. Equality holds only when $p_0 = \nu = 0$, but this violates the non-triviality condition. □

3.5 Main Result

The sufficient condition in Theorem 3.2 is quite conservative because it involves several approximations. Also, Theorem 3.3 is only applicable to final times between the minimum feasible time and the optimal time. In this section, is is shown that controllability is a sufficient condition to solve (P3.0) as a sequence of convex programs for any fixed final time. To do so, several facts related to reachable sets and minimum time problems are needed.

Define the point $w_f := x_f - \Phi(t_f, t_0)x_0$ and the reachable set as

$$\mathcal{R}(t_0, t_f, U) := \{\textstyle\int_{t_0}^{t_f} \Phi(t_f, t)Bu(t)dt,\ \forall u(t) \in U\}, \tag{3.14}$$

where U is a compact set of all admissible controls. It is clear that $w_f \in \mathcal{R}(t_0, t_f, U)$ is required for any optimal control problem, no matter the objective, to be feasible. It is known that the reachable sets are compact, convex, and continuous in both time arguments (see Lemmas 12.1 and 12.2 in [88]).

Definition 3.3. *A reachable set is expanding if for all* $t_1 < t_2$, $\mathcal{R}(t_0, t_1, U) \subset \text{int } \mathcal{R}(t_0, t_2, U)$.

Lemma 3.3. *Suppose* U *is compact, convex, and* $0 \in \text{int } U$. *The reachable set is expanding if and only if* (A, B) *is controllable. If the reachable set is expanding, then* t_f *is the minimum time if and only if* $w_f \in \partial\mathcal{R}(t_0, t_f, U)$.

Proof. See Corollary 17.1 and Theorem 17.3 of [88]. Also see Theorem 1 on page 301 of [89]. □

An illustration of the expanding reachable set and its connection with the minimum time is given in Figure 3.1.

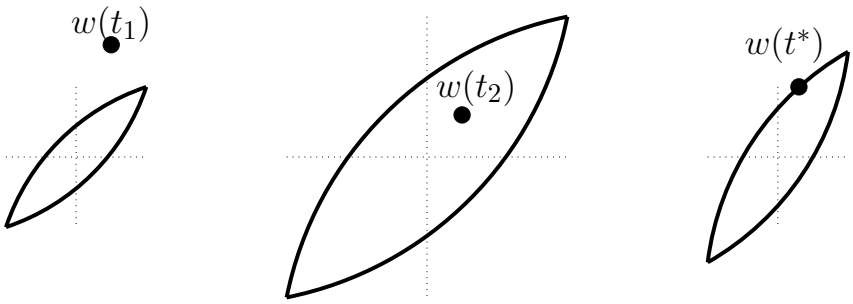


Fig. 3.1: (a) At time t_1, the problem is infeasible as the point, w, is outside the reachable set. (b) At some time $t_2 > t_1$, the point, w, is in the interior of the reachable set and the problem is feasible. (c) At the minimum time, t^*, such that $t_1 < t^* < t_2$, the point, w, is on the boundary of the reachable set.

The following minimum time problem is studied now.

$$\begin{aligned} \min \quad & J = \textstyle\int_{t_0}^{t_f} 1 \, dt && \text{(P3.2)} \\ \text{subj. to} \quad & \dot{x}(t) = Ax(t) + Bu(t) && \text{(F3.1)} \\ & x(t_0) = x_0, \;\; x(t_f) = x_f && \text{(F3.2)} \\ & u(t) \in U_2 = \{\omega : g(\omega) \leq \rho_1\} && \text{(F3.3)} \end{aligned}$$

The optimal control problem is referred as (P3.2) and the problem of only satisfying the constraints (F3.1)-(F3.3) for a fixed final time as the feasibility problem, which is convex. As before, the necessary conditions state that if (x, u) is optimal for (P3.2) then there exist a scalar $p_0 \in \{0, 1\}$ and a function $p : [t_0, t_f] \to \mathbb{R}^n$ such that the following hold.

$$0 \neq (p_0, p) \tag{3.15}$$

$$0 = p_0 + p(t)^T (Ax(t) + Bu(t)) \tag{3.16}$$

$$\dot{p}(t) = -A^T p(t) \tag{3.17}$$

$$u(t) \in \arg\min_{\omega \in U_2} p(t)^T B\omega \tag{3.18}$$

Lemma 3.4. *Let u be a solution of* (P3.2)*. If (A, B) is controllable, then $g(u(t)) = \rho_1$ almost everywhere.*

Proof. If (A, B) is controllable, then $p^T(t)B \neq 0$ almost everywhere (see the proof of Corollary 3.1) and the optimal point is on the boundary of the feasible set, i.e., $g(u(t)) = \rho_1$ almost everywhere. □

With these facts about reachable sets and minimum time problems, the main theorem regarding problem (P3.0) can be stated and proved.

Theorem 3.4. *Consider* (P3.0) *and assume fixed final time. Assume* (P3.1) *is not a lossless convexification of* (P3.0)*. If (A, B) is controllable, then there exists an optimal control u for* (P3.0) *such that $g(u(t)) = \rho_1$ almost everywhere obtained by solving a sequence of convex problems.*

Proof. By assumption, (P3.1) is not a lossless convexification for (P3.0) and (A, B) is controllable. It follows from Corollary 3.2 that an optimal control u for (P3.0) satisfies $g(u(t)) = \rho_1$ almost everywhere. Because U_2 is compact, convex, and $0 \in \text{int } U_2$, Lemma 3.3 indicates that reachable sets generated by U_2 are continuous, compact, convex, and expanding such that any point on the boundary of a reachable set must be reached in minimum time. Two cases are now considered.

Case 1:

If the final time is the minimum final time, then $w_f \in \partial\mathcal{R}(t_0, t_f, U_2)$ and Lemma 3.4 implies that any feasible control u satisfies $g(u(t)) = \rho_1$ almost everywhere. Therefore, one must solve a single instance of the convex feasibility problem (F3.1)-(F3.3) on $[t_0, t_f]$.

Case 2:

If the final time is greater than the minimum time, then $w_f \in \text{int } \mathcal{R}(t_0, t_f, U_2)$ since the reachable set is expanding. By definition of interior, $\exists \epsilon > 0$ such that $\forall w_1 \in B_\epsilon(0)$ the point $w_f - w_1 \in \text{int } \mathcal{R}(t_0, t_f, U_2)$. From closedness, convexity, and continuity (see Lemma 12.3 of [88]), $\exists \delta > 0$ such that $\forall t_1 \in B_\delta(t_0)$ the point $w_f - w_1 \in \text{int } \mathcal{R}(t_1, t_f, U_2)$.

Consider a function $v_1 : [t_0, t_1] \to \partial U_2$ and let w_1 and t_1 satisfy

$$w_1 = \int_{t_0}^{t_1} \Phi^{-1}(t, t_1) B v_1(t) dt, \tag{3.19}$$

which can always be done by making t_1 sufficiently close to t_0. That is, $w_1 \in \mathcal{R}(t_0, t_1, \partial U_2)$. If t_1 is increased to t_f and $w_f - w_1$ never leaves the reachable set, then v_1 is an optimal control for (P3.0) since $\forall t \in [t_0, t_f]$, $g(v_1(t)) = \rho_1$, which results in the least possible objective value.

If the point $w_f - w_1$ does leave the reachable set, then at the time t_1 when the point is on the boundary of the reachable set, there exists a minimum time control to $w_f - w_1$ denoted $u_1 : [t_1, t_f] \to \partial U_2$. It follows that the control $u : [t_0, t_f] \to \partial U_2$ given by

$$u(t) = \begin{cases} v_1(t), & t_0 \leq t < t_1 \\ u_1(t), & t_1 \leq t \leq t_f \end{cases} \tag{3.20}$$

is optimal for (P3.0) since $\forall t \in [t_0, t_f]$, $g(u(t)) = \rho_1$, which results in the least possible objective value.

By incrementally increasing t_1, introducing a perturbing function $v_1 : [t_0, t_1] \to \partial U_2$, solving the convex feasibility problem (F3.1)-(F3.3) from the perturbed point to the final point for u_1, and checking if $u_1(t) \in \partial U_2$, the optimal control u for (P3.0) is obtained. $\square$

The proof of Theorem 3.4 is constructive and indicates that a line search for t_1, selection of a perturbing control v_1, and solution of a convex constrained problem for u_1 yields the optimal control. This is similar to free final time problems where a line search for t_f is required. Lastly, since any v_1 satisfying $g(v_1(t)) = \rho_1$ works, it is clear that optimal solutions are non-unique. A specific algorithm for finding an optimal control is given in the following section.

3.6 Solution Procedure

This section summarizes the algorithm constructed in the proof of Theorem 3.4 and method used to solve the feasibility problem (F3.1)-(F3.3). Given a control u on $[t_0, t_f]$, a measure of how close its magnitude is to ρ_1 is given by the following formula.

$$E(u, t_0, t_f) = \int_{t_0}^{t_f} ||g(u(t)) - \rho_1||\, dt \tag{3.21}$$

The line search algorithm to solve (P3.0) when Theorem 3.4 applies is given in Algorithm 1.

To numerically solve the convex feasibility problem (F3.1)-(F3.3), it is discretized and constraints are enforced at the nodes resulting in a second-order cone program that can be solved using interior-point methods. A simple discretization method is used, and is summarized below.

The time domain $[t_0, t_f]$ is uniformly discretized into $N+1$ nodes separated by Δt. The states at time t_i are denoted by $x[i]$ and they exist at all nodes $i = 1, ..., N+1$. The controls at time t_i are denoted by $u[i]$ and they exist at nodes $i = 1, ..., N$. The controls are held

Algorithm 1 Algorithm to solve fixed final time problems with annular control constrains

Initialization:
Choose a $v \in \mathbb{R}^m$ such that $g(v) = \rho_1$.
Choose a $t_1 \in (t_0, t_f)$.
Choose a tolerance $\epsilon > 0$.
Set $t_{min} = t_0$ and $t_{max} = t_f$.
Loop:
loop
  Define $v_1(t) = v$ on $[t_0, t_1]$.
  Compute $x(t_1)$ generated by v_1.
  Solve (F3.1)-(F3.3) from $x(t_1)$ to x_f to find u_1 on $[t_1, t_f]$.
  if (F3.1)-(F3.3) is infeasible **then**
    Set $t_{max} = t_1$ and $t_1 = \frac{1}{2}(t_{min} + t_{max})$.
  else if (F3.1)-(F3.3) is feasible and $E(u_1, t_1, t_f) > \epsilon$ **then**
    Set $t_{min} = t_1$ and $t_1 = \frac{1}{2}(t_{min} + t_{max})$.
  else
    return v_1 and u_1 as an optimal control
  end if
end loop

constant over every interval. The system dynamics are discretized using the fundamental matrix resulting in

$$x[i+1] = A_d x[i] + B_d u[i], \quad i = 1, ..., N. \tag{3.22}$$

The discrete system matrices A_d and B_d are given by

$$A_d = e^{A\Delta t}, \quad B_d = \int_0^{\Delta t} e^{A\tau} B d\tau. \tag{3.23}$$

The boundary conditions are enforced at the initial and final nodes.

$$x[1] = x_0, \quad x[N+1] = x_f \tag{3.24}$$

The control constraints are enforced at nodes 1 to N.

$$|| u[i] || \leq \rho_1, \quad \forall\, i = 1, ..., N \tag{3.25}$$

Equations (3.22)-(3.25) represent a finite-dimensional second-order cone program that can be solved using commercially available solvers.

3.7 Examples

In this section, the algorithm is demonstrated on a simple double integrator and a harmonic oscillator. A Mars powered descent guidance example is then solved in a sample-and-hold scheme to emulate a real guidance system. Problems are solved using the Gurobi [90] solver with a MATLAB interface [91].

3.7.1 Double Integrator

Consider the following problem with double integrator dynamics and an annular control constraint. The problem fits into the definition of (P3.0) and is mathematically described below.

$$
\begin{aligned}
\min \quad & J = \int_{t_0}^{t_f} ||u(t)|| \, dt \\
\text{subj. to} \quad & \dot{x}_1(t) = x_2(t) \\
& \dot{x}_2(t) = u(t) \\
& x_1(t_0) = 1, \;\; x_1(t_f) = 0 \\
& x_2(t_0) = 1, \;\; x_2(t_f) = 0 \\
& 1 \leq ||u(t)|| \leq 6
\end{aligned}
$$

Recognizing $g(u(t)) = ||u(t)||$ and $\ell(g(u(t)) = g(u(t))$, the standard relaxation (P3.1) can be stated.

$$\begin{aligned}
\min \quad & J = \int_{t_0}^{t_f} \Gamma(t)\, dt \\
\text{subj. to} \quad & \dot{x}_1(t) = x_2(t) \\
& \dot{x}_2(t) = u(t) \\
& x_1(t_0) = 1, \;\; x_1(t_f) = 0 \\
& x_2(t_0) = 1, \;\; x_2(t_f) = 0 \\
& 1 \leq \Gamma(t) \leq 6 \\
& ||u(t)|| \leq \Gamma(t)
\end{aligned}$$

According to Theorem 3.2, the above convexification will hold if $F(t_f) > G(t_f)$, where $F(t_f)$ and $G(t_f)$ are defined by (3.11). Figure 3.2 shows the the difference $F(t_f) - G(t_f)$ as a function of the final time t_f. By solving the problem, it was determined that the minimum feasible time is approximately 1.016 time units and the optimal time is approximately 2.1 time units. The difference is positive between the minimum feasible and optimal times and negative for times greater than the optimal time. Consistent with Theorems 3.2 and 3.3, numerical tests demonstrate that the standard convexification holds when the difference is positive.

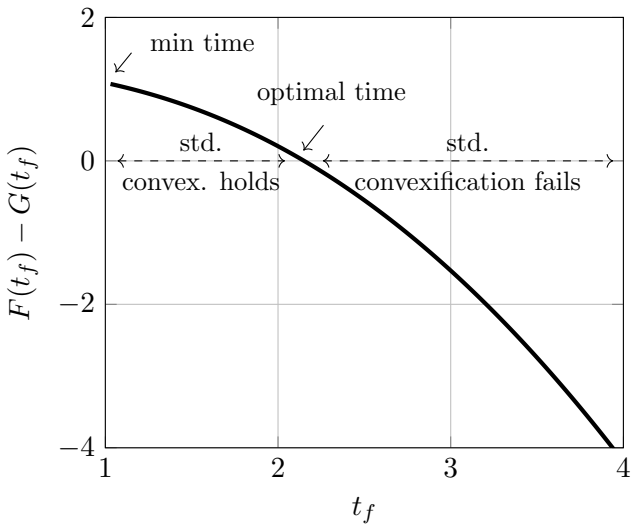


Fig. 3.2: The quantity $F - G$ is positive and convexification holds for final times between the minimum feasible time and the optimal time of about 2.1 time units. For larger times, the difference is negative and convexification fails.

For a final time of $t_f = 5$ time units, standard convexification fails. With $t_f = 5$, Algorithm 1 is used to solve the problem. According to Theorem 3.4, since the system is controllable, the optimal control u satisfies $||u(t)|| = 1$ almost everywhere. To demonstrate non-uniqueness, two solutions are generated using the perturbing functions $v_1(t) = 1$ on $[0, 0.42]$ and $v_1(t) = -1$ on $[0, 2.38]$, respectively. In each case, the feasibility problem (F3.1)-(F3.3) is solved to find the optimal control after the perturbing period. Figures 3.3-3.5 show the states, controls, and reachable set.

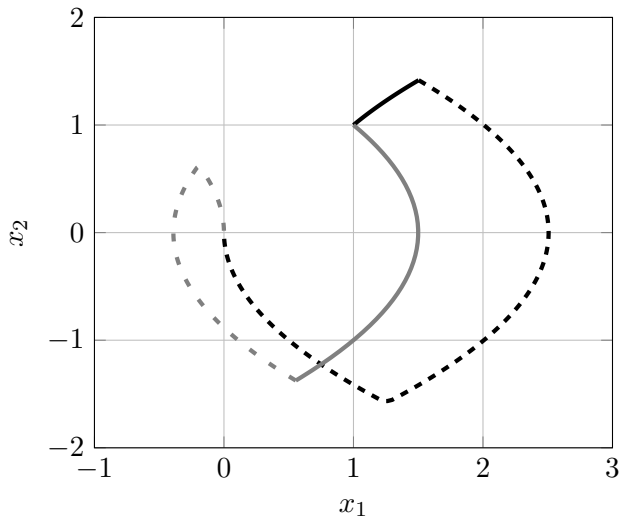


Fig. 3.3: State trajectories in phase plane. The solid black curve corresponds to the $v_1(t) = 1$ solution. The gray curve corresponds to the $v_1(t) = -1$ solution. After the perturbing periods, a minimum time control u_1 is computed that satisfies $||u_1(t)|| = 1$. The state then follows the so-called switching curve to reach the origin.

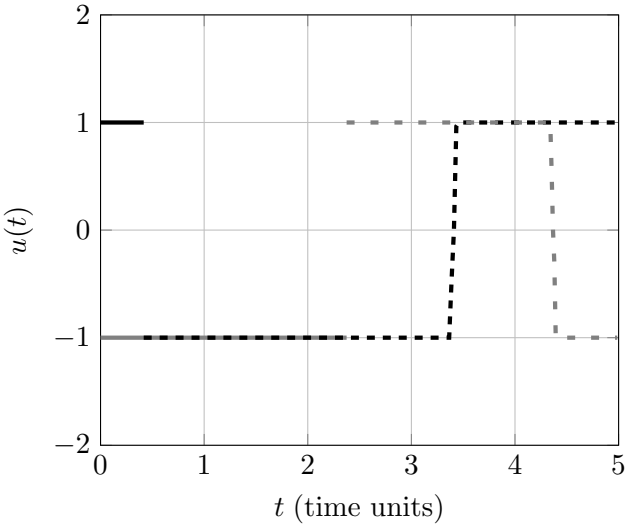


Fig. 3.4: The solid lines correspond to the perturbing controls v_1. The dashed lines correspond to the minimum time control u_1. Color scheme is the same as Figure 3.3.

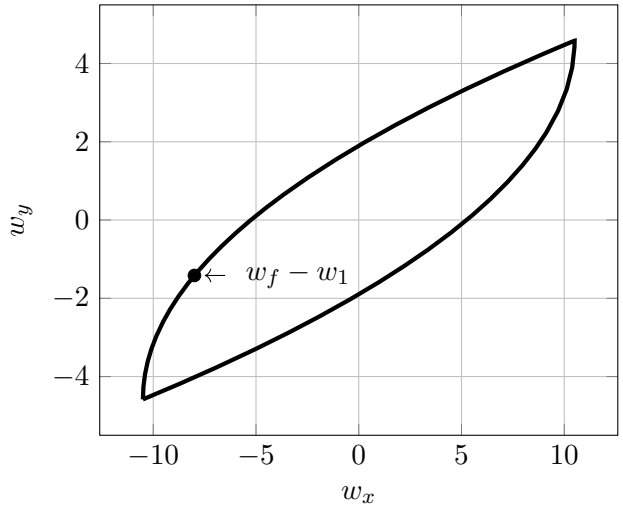


Fig. 3.5: Reachable set, $\mathcal{R}(t_1, t_f, U_2)$, for $t_1 = 0.42$ and $t_f = 5$ time units ($v_1(t) = 1$ case). As required, the point $w_f - w_1$ lies on the boundary of the set.

For both cases, Algorithm 1 was initialized with a guess of $t_1 = 2.5$. It turns out that both cases required the solution of seven second-order cone programs. On a laptop with 2.2 GHz processor, each program required about 0.2 seconds to solve for a total solution time of about 1.4 seconds.

3.7.2 Harmonic Oscillator

A fuel optimal problem with the dynamics of a harmonic oscillator is now considered. The problem is mathematically described below.

$$
\begin{aligned}
\min \quad & J = \textstyle\int_{t_0}^{t_f} ||u(t)||\, dt \\
\text{subj. to} \quad & \dot{x}_1(t) = x_2(t) \\
& \dot{x}_2(t) = -x_1(t) + u(t) \\
& x_1(t_0) = 2, \;\; x_1(t_f) = 0 \\
& x_2(t_0) = 0, \;\; x_2(t_f) = 0 \\
& 1 \leq ||u(t)|| \leq 3
\end{aligned}
$$

Below is the standard, convex relaxation to the problem.

$$
\begin{aligned}
\min \quad & J = \int_{t_0}^{t_f} \Gamma(t)\, dt \\
\text{subj. to} \quad & \dot{x}_1(t) = x_2(t) \\
& \dot{x}_2(t) = -x_1(t) + u(t) \\
& x_1(t_0) = 2, \;\; x_1(t_f) = 0 \\
& x_2(t_0) = 0, \;\; x_2(t_f) = 0 \\
& 1 \leq \Gamma(t) \leq 3 \\
& ||u(t)|| \leq \Gamma(t)
\end{aligned}
$$

Figure 3.6 shows the variation of the difference $F(t_f) - G(t_f)$ with the final time t_f. The minimum feasible time for the problem is approximately 1.51 time units and the difference is non-negative until $t_f \approx 2$ time units. However, standard convexification fails only when t_f is increased beyond 2 time units, which is the fuel optimal time for the problem.

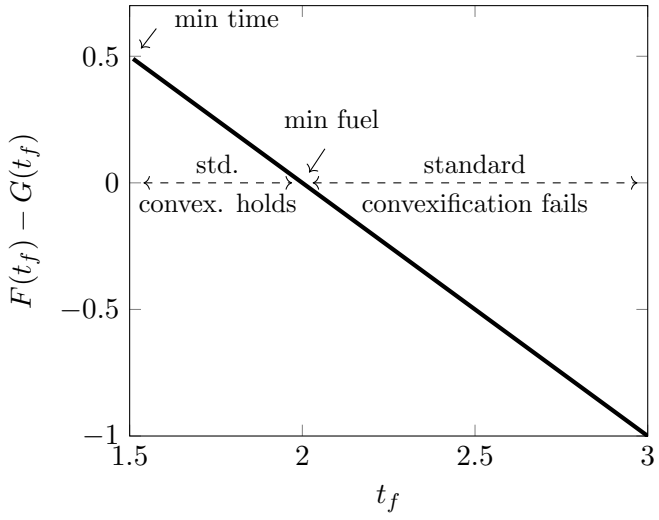


Fig. 3.6: The quantity $F - G$ is positive for final times between the minimum feasible time and $t_f \approx 2$ time units. The fuel optimal time is 2 time units. For $t_f > 2$ time units, the quantity is negative. Standard convexification holds for all final times between the minimum feasible time and fuel optimal time, after which it fails.

For a final time of $t_f = 5$ time units, standard convexification fails. Like before, $t_f = 5$ time units and it is shown that this fixed time problem can be solved as a sequence of convex programs. According to Theorem 3.4, since the system is controllable, there exists a control $u(t)$ such that $||u(t)|| = 1$ almost everywhere.

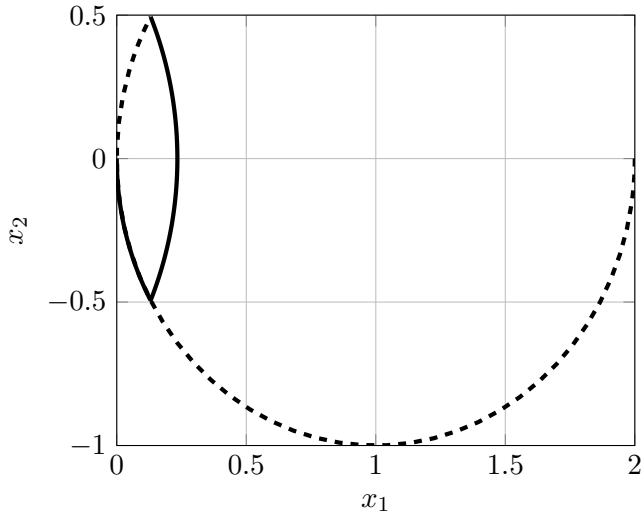


Fig. 3.7: State trajectory in phase plane. The initial state is perturbed using a arbitrary control $v_1(t)$ that satisfies $||v_1(t)|| = 1$ for a duration $t_1 = 3.65$ time units (indicated by the dashed curve). At the end of t_1 time units, a minimum time control, $u_1(t)$ is achieved that satisfies $||u_1(t)|| = 1$. The state then follows the switching curve to reach the origin (indicated by the solid curve).

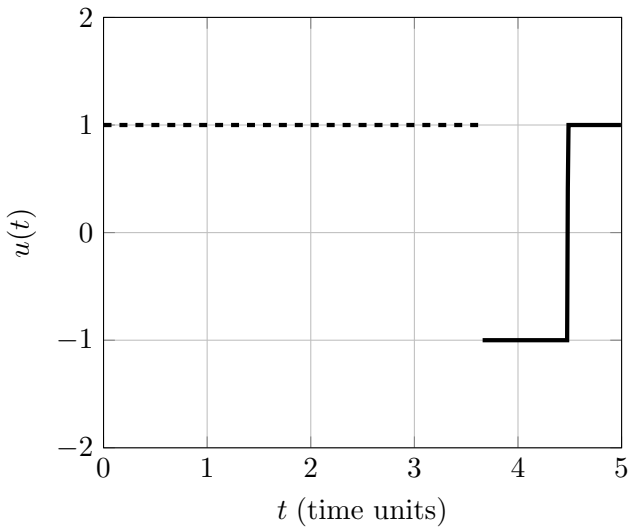


Fig. 3.8: Control trajectory. The dashed line corresponds to the arbitrary control $v_1(t)$ that satisfies $||v_1(t)|| = 1$. The solid line corresponds to the minimum time control, $u_1(t)$, which switches from -1 to +1.

It is found that on perturbing the initial state with a control $v_1(t)$ that satisfies $||v_1(t)|| = 1$ for a duration of ≈ 3.65 time units, convexification works, and we have an optimal control $u(t)$ that satisfies $||u(t)|| = 1$ almost everywhere. Fig 3.7 shows the trajectory in state plane and Figure 3.8 shows the control profile. Figure 3.9 shows that the point $w_f - w_1$ lies on the boundary of the reachable set $\mathcal{R}(t_1, t_f)$ at time $t_1 \approx 3.65$ time units.

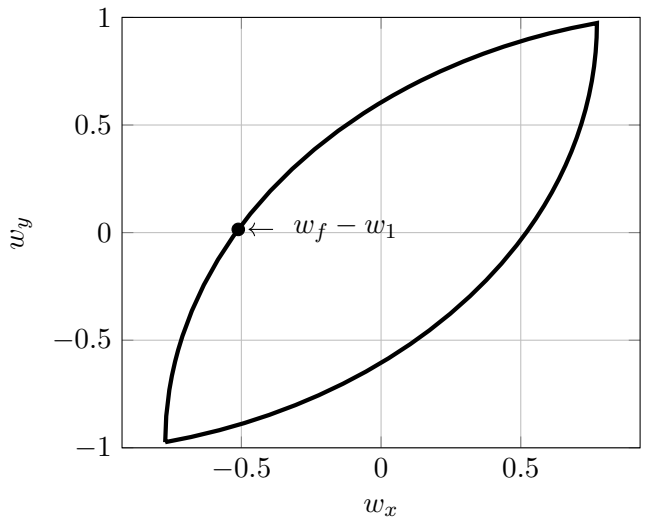


Fig. 3.9: Reachable set, $\mathcal{R}(t_1, t_f)$, for $t_1 = 3.65$ and $t_f = 5$ time units. As required, the point $w_f - w_1$ lies on the boundary of the set.

3.7.3 Mars Powered Descent Guidance

As a final example, consider a Mars powered descent guidance problem. Because the landing mission starts at a low altitude, it is assumed that the gravitational field is uniform. Aerodynamic forces are not considered since they are negligible compared to the propulsive and gravitational forces during powered descent. An illustration of the problem is given in [5].

The dynamics of the spacecraft are given by the following equations

$$\ddot{r}(t) = g + \frac{T(t)}{m(t)}, \tag{3.26}$$

$$\dot{m}(t) = -\alpha ||T(t)||, \tag{3.27}$$

where $r(t) \in \mathbb{R}^3$ is the position vector of the spacecraft relative to the target, $g \in \mathbb{R}^3$ is the constant gravitational vector, $T(t) \in \mathbb{R}^3$ is the net thrust vector, $m(t) \in \mathbb{R}$ is the spacecraft mass, and $\alpha \in \mathbb{R}$ is a positive constant that defines the fuel consumption rate. The net thrust is bounded by

$$0 < \rho_1 \leq ||T(t)|| \leq \rho_2, \tag{3.28}$$

which defines a non-convex set of feasible controls. Additional nonlinearities are present because of the nonlinearity of (3.26) and (3.27). To remove these nonlinearities, the following variable transformations are introduced.

$$u(t) := \frac{T(t)}{m(t)},\ \sigma := \frac{||T(t)||}{m(t)},\ z(t) := \ln(m(t)) \tag{3.29}$$

Upon defining the quantities

$$\mu_1(t) = \rho_1 e^{-\tilde{z}(t)} \quad \text{and} \quad \mu_2(t) = \rho_2 e^{-\tilde{z}(t)} \tag{3.30}$$

where

$$\tilde{z}(t) = \begin{cases} \ln(m_{wet} - \alpha\rho_2 t), & m_{wet} - \alpha\rho_2 t \geq m_{dry} \\ \ln(m_{dry}), & \text{otherwise} \end{cases}, \tag{3.31}$$

a second-order cone program can be formed that approximates the problem of interest [5].

$$\begin{aligned}
\min \quad & J = \textstyle\int_0^{t_f} \sigma(\tau)\, d\tau \\
\text{subj. to} \quad & \dot{r}(t) = v(t) \\
& \dot{v}(t) = u(t) + g \\
& \dot{z}(t) = -\alpha\sigma(t) \\
& ||u(t)|| \leq \sigma(t) \\
& \mu_1(t)\Big[1 - \big[z(t) - \tilde{z}(t)\big] + \tfrac{1}{2}\big[z(t) - \tilde{z}(t)\big]^2\Big] \\
& \qquad \leq \sigma(t) \leq \mu_2(t)\Big[1 - \big[z(t) - \tilde{z}(t)\big]\Big] \\
& \ln(m_{wet} - \alpha\rho_2 t) \leq z(t) \leq \ln(m_{wet} - \alpha\rho_1 t) \\
& m(0) = m_{wet},\ \ r(0) = r_0,\ \ \dot{r}(0) = \dot{r}_0 \\
& r(t_f) = \dot{r}(t_f) = 0
\end{aligned}$$

The following parameters are used to solve the problem.

$$
\begin{aligned}
r_0 &= [1500, 0, 2000]^T \text{ m} \\
\dot{r}_0 &= [-75, 0, 100]^T \text{ m/s} \\
g &= [-3.7114, 0, 0]^T \text{ m/s}^2 \\
m_{dry} &= 1505 \text{ kg}, m_{wet} = 2110 \text{ kg} \\
I_{sp} &= 225 \text{ s}, \alpha = 5.09 \cdot 10^{-4} \text{ s/m} \\
\rho_1 &= 13.151 \text{ kN}, \rho_2 = 19.727 \text{ kN}
\end{aligned}
\tag{3.32}
$$

Through numerical tests, the problem has a minimum feasible time of about 45 seconds and an optimal time of about 52 seconds. Tests indicate that the standard convexification holds for final times between 45 seconds and 72 seconds, and it fails thereafter. To demonstrate our results, the final time is set at $t_f = 90$ seconds and it is known that the optimal solution will have thrust magnitude equal to ρ_1 almost everywhere.

To emulate a practical guidance implementation, the problem is solved in a sample-and-hold manner. In each call to guidance, the feasibility problem is solved. If the control solution has magnitude ρ_1 at each node, the control at the current node is accepted and passed to the simulation. Otherwise, a perturbing control with magnitude ρ_1 is introduced and passed to the simulation. This process is repeated every second from 90 seconds down to 1 second. The resulting state and control trajectories are shown in Figures 3.10 to 3.12.

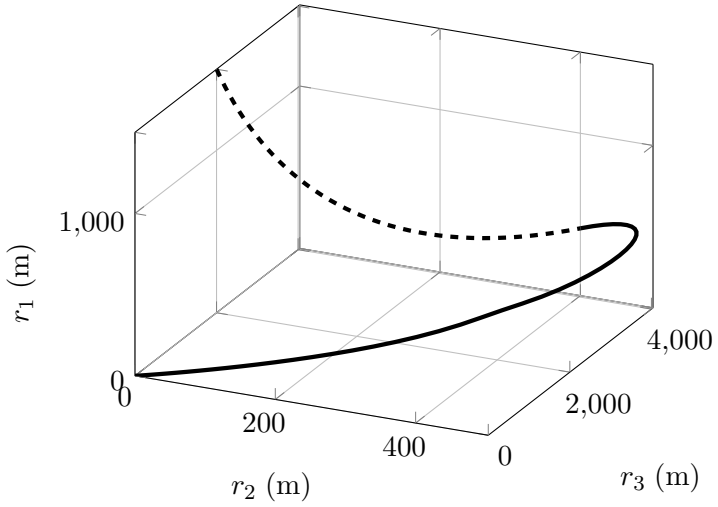


Fig. 3.10: The spacecraft's initial position is at the tip of the dashed curve. The dashed curve indicates the perturbed portion of the trajectory. After 43 seconds, a minimum time control lands the spacecraft at the desired point (indicated by the solid curve).

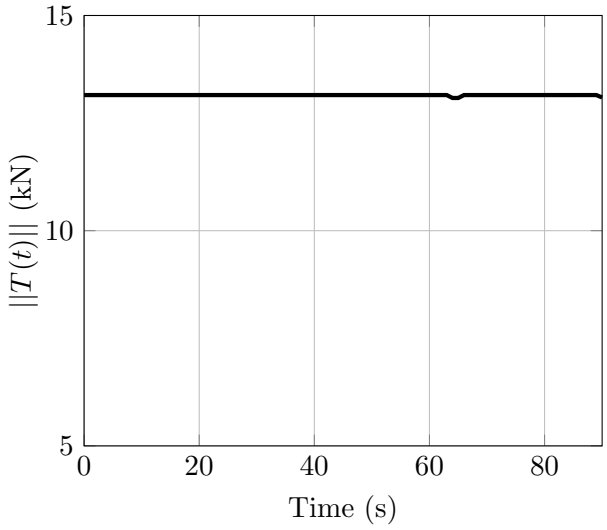


Fig. 3.11: The thrust magnitude is constant at the lower bound of $\rho_1 = 13.151$ kN.

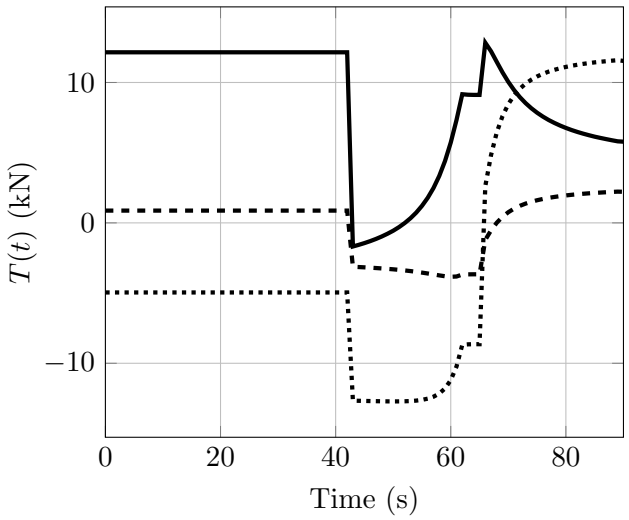


Fig. 3.12: The three components of thrust are shown as a function of time.

The problems were solved on a laptop with a 2.2 GHz processor. Each call to guidance requires the solution of one second-order cone program. On average, the solution time was 0.25 seconds.

3.8 Summary and Conclusions

This chapter presented new convexification results including a sufficient condition for the standard convexification to hold for both free and fixed final time problems, a sufficient condition for the standard convexification to hold for all final times between the minimum feasible time and the optimal time, and a perturbation technique to solve the general fixed time problem as a sequence of convex programs when the final time is greater than the optimal time. In short, the perturbation technique works as follows: perturb the initial point, solve a feasibility problem to the final point, and repeat until convexification works, which is guaranteed to happen. This results in a globally minimizing control. The perturbation technique has practical applications as demonstrated in the Mars landing example. In each call to guidance, the problem was solved in less than one second without customization, suggesting that the new perturbation technique is suitable for real-time guidance applications.

CHAPTER 4
CONTROL OF BOUNDED NONLINEAR SYSTEMS

4.1 Introduction

Many control systems include nonlinear dynamics. The popular book on nonlinear control by Khalil [2] gives several examples in Chapter 1 including circuits, robotic systems, automotive systems, and more. In fortunate cases, an exact or approximate linearization of the dynamics may be obtained so that linear control techniques are effective in achieving the control objectives [3]. In other cases, the control engineer must design a controller using techniques of nonlinear control theory. These techniques often involve finding a Lyapunov function, solving a non-convex optimal control problem, or something else of equal difficulty. The presence of practical actuator or state constraints further increases the control design challenge. This chapter describes a novel technique for control of systems with bounded nonlinearity, convex state constraints, and control constraints. The technique reduces the problem to finding bounding solutions associated with linear systems. This process is shown to be a second-order cone program (SOCP), for which efficient solvers exist [38, 39, 90, 92]. The nonlinear controller may then be interpolated from these bounding solutions in real-time. The technique benefits from a rigorous theoretical foundation and the efficient SOCP solvers. Note that although the technique uses optimization, the resulting nonlinear control solutions are not optimal in any pre-defined sense.

The chapter is organized as follows. First, systems with an additive scalar nonlinear term are considered in Section 4.2. The section includes a sufficient condition on when the nonlinear system can be controlled by bounding linear systems. The sufficient condition is then applied to control systems with convex and some non-convex control constraints. Section 4.3 generalizes the sufficient condition to systems with multi-dimensional additive nonlinearities. Two methods of controlling nonlinear systems through bounding linear sys-

tems are introduced based on this condition. The methods are demonstrated by applying them to aerospace examples. Section 4.4 provides a feedback linearization-like computational method for controlling systems with convex polytope bounded nonlinearities. A comparison between this method and a classic feedback linearization is given in this section as well through numerical examples. The chapter is concluded in Section 4.5.

4.2 Systems with Additive Scalar Nonlinearity

This section considers systems with additive scalar nonlinearities. A sufficient condition for when such systems can be controlled by bounding linear systems is given. This condition is also applied to multiple examples with varying constraints in states as well as in control input.

A brief outline of the section and its contributions is now given. Section 4.2.1 introduces a continuous-time nonlinear system. Two related linear systems are then introduced. The first contribution is a sufficient condition for solutions of these two systems to bound the nonlinear system. This condition is given in Theorem 4.1. A consequence is that the nonlinear system may be controlled by interpolating the linear controls which is demonstrated in the examples at the end of Section 4.2.1. Discrete-time nonlinear systems are investigated in Section 4.2.2. The second contribution is again a condition for the nonlinear control to be interpolated from linear controls. This is given in Theorem 4.2. The discrete-time formulation leads to a novel SOCP synthesis tool for control design in the presence of convex control and state constraints. The third contribution is the generalization of the theorems and synthesis tool for certain non-convex control constraints. The discrete-time results are demonstrated through engineering examples.

4.2.1 Continuous-Time Systems

Consider a nonlinear system of the form

$$\dot{x} = Ax + Bu + E\eta(x) \tag{4.1}$$

on spatial domain $D \subset \mathbb{R}^n$ and time domain $I = [t_0, t_f] \subset \mathbb{R}$ with a single scalar nonlinearity η satisfying the bounds

$$\forall x \in D, \quad \Delta_L \leq \eta(x) \leq \Delta_U. \tag{4.2}$$

The control objective is to drive the system to the terminal set $X_f \subset \mathbb{R}^n$ while keeping the state in the state constraint $X \subset D$ and the control in the control constraint $U \subset \mathbb{R}^n$. It is assumed that X_f, X, and U are convex. Note that the boundedness of the nonlinearity is not restrictive when X is compact since any continuous function attains a minimum and maximum on such a set. Because of the nonlinearity and the constraints, this is a challenging problem and classical nonlinear techniques such as feedback linearization, backstepping, etc. may not apply. Optimization-based techniques such as model predictive control or nonlinear optimal control require the solution of a nonlinear program, for which convergence is not guaranteed. It is now shown when the nonlinear system may be controlled by interpolation of linear controls.

Consider the auxiliary linear systems

$$\dot{x}_L = Ax_L + Bu_L + E\Delta_L, \tag{4.3}$$

$$\dot{x}_U = Ax_U + Bu_U + E\Delta_U, \tag{4.4}$$

which have been generated by replacing the nonlinearity with its bounds. The following question is now asked: Under what conditions will the resulting trajectories x_L and x_U bound the nonlinear trajectory x? In other words, under what conditions can the nonlinear control be determined by analyzing linear systems? Solutions of the linear differential equations are given by

$$x_L(t) = \Phi(t, t_0)x_0 + \int_{t_0}^{t} \Phi(t, \tau) \left[Bu_L(\tau) + E\Delta_L\right] d\tau = \Phi(t, t_0)x_0 + \int_{t_0}^{t} \alpha(t, \tau) d\tau, \tag{4.5}$$

$$x_U(t) = \Phi(t, t_0)x_0 + \int_{t_0}^{t} \Phi(t, \tau) \left[Bu_U(\tau) + E\Delta_U\right] d\tau = \Phi(t, t_0)x_0 + \int_{t_0}^{t} \gamma(t, \tau) d\tau. \tag{4.6}$$

For given functions u_L and u_U, the integrands α and γ become known vector-valued functions. Before proceeding, some preliminary results on "ordered" functions are given.

Definition 4.1. *Two scalar-valued functions $f : I \to \mathbb{R}$ and $h : I \to \mathbb{R}$ are "ordered" if $f \leq h$ or $f \geq h$ on I. Equivalently, they are ordered if there exists $\ell \in \{+1, -1\}$ such that $\ell f \leq \ell h$. Two vector-valued functions $f : I \to \mathbb{R}^n$ and $h : I \to \mathbb{R}^n$ are "ordered" if for every $j \in \{1, \ldots, n\}$ there exists $\ell_j \in \{+1, -1\}$ such that $\ell_j f^j \leq \ell_j h^j$. In other words, two vector-valued functions are ordered if their components are ordered. Note that all components do* not *have to be ordered the same.*

Lemma 4.1. *Let f and h be scalar-valued functions as in Definition 4.1. Let $\ell \in \{+1, -1\}$.*

$$\ell f \leq \ell h \quad \Longrightarrow \quad \ell \int_I f(t)dt \leq \ell \int_I h(t)dt \tag{4.7}$$

Corollary 4.1. *Let f and h be as in Lemma 4.1. Let $w : I \to [0, 1]$ be Riemann integrable. Let $g := (1 - w)f + wh$. Let $\ell \in \{+1, -1\}$. Then*

$$\ell f \leq \ell g \leq \ell h \quad \textit{and} \qquad \ell \int_I f(t)dt \leq \ell \int_I g(t)dt \leq \ell \int_I h(t)dt. \tag{4.8}$$

One of the main results is now given. It answers the above questions and provides sufficient conditions for a nonlinear control to be interpolated from linear controls.

Theorem 4.1. *If for every $t \in I$ the vector-valued functions*

$$\alpha(t, \cdot) : [t_0, t] \to \mathbb{R}^n \tag{4.9}$$

$$\gamma(t, \cdot) : [t_0, t] \to \mathbb{R}^n \tag{4.10}$$

are ordered, then $u = (1 - w)u_L + wu_U$ achieves the control objective for the nonlinear system with $w(t) = (\eta(x(t)) - \Delta_L)/(\Delta_U - \Delta_L)$.

Proof. For any Riemann integrable function $w : I \to [0, 1]$, the linear system

$$\dot{x} = Ax + B\left[(1 - w)u_L + wu_U\right] + E\left[(1 - w)\Delta_L + w\Delta_U\right] \tag{4.11}$$

has a solution given by

$$x(t) = \Phi(t, t_0)x_0 + \int_{t_0}^{t} (1 - w(\tau))\alpha(t, \tau) + w(\tau)\gamma(t, \tau)d\tau = \Phi(t, t_0)x_0 + \int_{t_0}^{t} \beta(t, \tau)d\tau. \quad (4.12)$$

If for every $t \in I$ the vector-valued functions $\alpha(t, \cdot)$ and $\gamma(t, \cdot)$ are ordered, then it follows from Definition 4.1, Lemma 4.1, and Corollary 4.1 that for every $j \in \{1, \ldots, n\}$ there exists $\ell_{tj} \in \{+1, -1\}$ such that

$$\ell_{tj}\alpha^j(t, \cdot) \leq \ell_{tj}\beta^j(t, \cdot) \leq \ell_{tj}\gamma^j(t, \cdot). \quad (4.13)$$

The double subscript tj on each ℓ is present to emphasize that the choice of ℓ depends on each t and each j. Consequently, their integrals are ordered in the same way.

$$\ell_{tj} \int_{t_0}^{t} \alpha^j(t, \tau)d\tau \leq \ell_{tj} \int_{t_0}^{t} \beta^j(t, \tau)d\tau \leq \ell_{tj} \int_{t_0}^{t} \gamma^j(t, \tau)d\tau \quad (4.14)$$

Adding $\ell_{tj}\Phi(t, t_0)x_0$ to each, it follows that

$$\ell_{tj}x_L^j(t) \leq \ell_{tj}x^j(t) \leq \ell_{tj}x_U^j(t). \quad (4.15)$$

That is, the x function is always between the x_L and x_U functions. From convexity of X and X_f, it is concluded that $\forall t \in I$,

$$x(t) \in X \quad \text{and} \quad x(T) \in X_f. \quad (4.16)$$

Upon choosing $w(t) \in [0, 1]$ such that $\eta(x(t)) = (1 - w(t))\Delta_L + w(t)\Delta_U$ and $u = (1 - w)u_L + wu_U$, the original differential equation is obtained. Furthermore, by convexity, $\forall t \in I, u(t) \in U$. □

Theorem 4.1 gives a sufficient condition for the nonlinear control problem to be solved using linear control techniques. An illustrative example is now given.

Example 1: As an example, consider the following problem

$$\dot{x}_1 = \tfrac{1}{10} \sin x_1 \cos x_2 + x_2 \tag{4.17}$$

$$\dot{x}_2 = -x_1 + u \tag{4.18}$$

with $D = X = \mathbb{R}^2$ and $U = \mathbb{R}$. The objective is to drive the first state to zero, i.e., $X_f = \{0\} \times \mathbb{R}$. The system is rewritten in the following form

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tfrac{1}{10} \sin x_1 \cos x_2 \tag{4.19}$$

so that A, B, E, and η are easily identified. The scalar nonlinearity is bounded between $\Delta_L = -1/10$ and $\Delta_U = +1/10$. Using LQR techniques to generate u_L, u_U, and their associated trajectories, it is seen in Figures 4.1 and 4.2 that the α and γ functions are ordered as required by Theorem 4.1.

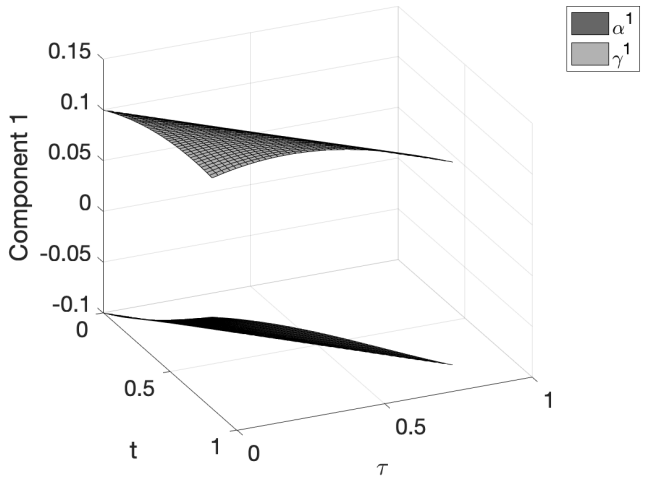


Fig. 4.1: Surfaces for α^1 and γ^1. For any fixed $t \in I$, curves along the dark surface in the τ direction $\alpha^1(t, \cdot)$ and curves along the light surface in the τ direction $\gamma^1(t, \cdot)$ are all ordered. In fact, for any (t, τ), it is evident that $\alpha^1(t, \tau) \leq \gamma^1(t, \tau)$.

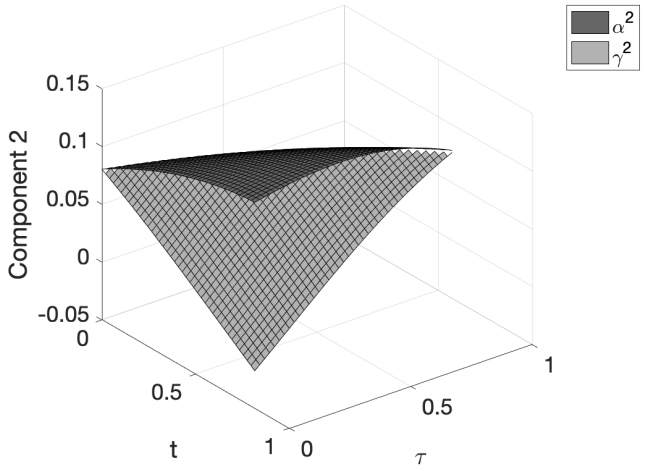


Fig. 4.2: Surfaces for α^2 and γ^2. For any fixed $t \in I$, curves along the dark surface in the τ direction $\alpha^2(t, \cdot)$ and curves along the light surface in the τ direction $\gamma^2(t, \cdot)$ are all ordered. In fact, for any (t, τ), it is evident that $\alpha^2(t, \tau) \geq \gamma^2(t, \tau)$.

Furthermore, the state trajectories x_L^1 and x_U^1 descend to the origin. Therefore, a control is obtained that drives the nonlinear system to the target set. This is shown in Figure 4.3. In this figure, and subsequent ones, the legend item L corresponds to the trajectory generated by the lower bounding system; the legend item U corresponds to the trajectory generated by the upper bounding system; the legend item N corresponds to the nonlinear trajectory.

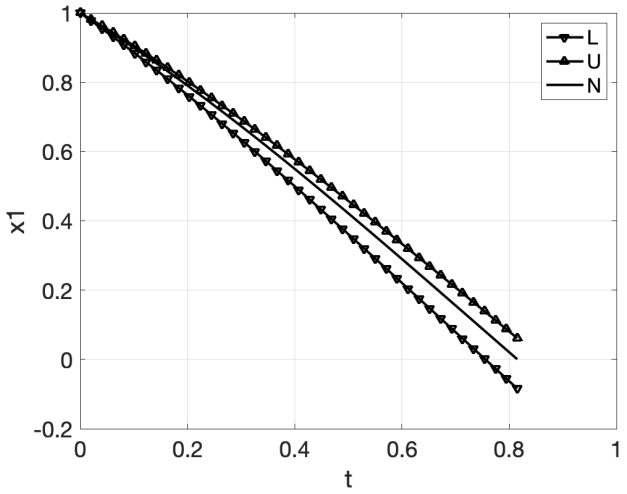


Fig. 4.3: Trajectories for x_L^1, x_U^1, and x^1, which is the nonlinear trajectory.

Theorem 4.1 requires ordering on the time domain of interest. Though this condition is sufficient, this example shows that it cannot be significantly weakened. Using the same LQR solution to generate u_L and u_U on a larger time horizon, a breakdown of the ordering is now seen with $t \approx 1.58$ in Figure 4.4. The interpolated x_1 trajectory escapes the bounding curves x_L^1 and x_U^1 at $t \approx 2.3$. This is seen in Figure 4.5.

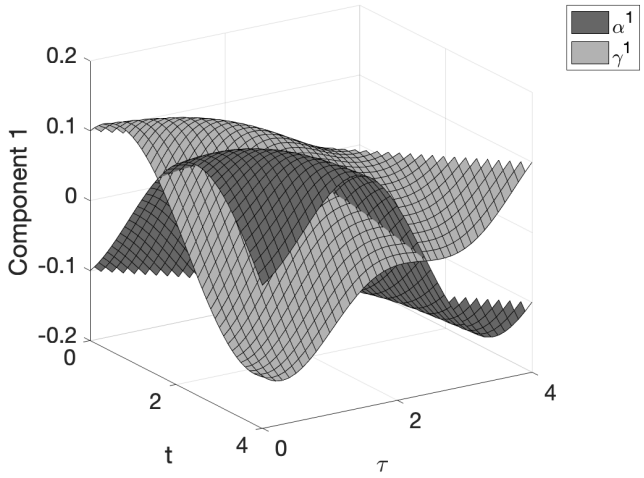


Fig. 4.4: Surfaces for α^1 and γ^1. With t fixed at approximately 1.58, the curve along the dark surface in the τ direction $\alpha^1(t, \cdot)$ and the curve along the light surface in the τ direction $\gamma^1(t, \cdot)$ cross, and hence, are not ordered.

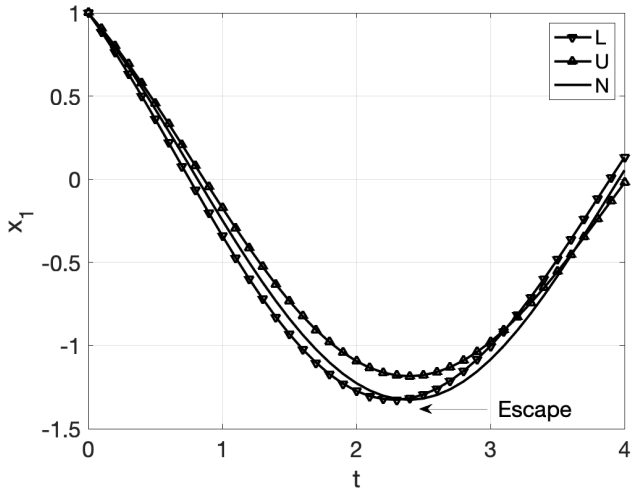


Fig. 4.5: Trajectories for x_L^1, x_U^1, and x_1, which is the nonlinear trajectory. The nonlinear trajectory escapes the lower and upper envelope curves.

In continuous-time, the theorem serves as an analysis tool: given functions u_L and u_U one is able to check if a nonlinear control can be derived from them. However, the synthesis question of how to generate suitable u_L and u_U so that the α and γ surfaces do not cross remains open. The synthesis problem seems best addressed in a discrete-time setting in which the surface crossing constraint can be embedded in a finite-dimensional optimization problem. This optimization problem can be formulated as a second-order cone program (SOCP), which can be solved efficiently. This topic is investigated next.

4.2.2 Discrete-Time Systems and Optimization

Consider a discrete-time, nonlinear system of the form

$$x_{k+1} = Ax_k + Bu_k + E\eta(x_k), \quad x_0 \text{ given} \tag{4.20}$$

with states $x_k \in X \subset \mathbb{R}^n$, controls $u_k \in U \subset \mathbb{R}^m$, and index domain $k \in I = \{0, \ldots, N-1\}$. There is a single scalar nonlinearity $\eta : X \to \mathbb{R}$ satisfying the bounds

$$\forall x \in X, \quad \Delta_L \leq \eta(x) \leq \Delta_U. \tag{4.21}$$

Discrete-time systems may arise naturally in digital settings or by discretizing continuous-time systems. Discretization method shown in Section 2.3.1 is used in subsequent examples. The problem is to determine a $\mathbb{R}^m$-valued control sequence

$$\{u_k\}_{k=0}^{N-1} = \{u_0, u_1, \ldots, u_{N-1}\} \tag{4.22}$$

to achieve the following objectives: drive the system to the terminal set X_f while keeping the state in the state constraint X and the control in the control constraint U. It is assumed that X_f and X are convex sets. It is assumed that the control set U is the union of convex sets, i.e.,

$$U = \bigcup_i U_i, \quad U_i \text{ is convex.} \tag{4.23}$$

Note that U may be non-convex and even disconnected.

Consider the auxiliary linear systems

$$x_{L,k+1} = Ax_{L,k} + Bu_{L,k} + E\Delta_L, \quad x_{L,0} = x_0, \tag{4.24}$$

$$x_{U,k+1} = Ax_{U,k} + Bu_{U,k} + E\Delta_U, \quad x_{U,0} = x_0, \tag{4.25}$$

which have been generated by replacing the nonlinearity with its bounds. Solutions of these equations are given by

$$x_{L,k} = A^k x_0 + \sum_{i=0}^{k-1} A^{k-1-i}(Bu_{L,i} + E\Delta_L), \tag{4.26}$$

$$x_{U,k} = A^k x_0 + \sum_{i=0}^{k-1} A^{k-1-i}(Bu_{U,i} + E\Delta_U). \tag{4.27}$$

Upon defining the $\mathbb{R}^n$-valued quantities

$$\alpha_{k,i} := A^{k-1-i}(Bu_{L,i} + E\Delta_L), \tag{4.28}$$

$$\gamma_{k,i} := A^{k-1-i}(Bu_{U,i} + E\Delta_U), \tag{4.29}$$

solutions of the auxiliary systems may be written more compactly as

$$x_{L,k} = A^k x_0 + \sum_{i=0}^{k-1} \alpha_{k,i}, \tag{4.30}$$

$$x_{U,k} = A^k x_0 + \sum_{i=0}^{k-1} \gamma_{k,i}. \tag{4.31}$$

For given control sequences $\{u_{L,k}\}_{k=0}^{N-1}$ and $\{u_{U,k}\}_{k=0}^{N-1}$, the sequences $\{\alpha_{k,i}\}_{i=0}^{k-1}$ and $\{\gamma_{k,i}\}_{i=0}^{k-1}$ become known for every $k \in \{1, \ldots, N\}$.

Definition 4.2. *The control sequences $\{u_{L,k}\}_{k=0}^{N-1}$ and $\{u_{U,k}\}_{k=0}^{N-1}$ are "compatible" if for every $k \in I$ there exists i such that $u_{L,k} \in U_i$ and $u_{U,k} \in U_i$.*

Definition 4.3. *Two scalar-valued sequences $\{f_i\}_{i=0}^{N-1}$ and $\{h_i\}_{i=0}^{N-1}$ are "ordered" if for every $i \in I$, $f_i \leq h_i$ or $f_i \geq h_i$. Equivalently, they are ordered if there exists $\ell \in \{+1, -1\}$ such that for every $i \in I$, $\ell f_i \leq \ell h_i$. To simplify notation, it is written as $\{\ell f_i\}_{i=0}^{N-1} \leq \{\ell h_i\}_{i=0}^{N-1}$. Two vector-valued sequences $\{f_i\}_{i=0}^{N-1}$ and $\{h_i\}_{i=0}^{N-1}$ are "ordered" if for every $j \in \{1, \ldots, n\}$ there exists $\ell_j \in \{+1, -1\}$ such that $\{\ell_j f_i^j\}_{i=0}^{N-1} \leq \{\ell_j h_i^j\}_{i=0}^{N-1}$. In other words, two vector-valued sequences are ordered if their components are ordered sequences. Note that all components do not have to be ordered the same.*

Lemma 4.2. *Let $\{f_i\}_{i=0}^{N-1}$ and $\{h_i\}_{i=0}^{N-1}$ be scalar-valued sequences. Let $\ell \in \{+1, -1\}$.*

$$\{\ell f_i\}_{i=0}^{N-1} \leq \{\ell h_i\}_{i=0}^{N-1} \implies \ell \sum_{i=0}^{N-1} f_i \leq \ell \sum_{i=0}^{N-1} h_i \tag{4.32}$$

Corollary 4.2. *Let $\{f_i\}_{i=0}^{N-1}$ and $\{h_i\}_{i=0}^{N-1}$ be as in Lemma 4.2. Let $\{w_i\}_{i=0}^{N-1}$ be a scalar-valued sequence taking values in $[0, 1]$. Let $g_i = (1 - w_i) f_i + w_i h_i$. Let $\ell \in \{+1, -1\}$. Then*

$$\begin{aligned} &\{\ell f_i\}_{i=0}^{N-1} \leq \{\ell g_i\}_{i=0}^{N-1} \leq \{\ell h_i\}_{i=0}^{N-1} \\ &\implies \ell \sum_{i=0}^{N-1} f_i \leq \ell \sum_{i=0}^{N-1} g_i \leq \ell \sum_{i=0}^{N-1} h_i \end{aligned} \tag{4.33}$$

The following theorem provides sufficient conditions for a nonlinear control to be interpolated from control sequences associated with the auxiliary linear systems.

Theorem 4.2. *Let* $\{u_{L,k}\}_{k=0}^{N-1}$ *and* $\{u_{U,k}\}_{k=0}^{N-1}$ *be compatible control sequences achieving the control objectives for the auxiliary systems in Equations* (4.24) *and* (4.25)*, respectively. Let* $\alpha_{k,i}$ *and* $\gamma_{k,i}$ *be given by Equations* (4.28) *and* (4.29)*, respectively. If for every* $k \in \{1, \ldots, N\}$ *the* $\mathbb{R}^n$*-valued sequences* $\{\alpha_{k,i}\}_{i=0}^{k-1}$ *and* $\{\gamma_{k,i}\}_{i=0}^{k-1}$ *are ordered, then the control sequence* $\{u_k\}_{k=0}^{N-1}$ *achieves the control objective for the nonlinear system in Equation* (4.20) *with* $u_k = (1 - w_k)u_{L,k} + w_k u_{U,k}$ *and* $w_k = (\eta(x_k) - \Delta_L)/(\Delta_U - \Delta_L)$*.*

Proof. For any sequence $\{w_k\}_{k=0}^{N-1}$ with elements in the set $[0, 1]$, the linear system

$$\begin{aligned} x_{k+1} = Ax_k + B[(1 - w_k)u_{L,k} + w_k u_{U,k}] \\ + E[(1 - w_k)\Delta_L + w_k \Delta_U] \end{aligned} \tag{4.34}$$

has a solution given by

$$x_k = A^k x_0 + \sum_{i=0}^{k-1} (1 - w_i)\alpha_{k,i} + w_i \gamma_{k,i}. \tag{4.35}$$

Upon defining the $\mathbb{R}^n$-valued quantity

$$\beta_{k,i} := (1 - w_i)\alpha_{k,i} + w_i \gamma_{k,i}, \tag{4.36}$$

the solution may be written more compactly as

$$x_k = A^k x_0 + \sum_{i=0}^{k-1} \beta_{k,i}. \tag{4.37}$$

If for every $k \in \{1, \ldots, N\}$ the $\mathbb{R}^n$-valued sequences $\{\alpha_{k,i}\}_{i=0}^{k-1}$ and $\{\gamma_{k,i}\}_{i=0}^{k-1}$ are ordered, then it follows from Definition 4.3, Lemma 4.2, and Corollary 4.2 that for every $k \in \{1, \ldots, N\}$ and $j \in \{1, \ldots, n\}$ there exists $\ell_{kj} \in \{+1, -1\}$ such that

$$\{\ell_{kj}\alpha_{k,i}^j\}_{i=0}^{k-1} \leq \{\ell_{kj}\beta_{k,i}^j\}_{i=0}^{k-1} \leq \{\ell_{kj}\gamma_{k,i}^j\}_{i=0}^{k-1}. \tag{4.38}$$

Consequently, summations of the sequences are ordered in the same way. For every $k \in \{1, \ldots, N\}$,

$$\ell_{kj} \sum_{i=0}^{k-1} \alpha_{k,i}^{j} \leq \ell_{kj} \sum_{i=0}^{k-1} \beta_{k,i}^{j} \leq \ell_{kj} \sum_{i=0}^{k-1} \gamma_{k,i}^{j}. \tag{4.39}$$

Adding $\ell_{kj} A^k x_0$ to each, it follows that for every $k \in \{1, \ldots, N\}$

$$\ell_{kj} x_{L,k}^{j} \leq \ell_{kj} x_{k}^{j} \leq \ell_{kj} x_{U,k}^{j}. \tag{4.40}$$

That is, the x sequence is always between the x_L and x_U sequences. From convexity of X and X_f, it is concluded that for every $k \in I$, $x_k \in X$ and $x_N \in X_f$. Upon choosing w_k such that $\eta(x_k) = (1 - w_k)\Delta_L + w_k \Delta_U$ and $u_k = (1 - w_k)u_{L,k} + w_k u_{U,k}$, the original nonlinear system in Equation (4.20) is obtained. Lastly, it follows from the compatibility condition stated in Definition 4.2 that for every $k \in I$ there exists i such that $u_{L,k} \in U_i$ and $u_{U,k} \in U_i$. Because u_k is obtained by a convex combination of $u_{L,k}$ and $u_{U,k}$ and because U_i is convex, it follows that $u_k \in U_i$. This completes the proof. □ □

The ordering and compatibility conditions in Theorem 4.2 may be incorporated as constraints in an optimization problem, which can be solved to generate suitable control sequences $\{u_{L,k}\}_{k=0}^{N-1}$ and $\{u_{U,k}\}_{k=0}^{N-1}$. Once these are known, the nonlinear control may be interpolated. The ordering condition can be simplified so that the same ordering holds for all times k, and this is observable in the upcoming examples. This eliminates the need for the integer variables ℓ_{kj} and reduces the computational complexity of the problem. Given an ordering, the optimization problem is used to prove feasibility of the ordering and generate the bounding controls; it is not used to find an ordering. Provided the sets X_f, X, and U are convex second-order cones, the resulting optimization problem is a SOCP. Provided U is not convex but each U_i is a convex second-order cone, the resulting optimization problem is a mixed-integer SOCP (MI-SOCP). This is illustrated in the following three sections in which engineering problems are solved.

Van der Pol Oscillator

In this first example, the Van der Pol oscillator is considered. The oscillator originally served as a model for an electrical circuit with a triode valve. It has later appeared in other applications such as biomedical engineering, power systems, combustion processes, and robotics [93]. It also serves as a classic example in nonlinear systems and control [2]. The Van der Pol oscillator has a similar structural form to the Duffing oscillator, which has been used to model various engineering systems including microelectromechanic systems (MEMS) [94]. The continuous-time Van der Pol equations are

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -x_1 + \epsilon x_2 - \epsilon x_2 x_1^2 + u. \tag{4.41}$$

Following the discretization approach outlined in Section 2.3.1 with a time step of 0.05 seconds and $\epsilon = 0.1$, the discrete-time Van der Pol oscillator is (to four decimals)

$$\begin{aligned} x_{k+1} = &\begin{bmatrix} 0.9987 & 0.0501 \\ -0.0501 & 1.0038 \end{bmatrix} x_k + \begin{bmatrix} 0.0013 \\ 0.0501 \end{bmatrix} u_k \\ &+ \begin{bmatrix} -0.0001 \\ -0.0050 \end{bmatrix} x_{2,k} x_{1,k}^2. \end{aligned} \tag{4.42}$$

The quantities A, B, E, and η are readily identified, and this system is consistent with that in Equation (4.20). The goal is to drive the system from its initial condition $x_0 = [\frac{1}{2}\text{ m}, -1\text{ m/s}]^\top$ to the origin in one second ($N = 21$). The control magnitude is bounded by 1.4 m/s^2. The remaining problem data are specified to be

$$\begin{aligned} &X_f = \{0\}, \quad X = [0, \tfrac{1}{2}] \times [-1, 0] \\ &U = [-1.4, 1.4], \quad \Delta_L = -\tfrac{1}{4}, \quad \Delta_U = 0, \\ &\forall k \in \{1, \ldots, N\}, \quad \ell_{k1} = 1, \quad \ell_{k2} = -1. \end{aligned} \tag{4.43}$$

By choosing X in this way, the states will not overshoot their final state. To find control solutions to the auxiliary linear systems that satisfy the conditions in Theorem 4.2, the

following finite-dimensional optimization problem is posed and solved using YALMIP [92] and Gurobi's SOCP solver [90].

$$\min \quad \sum_{k=0}^{N-1} ||u_{L,k}||^2 + ||u_{U,k}||^2 \tag{4.44a}$$

subj. to

$$x_{L,k+1} = Ax_{L,k} + Bu_{L,k} + E\Delta_L, k \in I \tag{4.44b}$$

$$x_{U,k+1} = Ax_{U,k} + Bu_{U,k} + E\Delta_U, k \in I \tag{4.44c}$$

$$x_{L,0} = x_{U,0} = x_0,\ x_{L,N} = x_{U,N} = 0 \tag{4.44d}$$

$$\tfrac{1}{2} \geq x^1_{L,k}, x^1_{U,k} \geq 0 \geq x^2_{L,k}, x^2_{U,k} \geq -1, k \in I \tag{4.44e}$$

$$1.4 \geq u_{L,k}, u_{U,k} \geq -1.4, \quad k \in I \tag{4.44f}$$

$$\alpha_{k,i} = A^{k-1-i}(Bu_{L,i} + E\Delta_L), (k,i) \in \mathcal{S} \tag{4.44g}$$

$$\gamma_{k,i} = A^{k-1-i}(Bu_{U,i} + E\Delta_U), (k,i) \in \mathcal{S} \tag{4.44h}$$

$$\alpha^1_{k,i} \leq \gamma^1_{k,i}, (k,i) \in \mathcal{S} \tag{4.44i}$$

$$\alpha^2_{k,i} \geq \gamma^2_{k,i}, (k,i) \in \mathcal{S} \tag{4.44j}$$

The objective function in (4.44a) helps regularize, or smooth, the controls but it is not required. The auxiliary linear systems appear in (4.44b) and (4.44c). The initial and final conditions are in (4.44d). The state constraints are in (4.44e) and the control constraints are in (4.44f). With $\mathcal{S}$ being the set of $k \in \{1, \ldots, N\}$ and $i \in \{0, \ldots, k-1\}$, the α and γ sequences are defined in (4.44g) and (4.44h). Lastly, the ordering constraints corresponding to $(\ell_{k1}, \ell_{k2}) = (+1, -1)$ for each k are in (4.44i) and (4.44j). Gurobi solves this problem in 0.01 seconds on a 2019 iMac with 3.7 GHz 6-core Intel Core i5 processor. This proves the given ordering is feasible and the nonlinear control is interpolatable from $u_{L,k}$ and $u_{U,k}$. As a minor remark, the optimization problem remains feasible for (ℓ_{k1}, ℓ_{k2}) equal to $(+1, +1)$, $(-1, -1)$, and $(-1, +1)$ for each k. The resulting state trajectories are shown in Figure 4.6. The control trajectory is shown in Figure 4.7. The figures show that the state reaches the origin and the control satisfies the bounds.

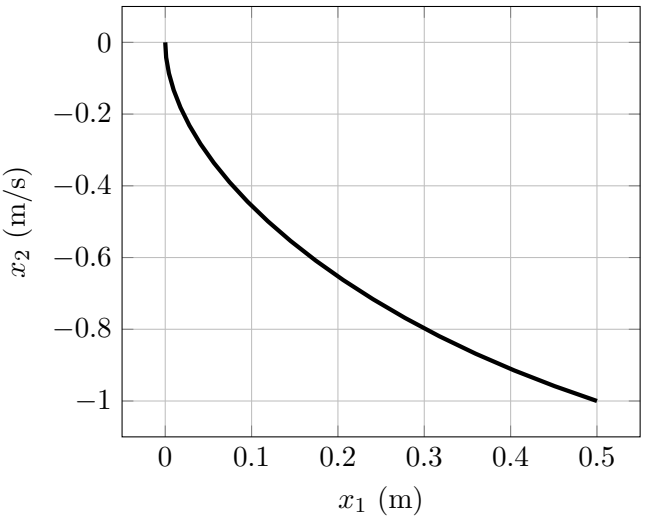


Fig. 4.6: The state trajectory begins at the bottom right and terminates at the origin in the upper left. The transfer time is one second.

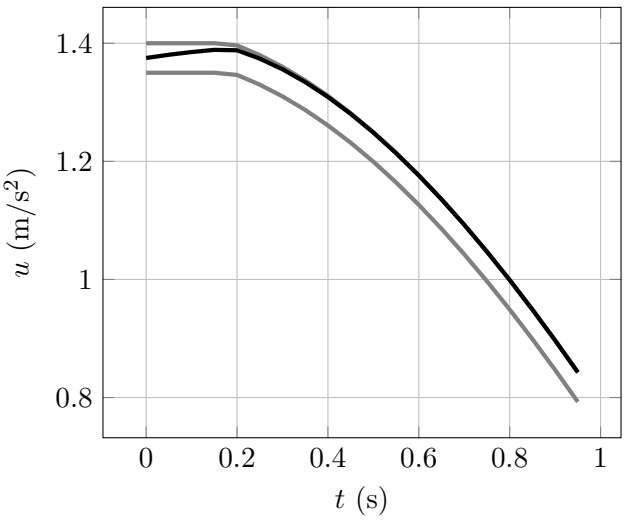


Fig. 4.7: The controls generated by the SOCP are the gray curves. They bound the black curve, which is the nonlinear control. The nonlinear control is obtained by interpolating between the gray curves.

To illustrate the technique on a problem with non-convex control constraints, a quantized control is sought that achieves the same objectives. The control is only permitted to take integer values between -3 and 3. A quantization threshold Q is set at 0.03. The threshold allows for small deviations in the control. For instance, because $u_k = 2.02$ is between $2 - Q = 1.97$ and $2 + Q = 2.03$, it is an allowable control value. As before, to find control solutions to the auxiliary linear systems that satisfy the conditions in Theorem 4.2, a finite-dimensional optimization problem is posed. This time, however, because of the integer nature of the control, the resulting problem is a mixed-integer SOCP (MI-SOCP). The problem is posed and solved using YALMIP [92] and Gurobi's MI-SOCP solver [90]. The optimization problem is similar to that in Equation (4.44). The only change is to replace the control constraint in Equation (4.44f) with the following two constraints.

$$b_k \in \{-3, -2, -1, 0, 1, 2, 3\}, \quad k \in I \tag{4.45a}$$

$$b_k - Q \leq u_{L,k}, u_{U,k} \leq b_k + Q, \quad k \in I \tag{4.45b}$$

Gurobi solves this problem in 0.17 seconds on a 2019 iMac with 3.7 GHz 6-core Intel Core i5 processor. The resulting state trajectories are shown in Figure 4.8. The control trajectory is shown in Figure 4.9. The figures show that the state reaches the origin and the control satisfies the quantization constraint.

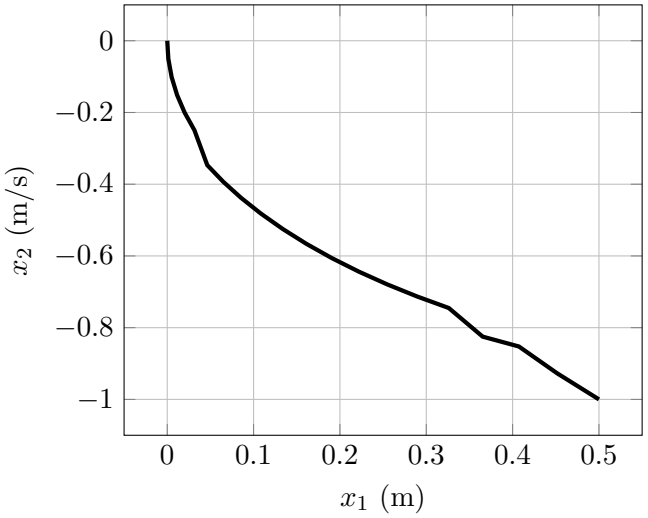


Fig. 4.8: The state trajectory begins at the bottom right and terminates at the origin in the upper left. The transfer time is one second.

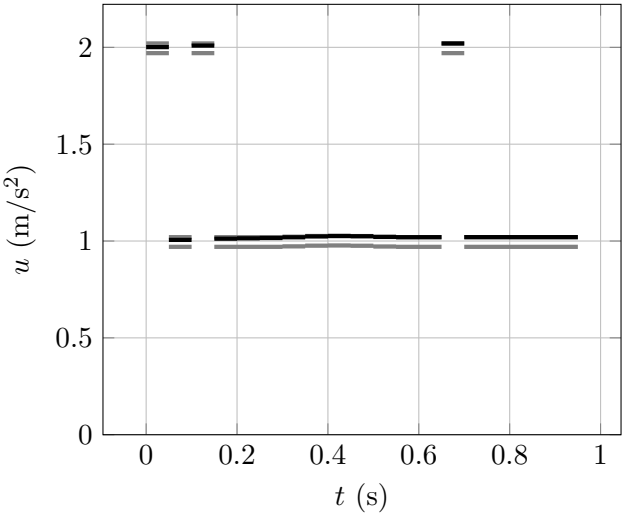


Fig. 4.9: The controls generated by the MI-SOCP are the gray curves. They bound the black curve, which is the nonlinear control. The nonlinear control is obtained by interpolating between the gray curves. It takes values of (approximately) one and two.

Pendulum Control with DC Motor

In the second example, a pendulum that is being driven by a DC motor [1, 95] is considered. The pendulum consists of a massless rigid rod which is connected directly to the motor shaft in one end and having a mass in the other end. The continuous-time equations for this system are

$$\begin{bmatrix} \dot{\theta} \\ \dot{\omega} \\ \dot{I} \end{bmatrix} = \begin{bmatrix} \omega \\ -\frac{mg\ell}{m\ell^2+J}\sin\theta - \frac{\beta}{m\ell^2+J}\omega + \frac{K_t}{m\ell^2+J}I \\ -\frac{K_b}{L}\omega - \frac{R}{L}I + \frac{1}{L}V \end{bmatrix}, \tag{4.46}$$

where θ is the angular position of the pendulum in radians measured from the downward position, ω is the angular velocity of the pendulum in radians per second, and I is the armature current of the DC motor in amperes. The motor armature voltage in volts is V; the voltage is the control input. The constant parameters are explained in Table 4.1. The table also shows the numerical values used in this example.

Table 4.1: Parameter explanations and values [1].

Symbol	Explanation	Value
m	Pendulum mass	0.1 kg
g	Gravity	9.81 m/s^2
ℓ	Pendulum length	0.24 m
β	Damping constant	0.01 Nms
J	Inertia of the motor	2.02×10^{-5} kgm^2
K_t	Motor torque	0.052 Nm/A
K_b	Motor back EMF	0.052 V/rad/s
L	Armature inductance	1.8×10^{-3} H
R	Armature resistance	1.3 Ω

Upon defining the state vector as $x = [\theta, \omega, I]^\top$ and the control input as $u = V$, the above continuous-time equations can be written more concisely as

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{\beta}{m\ell^2+J} & \frac{K_t}{m\ell^2+J} \\ 0 & -\frac{K_b}{L} & -\frac{R}{L} \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} u + \begin{bmatrix} 0 \\ -\frac{mg\ell}{m\ell^2+J} \\ 0 \end{bmatrix} \sin x_1. \tag{4.47}$$

Following the discretization approach in Section 2.3.1 with a time step of 0.05 seconds, the discrete-time equations are

$$\begin{aligned} x_{k+1} = &\begin{bmatrix} 1 & 0.0475 & 0.0006 \\ 0 & 0.9012 & 0.0113 \\ 0 & -0.0362 & -0.0005 \end{bmatrix} x_k + \begin{bmatrix} 0.0079 \\ 0.3200 \\ 0.7568 \end{bmatrix} u_k \\ &+ \begin{bmatrix} -0.0492 \\ -1.9347 \\ 0.0753 \end{bmatrix} \sin x_{1,k}. \end{aligned} \tag{4.48}$$

The quantities A, B, E, and η are readily identified, and the system is consistent with that in Equation (4.20). The goal is to drive the system from its initial, downward state $\theta_0 = 0$ rad and $\omega_0 = 0$ rad/s to the fixed, horizontal state of $\theta_N = \frac{\pi}{2}$ rad and $\omega_N = 0$ rad/s. The initial current is $I_0 = 0$ amperes. The final current I_N is free. The control magnitude is bounded by 8 volts. The remaining problem data are specified to be

$$\begin{aligned} &X_f = \{\tfrac{\pi}{2}\} \times \{0\} \times \mathbb{R}, \quad X = [0, \tfrac{\pi}{2}] \times \mathbb{R} \times \mathbb{R}, \\ &U = [-8, 8], \quad \Delta_L = 0, \quad \Delta_U = 1, \\ &\forall k \in \{1, \ldots, N\}, \quad \ell_{k1} = -1, \quad \ell_{k2} = -1, \quad \ell_{k3} = 1. \end{aligned} \tag{4.49}$$

To find control solutions to the auxiliary linear systems that satisfy the condition in Theorem 4.2, N is set to 15 and a finite-dimensional optimization problem is posed and solved using YALMIP [92] and Gurobi's SOCP solver [90]. The problem structure is similar to the one presented in Equation (4.44), and for this reason, it is not given explicitly. Gurobi

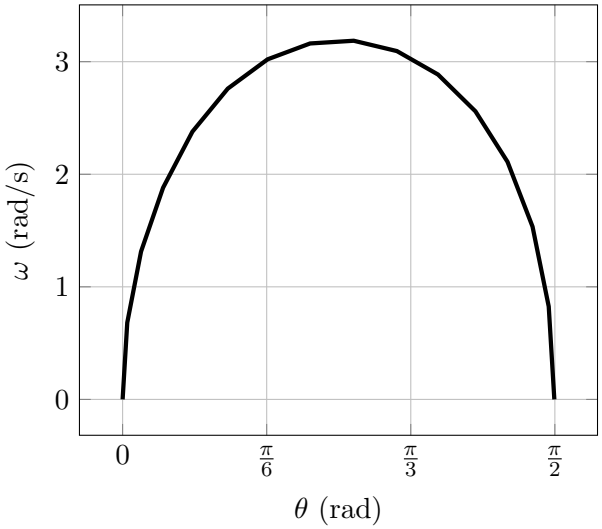


Fig. 4.10: The state trajectory begins at the origin in the bottom left and terminates at the bottom right of the figure corresponding to the horizontal position of the pendulum.

solves this problem in less than 0.001 seconds on a 2019 iMac with 3.7 GHz 6-core Intel Core i5 processor. This proves the given ordering is feasible and the nonlinear control is interpolatable from $u_{L,k}$ and $u_{U,k}$. The resulting state trajectories are shown in Figure 4.10. The control trajectory is shown in Figure 4.11. The figures show that the state reaches the terminal set and the control satisfies the bounds.

Automatic Lane Shift Maneuver

In the final example, let us consider a lane shift maneuver of a vehicle moving at constant speed. The continuous-time equations for a continuous-steering car are [96]

$$\begin{bmatrix} \dot{a} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} V\cos\theta \\ V\sin\theta \\ \frac{V}{L}\tan\phi \\ u_\phi \end{bmatrix} \tag{4.50}$$

where a is the along-track (horizontal) position of the vehicle, y is the cross-track (vertical) position of the vehicle, θ is the orientation of the vehicle from the positive a-axis, and ϕ is the turning angle of the vehicle. V is the constant speed of the vehicle, L is the wheelbase

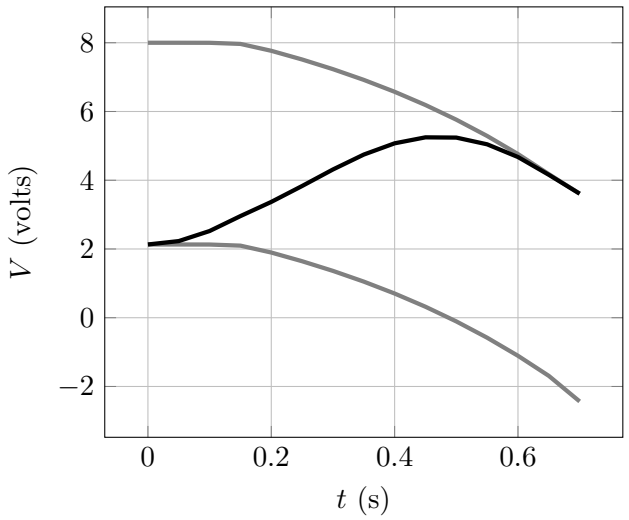


Fig. 4.11: The controls generated by the SOCP are the gray curves. They bound the black curve, which is the nonlinear control. The nonlinear control is obtained by interpolating between the gray curves. Observe that the upper gray curve saturates at the control limit of 8 volts.

of the vehicle, and u_ϕ is the control input.

A lane shift maneuver is one in which the cross-track position y measured along the vertical axis is shifted. With this maneuver in mind, the along-track position a is free and decoupled from the problem; hence, the $\dot{a}$ equation is dropped in the subsequent analysis. Because the speed of the vehicle V is constant, a new variable $\omega = V/L \tan\phi$ is introduced. With the state vector $x = [y, \theta, \omega]^\top$, the continuous-time system is

$$\begin{bmatrix} \dot{y} \\ \dot{\theta} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} 2\sqrt{2}V\theta/\pi + V\left(\sin\theta - 2\sqrt{2}\theta/\pi\right) \\ \omega \\ u \end{bmatrix}. \tag{4.51}$$

Observe that in the $\dot{y}$ equation, a linear term $2\sqrt{2}V\theta/\pi$ has been added and subtracted. For vehicle orientations θ in the set $[-\frac{\pi}{4}, \frac{\pi}{4}]$, this represents a linear approximation to the $\sin\theta$ term and reduces the effect of the nonlinearity. The new control variable is $u = V/L \sec^2(\phi) u_\phi$.

Following the discretization approach in Section 2.3.1 with a time step of 0.02 seconds, speed of $V = 20$ m/s, and wheelbase of $L = 3$ m, the discrete-time system is

$$\begin{aligned} x_{k+1} = &\begin{bmatrix} 1.0000 & 0.3601 & 0.0036 \\ 0 & 1.0000 & 0.0200 \\ 0 & 0 & 1.0000 \end{bmatrix} x_k + \begin{bmatrix} 0.0000 \\ 0.0002 \\ 0.0200 \end{bmatrix} u_k \\ &+ \begin{bmatrix} 0.4000 \\ 0 \\ 0 \end{bmatrix} \left(\sin(x_{2,k}) - \frac{2\sqrt{2}}{\pi} x_{2,k} \right) \end{aligned} \tag{4.52}$$

Again, the quantities A, B, E, and η are easily identified, and the system is consistent with that in Equation (4.20). Because the linear term has been added and subtracted, the nonlinear term is now $\sin(x_{2,k}) - \frac{2\sqrt{2}}{\pi} x_{2,k}$, which is bounded in magnitude by 0.1486. Upon restricting the turning angle ϕ to the set $[-\frac{\pi}{4}, \frac{\pi}{4}]$, magnitude of the third state becomes bounded by $\frac{V}{L} = \frac{20}{3}$. The control magnitude is bounded by 10 rad/s^2. The goal is to perform a plus 5 m lane shift in one second. With these requirements in place, the problem data are specified as

$$\begin{aligned} &X_f = \{5\} \times \{0\} \times \{0\}, X = \mathbb{R} \times [-\tfrac{\pi}{4}, \tfrac{\pi}{4}] \times [-\tfrac{20}{3}, \tfrac{20}{3}], \\ &U = [-10, 10], \Delta_L = -0.1486, \Delta_U = 0.1486, \\ &\forall k \in \{1, \ldots, N\}, \quad \ell_{k1} = 1, \quad \ell_{k2} = -1, \quad \ell_{k3} = 1. \end{aligned} \tag{4.53}$$

To find control solutions to the auxiliary linear systems that satisfy the condition in Theorem 4.2, a finite-dimensional optimization problem is posed and solved using YALMIP [92] and Gurobi's SOCP solver [90]. The problem structure is similar to the one presented in Equation (4.44), and for this reason, it is not given explicitly. Gurobi solves this problem in less than 0.001 seconds on a 2019 iMac with 3.7 GHz 6-core Intel Core i5 processor. This proves the given ordering is feasible and the nonlinear control is interpolatable from $u_{L,k}$ and $u_{U,k}$. The cross-track position of the vehicle is shown in Figure 4.12. The control

trajectory is shown in Figure 4.13. The figures show that the lane change is achieved and the control satisfies the bounds.

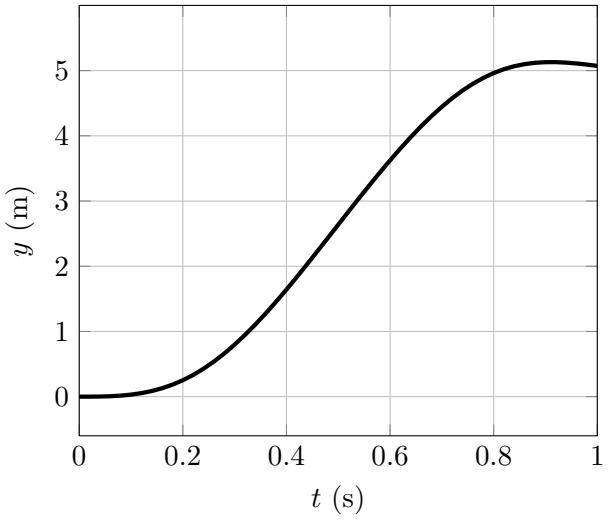


Fig. 4.12: The y-position of the vehicle as a function of time.

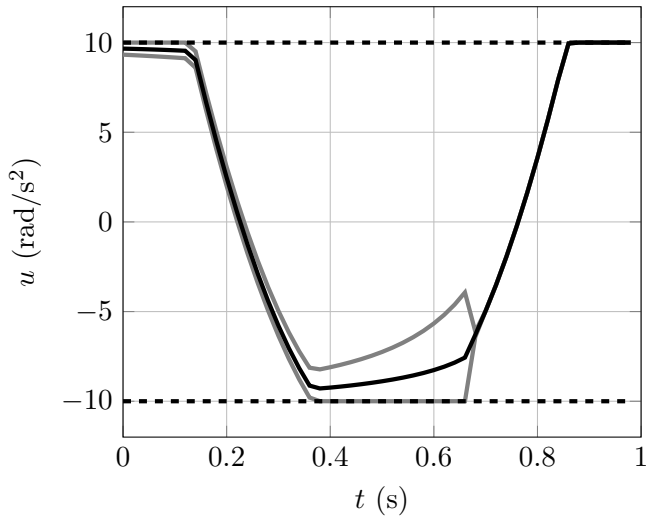


Fig. 4.13: The controls generated by the SOCP are the gray curves. They bound the black curve, which is the nonlinear control. The nonlinear control is obtained by interpolating between the gray curves. All control functions remain between -10 and +10 rad/s^2.

4.3 Systems with Additive Multi-dimensional Nonlinearities

This section generalizes the sufficient condition from previous section to discrete systems with additive multi-dimensional nonlinearities in the presence of convex state and control constraints. This theoretical result leads to two numerical approaches for solving the nonlinear constrained problem: one requires solving a single convex optimization problem and the other requires solving a sequence of convex optimization problems.

The techniques are applied to two spacecraft trajectory design and control problems: 1) a spacecraft is constrained to stay a certain distance from another space object in orbit while moving from one location to another in a finite time and 2) a spacecraft is to change its attitude in a finite time. The first problem may arise in the inspection phase of an on-orbit servicing (OOS) mission [97]. For such a mission, the controlled spacecraft must stay far enough from the target space object for safety reasons but also close enough to efficiently perform the inspection. In general, OOS missions are needed to either repair a damaged spacecraft or to extend the active lifetime of a spacecraft. An example of a spacecraft that has had multiple OOS missions to service it is the Hubble Space Telescope [98]. Beyond OOS missions, there are other types of missions where this type of problem could arise such as asteroid proximity operations [99, 100].

The finite time attitude control problem arises in optical applications where a picture of a space object is to be taken at a certain time [101]. There may, for example, be a short time window when lighting of a space object is sufficient due to the relative positions of the space object, the controlled spacecraft, and the sun [102, 103]. The problem could also arise when a spacecraft wants to do an impulsive or so-called "delta-v" maneuver [104] and its thrusters need to be oriented in the correct direction.

The remainder of the section is structured as follows. Section 4.3.1 describes the problem of interest, which is a nonlinear control problem with convex state and control constraints, and proves sufficient conditions (see Theorem 4.3) for the problem to be solvable in a single convex program. A sequential or resetting approach is then outlined for cases in which the sufficient conditions are not satisfied. The theoretical results and associated

algorithms are applied to a constrained relative orbital motion problem in Section 4.3.2 and an attitude control problem in Section 4.3.3.

4.3.1 Problem and Main Result

This section describes the problem of interest, provides a sufficient condition for its solution as a single convex program, and describes a practical algorithm for implementing the theoretical results. Consider a nonlinear system of the form

$$\dot{x} = Ax + Bu + E\eta(x), \quad x_0 = x(t_0) \text{ given} \tag{4.54}$$

where $A \in \mathbb{R}^{n\times n}$ is the system matrix, $B \in \mathbb{R}^{n\times m}$ is the control-influence matrix, and $E \in \mathbb{R}^{n\times p}$ is a mapping for the nonlinearity $\eta : D \to \mathbb{R}^p$. The system is defined on a spatial domain $D \subset \mathbb{R}^n$ and time domain $I = [t_0, t_f]$ where t_0 is the initial time and t_f is the final time. The given initial condition is $x_0 = x(t_0)$. It is assumed that the range of the nonlinearity η is bounded by a convex polytope as

$$\forall x \in D, \quad \eta(x) \in \text{co}\{\delta_1, \delta_2, \ldots, \delta_q\} \subset \mathbb{R}^p \tag{4.55}$$

where $\delta_i \in \mathbb{R}^p$ are the vertices of the polytope and 'co' denotes the convex hull. The trajectory design and control problem is to drive the state to the terminal constraint set $X_f \subset D$ while maintaining $x \in X \subset D$ and $u \in U \subset \mathbb{R}^m$. All of the constraint sets X_f, X, and U are assumed to be convex. The polytopic range assumption is satisfied when X is compact and η is continuous since each component of η is guaranteed to attain a minimum and maximum. These minima and maxima may then serve as vertices of the polytope. Note that this is a feasibility problem; no objective function is present.

Though motivated by continuous-time dynamics in aerospace applications, all analysis is conducted on an analogous discrete-time system

$$x[k+1] = A_d x[k] + B_d u[k] + E_d \eta(x[k]), \quad x[0] = x(t_0) \text{ given} \tag{4.56}$$

where the discrete-time index is $k \in I_d = \{0, \ldots, N-1\}$. A significant portion of the analysis in this section can be done in either continuous or discrete-time. However, for use as a control synthesis tool using finite-dimensional optimization, it is best to use a discrete-time formulation. To the author's knowledge, all results on lossless convexification of optimal control problems are based on continuous-time formulations and measure-theoretic considerations. As such, the 'lossless' guarantees do not hold up in the associated discrete-time algorithms. The use of discrete-time dynamics in the formulation is superior in this respect. A particular discretization strategy used in the forthcoming examples is provided in Section 2.3.1.

Now consider the auxiliary systems

$$x_i[k+1] = A_d x_i[k] + B_d u_i[k] + E_d \delta_i, \quad x_i[0] = x_0, \quad i \in \{1, \ldots, q\} \tag{4.57}$$

where in the i^{th} system the nonlinearity η is replaced by the δ_i vertex of its bounding polytope. Solutions to these linear time-invariant (LTI) difference equations are, for every $k \in \{1, \ldots, N\}$, given by

$$\begin{aligned} x_i[k] &= A_d^k x_0 + \sum_{j=0}^{k-1} A_d^{k-1-j} \left(B_d u_i[j] + E_d \delta_i \right) \\ &= A_d^k x_0 + \sum_{j=0}^{k-1} \alpha_i[k,j] \end{aligned} \tag{4.58}$$

where for every $k \in \{1, \ldots, N\}$ and $j \in \{0, \ldots, k-1\}$

$$\alpha_i[k,j] = A_d^{k-1-j} \left(B_d u_i[j] + E_d \delta_i \right). \tag{4.59}$$

For a given sequence $\{u_i[k]\}_{k=0}^{N-1}$, the sequence $\{\alpha_i[k,j]\}_{j=0}^{k-1}$ becomes known. This fact motivates the following theorem.

Theorem 4.3. *For each $i \in \{1, \ldots, q\}$, let $\{u_i[k]\}_{k=0}^{N-1}$ be a control sequence that solves the trajectory design and control problem for the i^{th} auxiliary system. Let $\alpha_i[k,j]$ be given*

by (4.59). *If for every* $k \in \{1, \ldots, N\}$ *and* $l \in \{1, \ldots, n\}$ *there exist indices* $\curlyvee, \curlywedge \in \{1, 2, \ldots, q\}$ *such that for every* $i \in \{1, \ldots, q\}$ *and* $j \in \{0, \ldots, k-1\}$

$$\begin{aligned} \alpha^l_{\curlyvee}[k, j] &\leq \alpha^l_i[k, j] \\ \alpha^l_{\curlywedge}[k, j] &\geq \alpha^l_i[k, j] \end{aligned} \tag{4.60}$$

then the sequence $\{u[k]\}_{k=0}^{N-1}$ *with* $u[k] = u_1[k]\lambda_1[k] + \ldots + u_q[k]\lambda_q[k]$ *solves the trajectory design and control problem for the nonlinear system where* $\lambda_i[k]$ *satisfies the interpolation constraint*

$$\eta(x[k]) = \delta_1\lambda_1[k] + \ldots + \delta_q\lambda_q[k] \tag{4.61}$$

and the convex combination constraints

$$\sum_{i=1}^{q} \lambda_i[k] = 1, \quad 0 \leq \lambda_i[k] \leq 1. \tag{4.62}$$

Proof. For every $k \in I_d$, define the following quantities.

$$\begin{aligned} U[k] &= [u_1[k], \ldots, u_q[k]] \\ \Delta &= [\delta_1, \ldots, \delta_q] \\ \lambda[k] &= [\lambda_1[k], \ldots, \lambda_q[k]]^\top \end{aligned} \tag{4.63}$$

For any sequence $\{\lambda[k]\}_{k=0}^{N-1}$ satisfying (4.62), the linear system

$$x[k+1] = A_d x[k] + B_d U[k]\lambda[k] + E\Delta\lambda[k] \tag{4.64}$$

has a unique solution, for every $k \in \{1, \ldots, N\}$, given by

$$\begin{aligned} x[k] &= A_d^k x_0 + \sum_{j=0}^{k-1} A_d^{k-1-j}(BU[j] + E\Delta)\lambda[j] \\ &= A_d^k x_0 + \sum_{j=0}^{k-1} \Big[\alpha_1[k,j], \ldots, \alpha_q[k,j]\Big]\lambda[j] \\ &= A_d^k x_0 + \sum_{j=0}^{k-1} \beta[k,j] \end{aligned} \tag{4.65}$$

where for every $k \in \{1, \ldots, N\}$ and $j \in \{0, \ldots, k-1\}$

$$\beta[k,j] = \Big[\alpha_1[k,j], \ldots, \alpha_q[k,j]\Big]\lambda[j]. \tag{4.66}$$

Because $\beta[k,j]$ is defined as a convex combination of $\alpha_1[k,j], \ldots, \alpha_q[k,j]$, it is known that each component $l \in \{1, 2, \ldots, n\}$ of $\beta[k,j]$ is constrained based on (4.60) as

$$\forall k \in \{1, \ldots, N\}, \quad \forall j \in \{0, \ldots, k-1\}, \quad \alpha_\curlyvee^l[k,j] \leq \beta^l[k,j] \leq \alpha_\curlywedge^l[k,j]. \tag{4.67}$$

Consequently, their sums are ordered in the same way.

$$\forall k \in \{1, \ldots, N\}, \quad \sum_{j=0}^{k-1} \alpha_\curlyvee^l[k,j] \leq \sum_{j=0}^{k-1} \beta^l[k,j] \leq \sum_{j=0}^{k-1} \alpha_\curlywedge^l[k,j]. \tag{4.68}$$

Adding the l^{th} component of $A_d^k x_0$ to each, it follows that for every $k \in \{1, \ldots, N\}$

$$x_\curlyvee^l[k] \leq x^l[k] \leq x_\curlywedge^l[k]. \tag{4.69}$$

That is, elements of the sequence $\{x^l[k]\}_{k=0}^{k=N}$ are always between elements of the sequences $\{x_\curlyvee^l[k]\}_{k=0}^{k=N}$ and $\{x_\curlywedge^l[k]\}_{k=0}^{k=N}$. From convexity of X and X_f, it is concluded that for every $k \in \{0, \ldots, N\}$ the state $x[k] \in X$ and $x[N] \in X_f$.

Upon choosing, for every $k \in I_d$, $\lambda[k]$ such that (4.61) is satisfied and $u[k] = U[k]\lambda[k]$, the original discrete-time system (4.56) is obtained. By convexity of U, the interpolated

control $u[k] \in U$. That is, the sequence $\{u[k]\}_{k=0}^{N-1}$ solves the nonlinear trajectory design and control problem. □

Conceptually, it is convenient to think of the controls and nonlinearities as non-homogeneous forcing terms whose effects are captured in $\alpha_i[k, j]$ – see (4.59). The theorem is requiring through (4.60) that it be possible for the non-homogeneous effects to be bounded componentwise by only two of the possibly many $\alpha_i[k, j]$ terms. For this reason, (4.60) is referred as the 'ordering constraint'. It is this restriction, as detailed in the proof, that allows the nonlinear trajectories to be bounded by the linear ones thus allowing interpolation for the nonlinear controls.

Next, two techniques to design trajectories for nonlinear systems that are both based on Theorem 4.3 are introduced. The first technique sets (4.60) as a constraint in an optimization problem. Including this constraint does not add any nonlinearities to the optimization problem. However, it does require the control engineer to pre-determine what the $\curlyvee$ and $\curlywedge$ are for each l and each k. In other words, the lower and upper limits on α can vary by the element of α as well by the time instance k but must be pre-determined by the control engineer in this formulation. In the forthcoming examples, this is called the "constrained approach."

The second technique computes a trajectory for a full time domain I_d and then checks if the condition (4.60) is violated. If it is, the problem is reset at the first $k \in I_d$ where the condition is violated. Not introducing the constraint (4.60) into the optimization problem makes solving the optimization problem faster, but this technique does require computing the controller multiple times with a shrinking time horizon (similar to MPC). This technique also does not require the control engineer to pre-determine $\curlyvee$ or $\curlywedge$, which can be a challenging task especially for higher dimensional systems. The algorithm for this technique is given in Algorithm 2 and is referred to as the "resetting approach."

Algorithm 2 Resetting Approach

Require: $x_0, N, A_d, B_d, E_d, \Delta, \eta$
Ensure: u
Set $K = 0$ and $x[0] = x_0$.
while $K < N - 1$ **do**
  For each $k \in \{K, K+1, \ldots, N-1\}$ find $u_1[k], u_2[k], \ldots, u_q[k]$.
  for $k = K$ **to** $N - 1$ **do**
    For each $j \in \{K, K+1, \ldots, k\}$ compute $\alpha_1[k,j], \alpha_2[k,j], \ldots, \alpha_q[k,j]$.
    Determine $\alpha_\curlyvee[k]$ and $\alpha_\curlywedge[k]$.
    if $\exists j \in \{K, K+1, \ldots, k\}$ such that $\alpha[k,j] < \alpha_\curlyvee[k]$ **or** $\alpha[k,j] > \alpha_\curlywedge[k]$ **then**
      Set $K = k$.
      break for
    end if
    Use (4.62) to get $\lambda[k]$.
    Compute nonlinear control as $u[k] = U[k]\lambda[k]$.
    Use $u[k]$ in (4.56) to get $x[k+1]$.
  end for
end while

4.3.2 Spherically Constrained Relative Motion Trajectory Design

The approaches developed in Section 4.3.1 are now applied to a spherically constrained relative orbital motion problem. Variations of the problem have motivated a recent lossless convexification [33] and nonlinear dynamical analysis [105], which identified periodic and chaotic motion.

The relative motion of two spacecraft in proximity to each other in low earth orbit is described by the Clohessy-Wiltshire (CW) equations [106] as

$$\begin{aligned} \dot{y} &= v \\ \dot{v} &= M_1 y + M_2 v + \tau \end{aligned} \tag{4.70}$$

where $y \in \mathbb{R}^3$ is the relative position of the spacecraft in the local vertical local horizontal (LVLH) frame, $v \in \mathbb{R}^3$ is the relative velocity of the spacecraft in the LVLH frame, and

$\tau \in \mathbb{R}^3$ is the thrust-to-mass ratio. The matrices M_1 and M_2 are

$$M_1 = \begin{bmatrix} 3\omega^2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\omega^2 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 2\omega & 0 \\ -2\omega & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.71}$$

where the constant $\omega \in \mathbb{R}$ is the mean motion of the reference orbit. To prevent the vehicles from colliding while keeping them close together, the relative position is constrained to a sphere of radius R, i.e., $||y|| = R$. Because of this constraint, use of a spherical coordinate system is convenient. The transformation from Cartesian to spherical coordinates is given as

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = r \begin{bmatrix} \cos(\phi)\cos(\theta) \\ \cos(\phi)\sin(\theta) \\ \sin(\phi) \end{bmatrix} \tag{4.72}$$

where $r \in \mathbb{R}$ is the radial distance of the spacecraft from the target space object, which in the case of a spherical constraint is constant and equal to R. The spherical angles are $\phi \in (-\pi/2, \pi/2)$ and $\theta \in [0, 2\pi]$. An illustration of the coordinate transformation is shown in Figure 4.14.

The thrust-to-mass ratio may be transformed from Cartesian to spherical coordinates as

$$\begin{bmatrix} u_r \\ u_\theta \\ u_\phi \end{bmatrix} = \begin{bmatrix} \cos(\phi)\cos(\theta) & \cos(\phi)\sin(\theta) & \sin(\phi) \\ -\sin(\theta) & \cos(\theta) & 0 \\ -\sin(\phi)\cos(\theta) & -\sin(\phi)\sin(\theta) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}. \tag{4.73}$$

Using the the defined transformations, the CW dynamics in the spherical coordinate system are

$$\ddot{r} = r\dot{\phi}^2 + r\omega^2\left(3\cos(\phi)^2\cos(\theta)^2 - \sin(\theta)^2\right) + r\dot{\theta}^2\cos(\phi)^2 + 2\omega r\dot{\theta}\cos(\phi)^2 + u_r \tag{4.74a}$$

$$\ddot{\theta} = 2(\dot{\theta} + \omega)(\dot{\phi}\tan(\phi) - \dot{r}/r) - 3\omega^2\sin(\theta)\cos(\theta) + u_\theta/\left(r\cos(\phi)\right) \tag{4.74b}$$

$$\ddot{\phi} = -2\dot{\phi}\dot{r}/r - \sin(2\phi)(\dot{\theta} + \omega)^2/2 - 3\omega^2\sin(\phi)\cos(\phi)\cos^2(\theta) + u_\phi/r. \tag{4.74c}$$

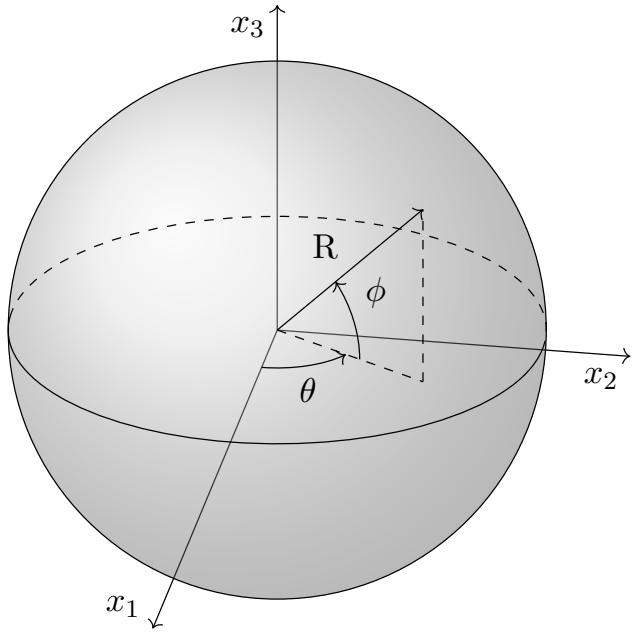


Fig. 4.14: Coordinate Transformation of Relative Coordinates.

Keeping in mind that the constraint is to stay on a spherical surface centered at the target space object, it is known that $r = R$, $\dot{r} = 0$, and $\ddot{r} = 0$. Using this information, (4.74a) can be solved for the radial control u_r to get

$$u_r = -R\left(\dot{\phi}^2 + \omega^2(3\cos(\phi)^2\cos(\theta)^2 - \sin(\theta)^2) + \dot{\theta}^2\cos(\phi)^2 + 2\omega\dot{\theta}\cos(\phi)^2\right). \tag{4.75}$$

The spherical controls u_θ and u_ϕ may then take any values, and the relative distance between the spacecraft will remain R. Also since the $\ddot{r}$ equation is constant, it does not need to be considered for the dynamics and the simplified equations for $\ddot{\theta}$ and $\ddot{\phi}$ with $r = R$ and $\dot{r} = 0$ are

$$\begin{aligned}\ddot{\theta} &= 2(\dot{\theta} + \omega)\dot{\phi}\tan(\phi) - 3\omega^2\sin(\theta)\cos(\theta) + u_\theta/\left(R\cos(\phi)\right)\\ \ddot{\phi} &= -\sin(2\phi)(\dot{\theta} + \omega)^2/2 - 3\omega^2\sin(\phi)\cos(\phi)\cos^2(\theta) + u_\phi/R.\end{aligned} \tag{4.76}$$

Now if a new control vector is defined as $u = [u_\theta/\cos(\phi), u_\phi]^\top$ and a state vector as $x = [\theta, \phi, \dot{\theta}, \dot{\phi}]^\top$, the above equations can be written in the following form

$$\begin{aligned}\dot{x} = &\begin{bmatrix} 0_{2\times2} & I_{2\times2} \\ 0_{2\times2} & 0_{2\times2} \end{bmatrix} x + \frac{1}{R}\begin{bmatrix} 0_{2\times2} \\ I_{2\times2} \end{bmatrix} u \\ &+ \begin{bmatrix} 0_{2\times2} \\ 2(\dot{\theta}+\omega)\dot{\phi}\tan(\phi) - 3\omega^2\sin(\theta)\cos(\theta) \\ -\sin(2\phi)(\dot{\theta}+\omega)^2/2 - 3\omega^2\sin(\phi)\cos(\phi)\cos^2(\theta) \end{bmatrix}.\end{aligned} \tag{4.77}$$

This nonlinear system is in the form of (4.54) with the system matrices being

$$A = \begin{bmatrix} 0_{2\times2} & I_{2\times2} \\ 0_{2\times2} & 0_{2\times2} \end{bmatrix}, \quad B = \frac{1}{R}\begin{bmatrix} 0_{2\times2} \\ I_{2\times2} \end{bmatrix}, \quad E = \begin{bmatrix} 0_{2\times2} \\ I_{2\times2} \end{bmatrix} \tag{4.78}$$

and the nonlinear function η given by

$$\eta(\theta, \phi, \dot{\theta}, \dot{\phi}) = \begin{bmatrix} 2(\dot{\theta}+\omega)\dot{\phi}\tan(\phi) - 3\omega^2\sin(\theta)\cos(\theta) \\ -\sin(2\phi)(\dot{\theta}+\omega)^2/2 - 3\omega^2\sin(\phi)\cos(\phi)\cos^2(\theta) \end{bmatrix}. \tag{4.79}$$

For numerical purposes, the radius R of the sphere is 100 m and the mean motion ω is 4 rad/hr, which corresponds to a low earth orbit with period $\pi/2$ hr. The dynamics in (4.77) are discretized according to the procedure in Section 2.3.1 with $\Delta t = 0.05$ hr to get

$$A_d = \begin{bmatrix} 1 & 0 & 0.05 & 0 \\ 0 & 1 & 0 & 0.05 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \; B_d = \begin{bmatrix} 0.0125 & 0 \\ 0 & 0.0125 \\ 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \times 10^{-3}, \; E_d = \begin{bmatrix} 0.00125 & 0 \\ 0 & 0.00125 \\ 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}. \tag{4.80}$$

The initial relative position of the spacecraft is defined as $\theta_0 = 3\pi/4$ rad and $\phi_0 = \pi/4$ rad with initial relative velocity of zero. The desired final relative position of the spacecraft is $\theta_f = \phi_f = 0$ rad and the desired final relative velocity is zero. This maneuver is to be

completed in 6 hours. With the chosen time step 0.05 hours, N becomes 120. The vertices for the bounding polytope of the nonlinearity are chosen as (all in rad/hr^2)

$$\delta_1 = \begin{bmatrix} -35 \\ -42 \end{bmatrix}, \quad \delta_2 = \begin{bmatrix} -35 \\ +42 \end{bmatrix}, \quad \delta_3 = \begin{bmatrix} +35 \\ -42 \end{bmatrix}, \quad \delta_4 = \begin{bmatrix} +35 \\ +42 \end{bmatrix}. \tag{4.81}$$

With this in place, Theorem 4.3 may be applied to generate feasible solutions to the nonlinear trajectory design and control problem. First, the "constrained approach" is used and then the "resetting approach." Both of the solutions are compared against a solution that was achieved by solving a non-convex optimization problem using MATLAB's [91] `fmincon` solver. The "constrained" and "resetting" solutions are used as initial guesses for the non-convex solver.

Constrained Approach

In this approach, the constraints in (4.60) are incorporated into an optimization problem. The second-order cone programming (SOCP) problem to be solved is

$$\min_{u} \quad \sum_{i=1}^{4} \left(\sum_{k=0}^{N-1} 10^{-12} \|u_i[k]\|^2 + \|x_i[N] - x_f\|^2 \right) \tag{4.82a}$$

$$\text{s.t.} \quad x_i[k+1] = A_d x_i[k] + B_d u_i[k] + E_d \delta_i, \; i = 1, \ldots, 4, \; k = 0, \ldots, N-1 \tag{4.82b}$$

$$x_i[0] = x_0, \; i = 1, \ldots, 4 \tag{4.82c}$$

$$\text{Equation (4.60) with} \curlyvee = [4,4,4,4]^\top \text{ and } \curlywedge = [2,3,2,3]^\top, k = 1, \ldots, N \tag{4.82d}$$

The cost function in (4.82a) has two quadratic terms. The first term penalizes the control inputs of the auxiliary LTI systems. This term is not needed; it serves to regularize or smooth the resulting solutions. The second term penalizes error in the final state. Weights multiplying the terms have been chosen based on their magnitudes. The control inputs for the auxiliary LTI systems are on the order of 10^4 whereas the final state error gets very close to zero so the multiplier for the final state error is chosen to be significantly larger than that for the control inputs. The dynamics of the auxiliary LTI systems are enforced

in (4.82b) with initial conditions in (4.82c). The ordering constraint (4.60) of Theorem 4.3 is enforced in (4.82d). Note that other orderings can be used, and it is not required for one ordering be enforced at all k. The user is free to choose the ordering.

Solving the optimization problem (4.82) took 1.14 seconds on a laptop with 2.30 GHz Intel i7 processor using Gurobi [90] in MATLAB [91] through YALMIP [92]. The same setup was used for all the computations in the following sections as well.

Figure 4.15 shows the relative angular displacement trajectories of the spacecraft and Figure 4.16 shows the relative angular velocity trajectories of the spacecraft. In the figures, the linear system trajectories are covered by the actual nonlinear system trajectories, but the nonlinear system trajectory is between the upper and lower bounds of the linear system trajectories. From inspection of the magnified inset, it is evident that the final angular displacements are on the order of 10^{-4} rad corresponding to a final position error on the centimeter level.

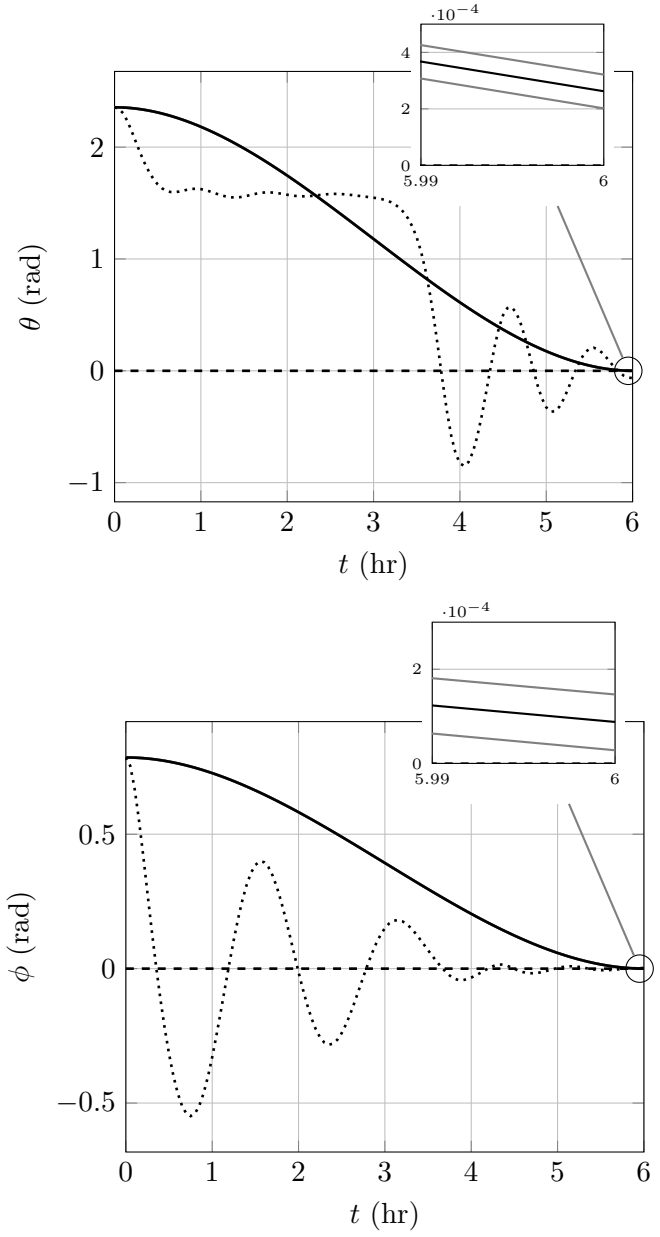


Fig. 4.15: Angular displacement trajectories of the spacecraft. The black curves are the angular displacements of the actual nonlinear system whereas the gray curves are the angular displacements of the auxiliary linear systems. The dashed line is the desired final position and the dotted line a solution using a non-convex solver.

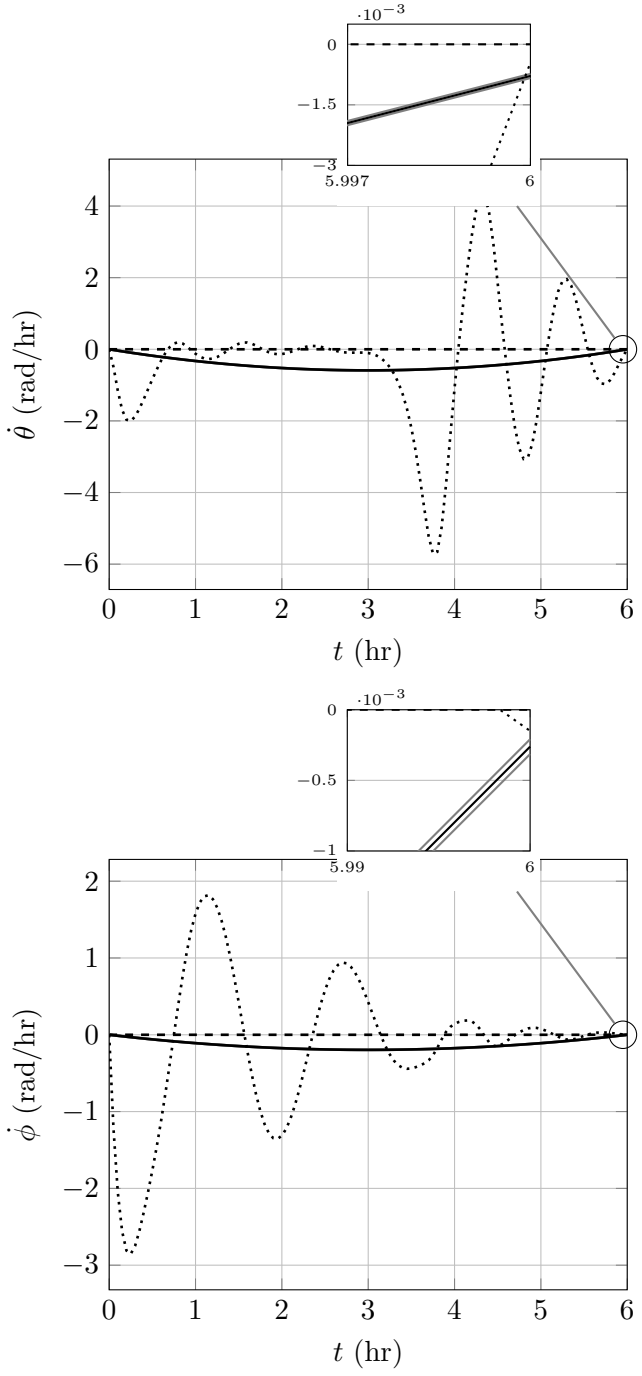


Fig. 4.16: Angular velocity trajectories of the spacecraft. The black curves are the angular velocities of the actual nonlinear system whereas the gray curves are the angular velocities of the auxiliary linear systems. The dashed line is the desired velocity at final time and the dotted line a solution using a non-convex solver.

Resetting Approach

In this section, the constrained relative motion problem is solved using the "resetting approach" described in Algorithm 2. The SOCP problem used to find control solutions in Line 3 of Algorithm 2 is the same as (4.82) but with (4.82d) removed. The resets are done as described in Algorithm 2.

The results are shown in Figures 4.17 and 4.18. The results are similar to those shown in Figures 4.15 and 4.16 except that towards the end of the time horizon, the angular velocity trajectories of the linear systems diverge from the desired final velocity. The noticeable difference is that designing this controller requires a solution to 11 SOCP problems as there were 10 resets due to violations of the ordering constraint in (4.60). On average, solving each problem required 0.17 seconds for total solve time of 1.87 seconds. The number of variables in each optimization problem also reduces on each reset as the time horizon gets shorter.

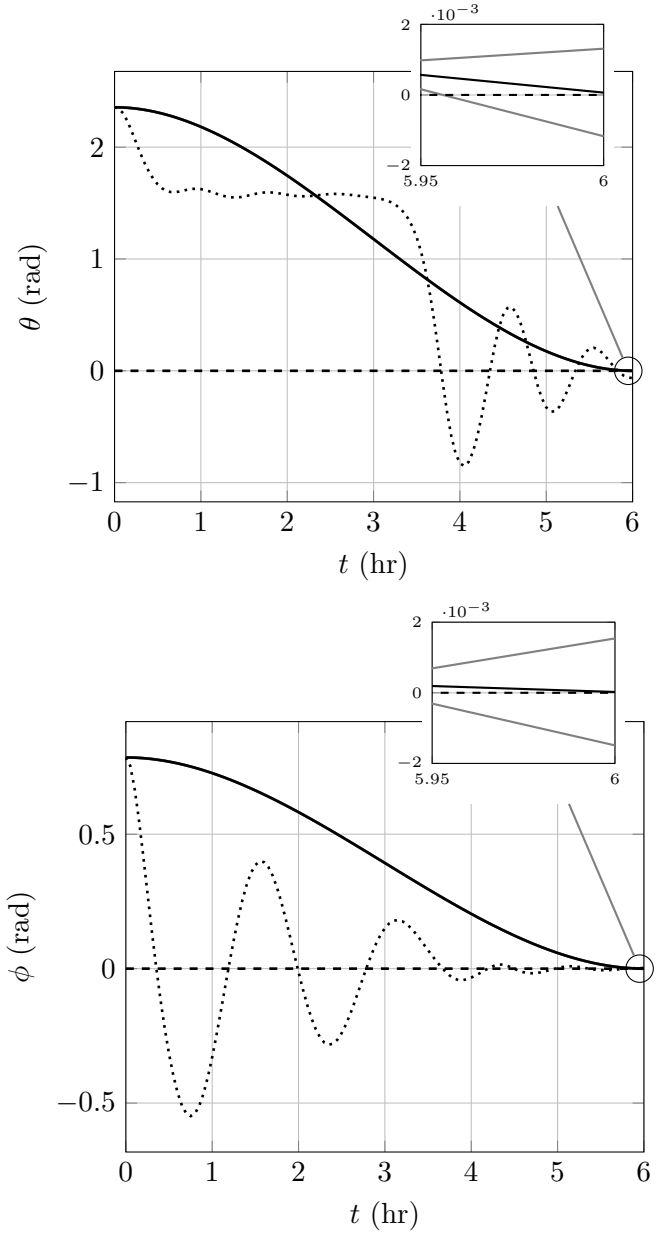


Fig. 4.17: Angular displacement trajectories of the spacecraft. The black curves are the angular displacements of the actual nonlinear system whereas the gray curves are the angular displacements of the auxiliary linear systems. The dashed line is the desired final position and the dotted line a solution using a non-convex solver.

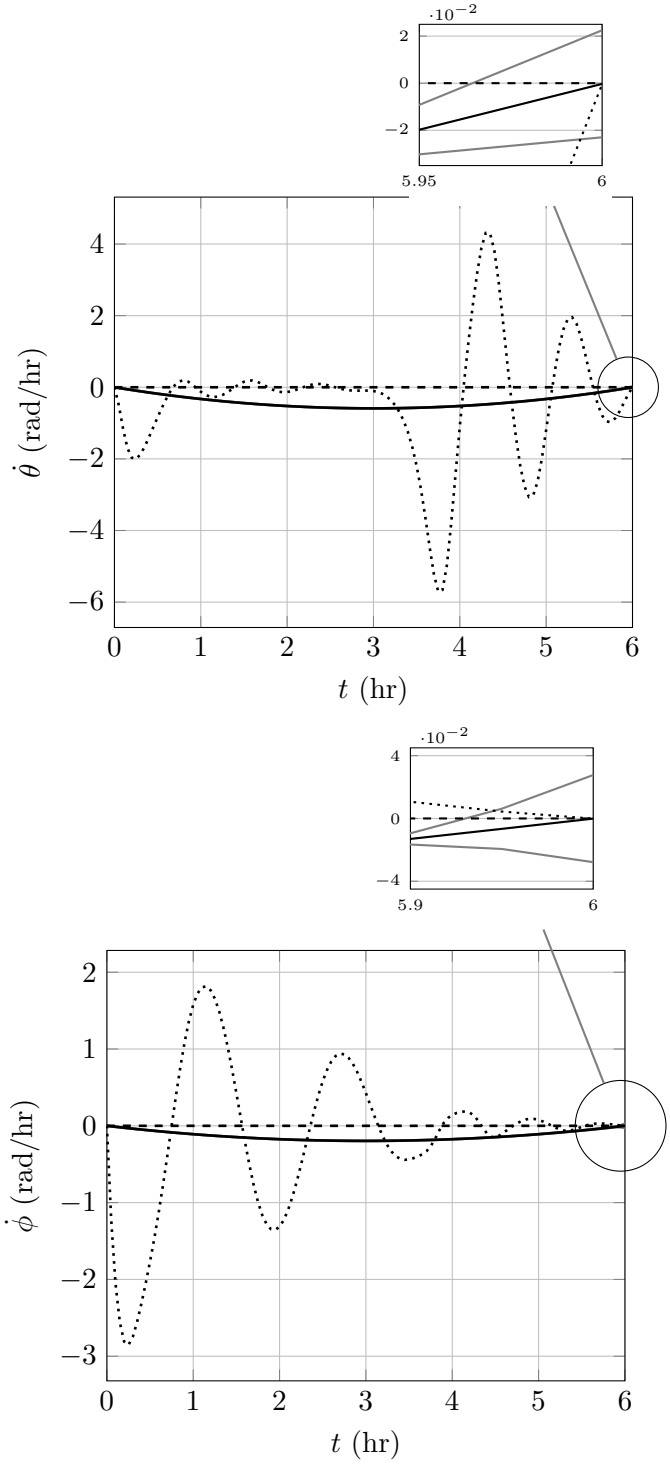


Fig. 4.18: Angular velocity trajectories of the spacecraft. The black curves are the angular velocities of the actual nonlinear system whereas the gray curves are the angular velocities of the auxiliary linear systems. The dashed line is the desired velocity at final time and the dotted line a solution using a non-convex solver.

4.3.3 Spacecraft Attitude Control

The approaches developed in Section 4.3.1 are now applied to a spacecraft attitude control problem. To represent the attitude of a spacecraft, a rotation vector $\theta = \alpha e \in \mathbb{R}^3$ is used. The unit vector $e \in \mathbb{R}^3$ is the Euler rotation axis and $\alpha \in \mathbb{R}$ is the angular displacement. To derive the kinematics of the spacecraft attitude, a quaternion $q \in \mathbb{R}^4$ is defined representing the spacecraft attitude as [107]

$$q = \begin{bmatrix} \frac{1}{\alpha}\sin(\alpha/2)\theta \\ \cos(\alpha/2) \end{bmatrix}. \tag{4.83}$$

Taking the time derivative leads to

$$\dot{q} = \begin{bmatrix} \frac{1}{\alpha}\sin(\alpha/2)\dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \left(-\frac{1}{\alpha^2}\sin(\alpha/2) + \frac{1}{2\alpha}\cos(\alpha/2)\right)\dot{\alpha}\theta \\ -\frac{1}{2}\sin(\alpha/2)\dot{\alpha} \end{bmatrix}. \tag{4.84}$$

It is also well-known that the quaternion kinematics can be written as [108]

$$\dot{q} = \frac{1}{2}\begin{bmatrix} q_4\omega + q_{1:3} \times \omega \\ -q_{1:3}^\top \omega \end{bmatrix} \tag{4.85}$$

$$= \frac{1}{2}\begin{bmatrix} \cos(\alpha/2)\omega - \frac{1}{\alpha}\sin(\alpha/2)\theta \times \omega \\ -\frac{1}{\alpha}\sin(\alpha/2)\theta^\top \omega \end{bmatrix} \tag{4.86}$$

where $\omega \in \mathbb{R}^3$ is the attitude rate of the spacecraft expressed in its body frame. Comparing the last element of (4.84) and (4.86) gives an expression for $\dot{\alpha}$ as $\dot{\alpha} = \alpha^{-1}\theta^\top\omega$. Comparing the first three elements of (4.84) and (4.86), substituting the expression of $\dot{\alpha}$, and solving for $\dot{\theta}$ gives

$$\dot{\theta} = \left(\frac{1}{\alpha} - \frac{1}{2}\cot(\alpha/2)\right)\frac{1}{\alpha}\theta^\top\omega\theta + \frac{1}{2}\left(\alpha\cot(\alpha/2)\omega + \theta \times \omega\right). \tag{4.87}$$

Subtracting and adding $\alpha^{-2}\theta^\top\theta\omega$ on the right hand side, using the vector triple product, and noting that $\alpha = \|\theta\|$ (provided $0 < \alpha < 2\pi$), simplifies the above to [109]

$$\dot{\theta} = \omega + \frac{1}{2}\theta \times \omega + \frac{1}{\|\theta\|^2}\left(1 - \frac{\|\theta\|}{2}\cot(||\theta||/2)\right)\theta \times (\theta \times \omega). \tag{4.88}$$

The attitude dynamics of the spacecraft can be derived from Euler's equations [108]. Assuming that the only external torque affecting the system is a control torque $\tau \in \mathbb{R}^3$, the dynamics are

$$\dot{\omega} = J^{-1}\left(\omega \times (J\omega) + \tau\right) \tag{4.89}$$

where J is moment of inertia of the spacecraft about its center of mass represented in the body frame. Defining the state vector as $x = [\theta^\top, \omega^\top]^\top \in \mathbb{R}^6$, equations (4.88) and (4.89) can be written as

$$\begin{aligned}\dot{x} = &\begin{bmatrix} 0_{3\times3} & I_{3\times2} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} x + \begin{bmatrix} 0_{3\times3} \\ J^{-1} \end{bmatrix} \tau \\ &+ \begin{bmatrix} I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & J^{-1} \end{bmatrix} \begin{bmatrix} \frac{1}{2}\theta \times \omega + \frac{1}{\|\theta\|^2}\left(1 - \frac{\|\theta\|}{2}\cot(||\theta||/2)\right)\theta \times (\theta \times \omega) \\ \omega \times (J\omega) \end{bmatrix}.\end{aligned} \tag{4.90}$$

To simplify the nonlinear effects, it is assumed that the rate vector ω is known from navigation and available for feedback linearization. Upon defining the control $\tau = u - \omega \times (J\omega)$, the nonlinear system is

$$\begin{aligned}\dot{x} = &\begin{bmatrix} 0_{3\times3} & I_{3\times2} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix} x + \begin{bmatrix} 0_{3\times3} \\ J^{-1} \end{bmatrix} u \\ &+ \begin{bmatrix} I_{3\times3} \\ 0_{3\times3} \end{bmatrix} \left[\frac{1}{2}\theta \times \omega + \frac{1}{\|\theta\|^2}\left(1 - \frac{\|\theta\|}{2}\cot(||\theta||/2)\right)\theta \times (\theta \times \omega)\right].\end{aligned} \tag{4.91}$$

Feedback linearization has the effect of reducing the dimension of the nonlinearity from six to three, which in turn reduces the dimension of the bounding convex polytope.

The nonlinear system in (4.91) is in the form of (4.54) with the system matrices being

$$A = \begin{bmatrix} 0_{3\times3} & I_{3\times2} \\ 0_{3\times3} & 0_{3\times3} \end{bmatrix}, \quad B = \begin{bmatrix} 0_{3\times3} \\ J^{-1} \end{bmatrix}, \quad E = \begin{bmatrix} I_{3\times3} \\ 0_{3\times3} \end{bmatrix} \tag{4.92}$$

and the nonlinear function η given by

$$\eta(\theta,\omega) = \frac{1}{2}\theta\times\omega + \frac{1}{\|\theta\|^2}\left(1 - \frac{\|\theta\|}{2}\cot(||\theta||/2)\right)\theta\times(\theta\times\omega). \tag{4.93}$$

For numerical purposes, the spacecraft is assumed to have a cylindrical shape with constant density. The radius of the spacecraft is $r = 0.25$ m, height $h = 0.5$ m, and mass $m = 100$ kg. The body z-axis is defined to be aligned with the axis of the cylinder. The moment of inertia matrix is then given as $J = \text{diag}\{J_{xx}, J_{yy}, J_{zz}\}$, where $J_{xx} = J_{yy} = \frac{1}{4}mr^2 + \frac{1}{12}mh^2$ and $J_{zz} = \frac{1}{2}mr^2$. The dynamics in (4.90) are discretized according to the procedure in Section 2.3.1 with $\Delta t = 0.05$ sec to get

$$A_d = \begin{bmatrix} 1 & 0 & 0 & 0.05 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.05 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.05 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_d = \begin{bmatrix} 0.343 & 0 & 0 \\ 0 & 0.343 & 0 \\ 0 & 0 & 0.4 \\ 13.7 & 0 & 0 \\ 0 & 13.7 & 0 \\ 0 & 0 & 16 \end{bmatrix} \times 10^{-3}, \tag{4.94}$$

$$E_d = \begin{bmatrix} 0.05 & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 0.05 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The initial attitude of the spacecraft is given with an axis of rotation of $\frac{1}{\sqrt{3}}[1,1,1]^\top$ and angle 45°. The desired final attitude is given by an axis of rotation of $-\frac{1}{\sqrt{3}}[1,1,1]^\top$ and angle 90°. The initial angular velocity of the spacecraft is $[5,-10,15]^\circ/\text{s}$ and the desired final angular velocity is zero. The maneuver is to be done in 6 seconds. With the chosen time step of 0.05 seconds, N becomes 120. The vertices for the bounding polytope of the nonlinearity are chosen as (all in rad/s)

$$\begin{aligned}
\delta_1 &= \begin{bmatrix} -0.1 \\ -0.1 \\ -0.1 \end{bmatrix}, \quad \delta_2 = \begin{bmatrix} -0.1 \\ -0.1 \\ +0.1 \end{bmatrix}, \quad \delta_3 = \begin{bmatrix} -0.1 \\ +0.1 \\ -0.1 \end{bmatrix}, \quad \delta_4 = \begin{bmatrix} -0.1 \\ +0.1 \\ +0.1 \end{bmatrix}, \\
\delta_5 &= \begin{bmatrix} +0.1 \\ -0.1 \\ -0.1 \end{bmatrix}, \quad \delta_6 = \begin{bmatrix} +0.1 \\ -0.1 \\ +0.1 \end{bmatrix}, \quad \delta_7 = \begin{bmatrix} +0.1 \\ +0.1 \\ -0.1 \end{bmatrix}, \quad \delta_8 = \begin{bmatrix} +0.1 \\ +0.1 \\ +0.1 \end{bmatrix}.
\end{aligned} \tag{4.95}$$

Again, Theorem 4.3 may be applied to generate feasible solutions to the nonlinear trajectory design and control problem. The "constrained" and "resetting" approaches are compared against a solution that was achieved by solving a non-convex optimization problem using MATLAB's [91] `fmincon` solver with initial guesses gotten from the "constrained" and "resetting" solutions.

Constrained Approach

In this approach, the constraints in (4.60) are incorporated into an optimization problem. The second-order cone programming (SOCP) problem to be solved is

$$\min_{u} \quad \sum_{i=1}^{8} \left(\sum_{k=0}^{N-1} 10^{-8} \|u_i[k]\|^2 + \|x_i[N] - x_f\|^2 \right) \tag{4.96a}$$

$$\text{s.t.} \quad x_i[k+1] = A_d x_i[k] + B_d u_i[k] + E_d \delta_i, \ i = 1, \ldots, 8, \ k = 0, \ldots, N-1 \tag{4.96b}$$

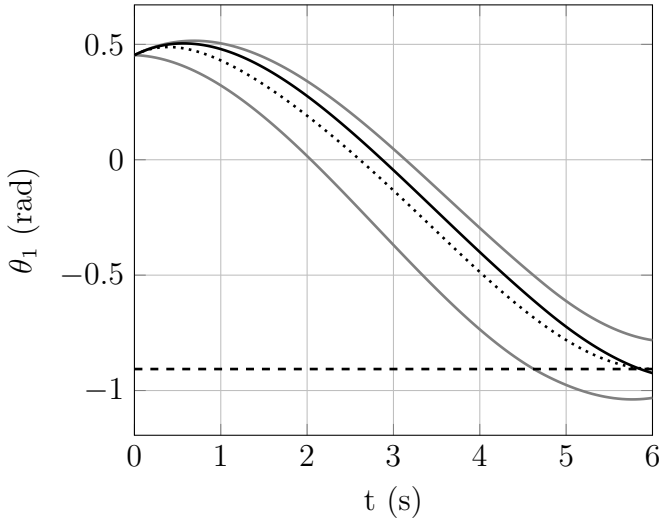
$$x_i[0] = x_0, \ i = 1, \ldots, 8 \tag{4.96c}$$

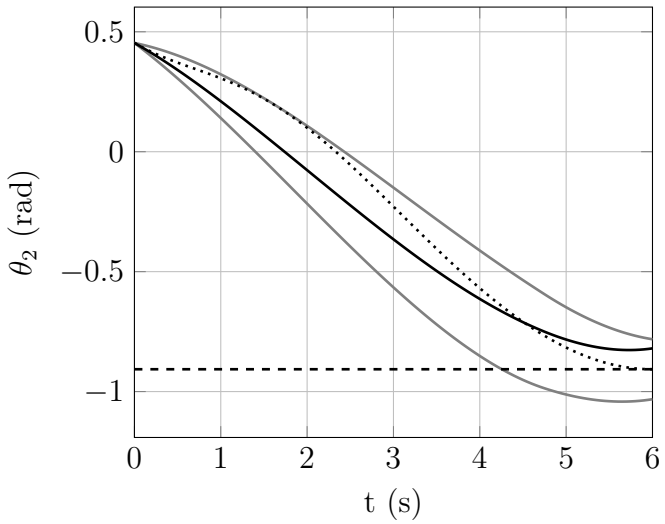
$$\text{Equation (4.60) with} \curlyvee = [4, 6, 7, 8, 8, 8]^\top, k = 1, \ldots, N \tag{4.96d}$$

$$\curlywedge = [8, 8, 8, 4, 6, 7]^\top, k = 1, \ldots, N$$

Due to different scaling between the auxiliary system control magnitudes and errors between actual and desired final states compared to the relative motion problem of the previous section, the weight on the control term is set to 10^{-8}. As before, the control penalty is not needed but has a regularizing effect. The dynamics of the auxiliary LTI systems are enforced in (4.96b) with initial conditions in (4.96c). The ordering constraint (4.60) of Theorem 4.3 is enforced in (4.96d).

Solving the SOCP problem took 1.09 seconds. Figure 4.19 shows the attitude of the spapecraft with the dashed line as the desired final attitude. Figure 4.20 shows the angular velocity of the spacecraft. It can be seen in these figures that the nonlinear system does not get exactly to the desired final state but stays between the upper and lower bounding linear systems. The magnitude of error between the desired final rotation vector and the final rotation vector of the nonlinear system is 0.107 rad. However, the average magnitude of error between the desired final rotation vector and the final rotation vectors of the linear auxiliary systems is 0.217 rad. Since the linear auxiliary systems provide the limits for the nonlinear system trajectories, better performance is desired and is investigated as part of the "resetting approach."


0.5
0
−0.5
−1
θ_1 (rad)
0
1
2
3
4
5
6
t (s)


0.5
0
−0.5
−1
θ_2 (rad)
0
1
2
3
4
5
6
t (s)

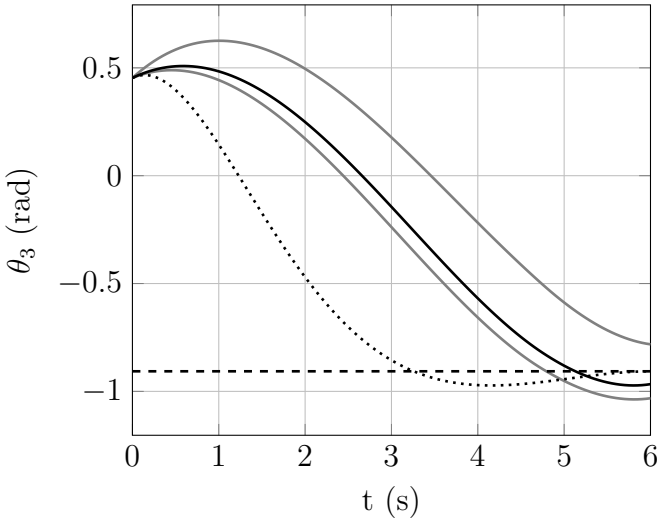
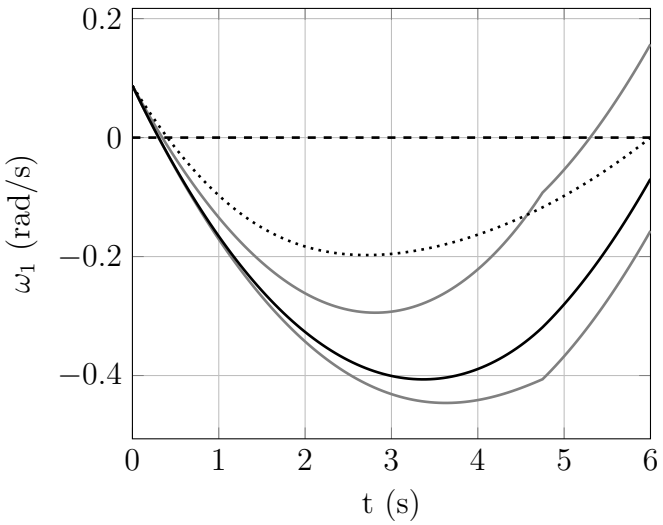


Fig. 4.19: Attitude of the spacecraft. The black curves are the attitude of the actual nonlinear system whereas the gray curves are the attitudes of the auxiliary linear system. The dashed line is the desired final attitude and the dotted line a solution using a non-convex solver.



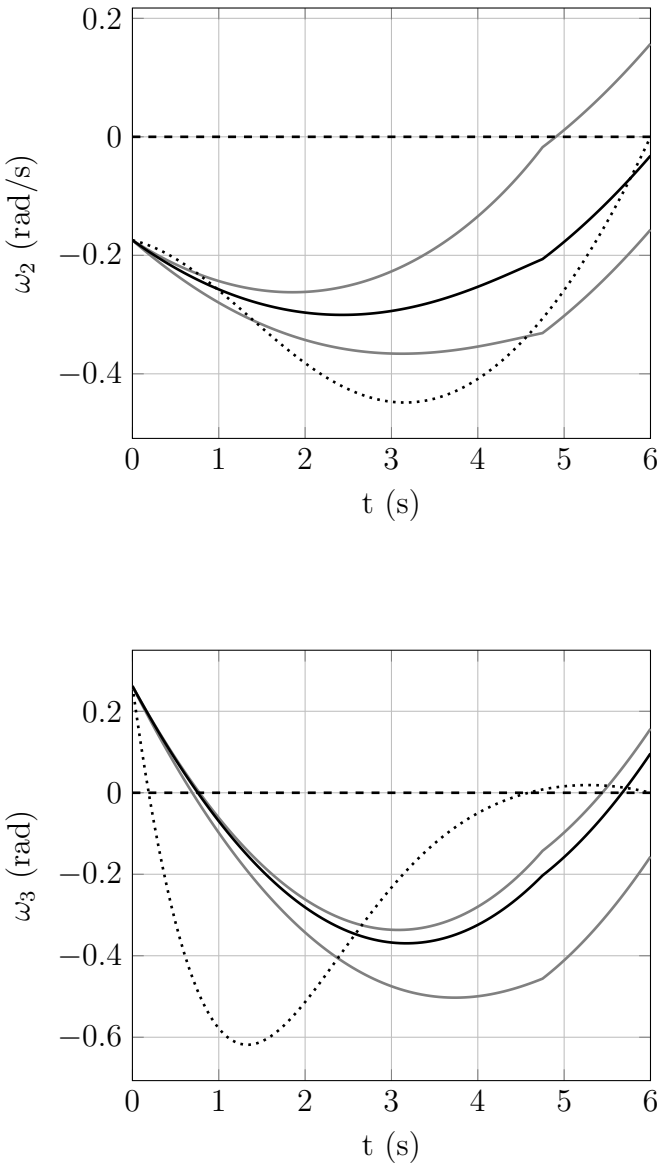
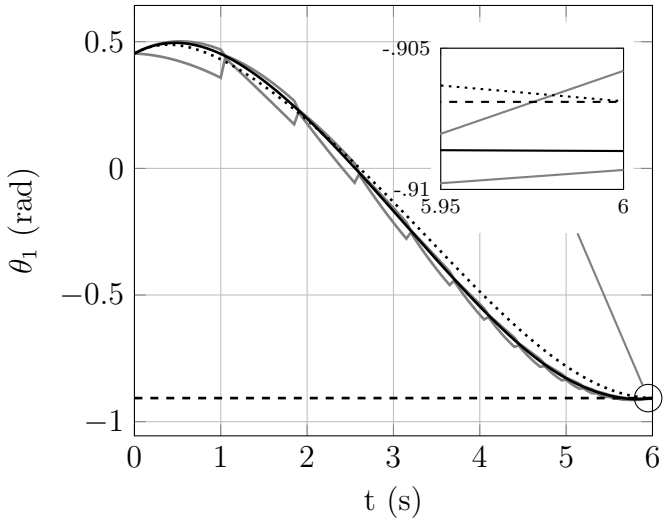


Fig. 4.20: Angular velocity of the spacecraft. The black curves are the angular velocities of the actual nonlinear system whereas the gray curves are the angular velocities of the auxiliary linear systems. The dashed line is the desired angular velocity and the dotted line a solution using a non-convex solver.

Resetting Approach

In this section, the attitude control problem is solved using the "resetting approach" described in Algorithm 2. The SOCP problem used to find control solutions in Line 3 of Algorithm 2 is the same as (4.96) but with (4.96d) removed. The resets are done as described in Algorithm 2.

Figure 4.21 shows the attitude of the spacecraft with the dashed line as the desired final attitude. Figure 4.22 shows the angular velocity of the spacecraft. It can be seen on these figures that the resetting controller seems to perform significantly better than the constrained controller in the previous section especially when comparing the errors between the desired final attitude and the actual final attitude. The magnitude of the error between the desired final rotation vector and the final rotation vector of the nonlinear system is 2.17×10^{-3} rad. Even the average magnitude of the error between the desired final rotation vector and the final rotation vectors of the auxiliary systems is only 2.9×10^{-3} rad. These are significantly lower compared to the final attitude errors reached with constrained controller. Design of this controller required solving 20 SOCP problems with an average solution time of 0.233 seconds for a total solver time of 4.67 seconds. The resets are also easily noticeable with the saw blade like linear system trajectories.



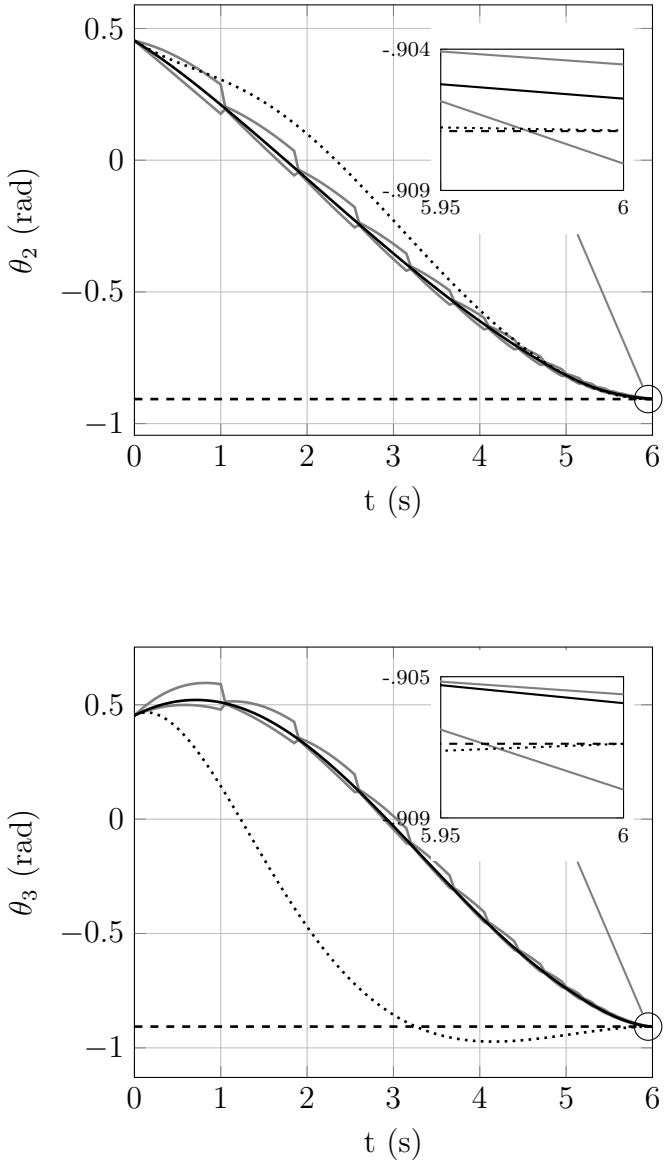
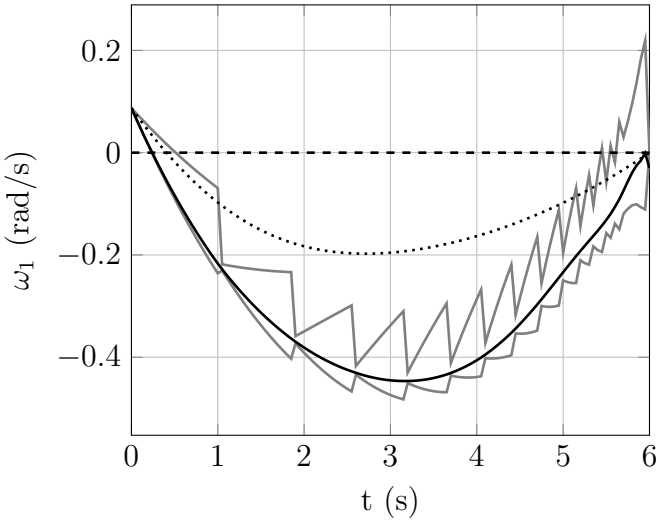
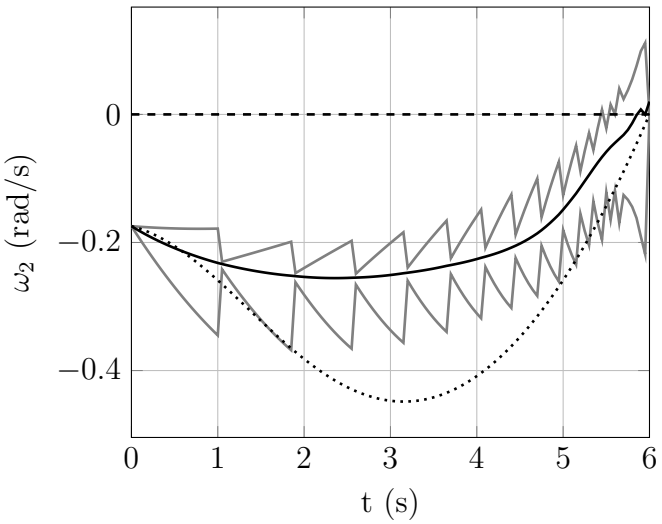


Fig. 4.21: Attitude of the spacecraft. The black curves are the attitude of the actual nonlinear system whereas the gray curves are the attitudes of the auxiliary linear systems. The dashed line is the desired final attitude and the dotted line a solution using a non-convex solver.


0.2
0
−0.2
−0.4
ω_1 (rad/s)
0
1
2
3
4
5
6
t (s)


0
−0.2
−0.4
ω_2 (rad/s)
0
1
2
3
4
5
6
t (s)

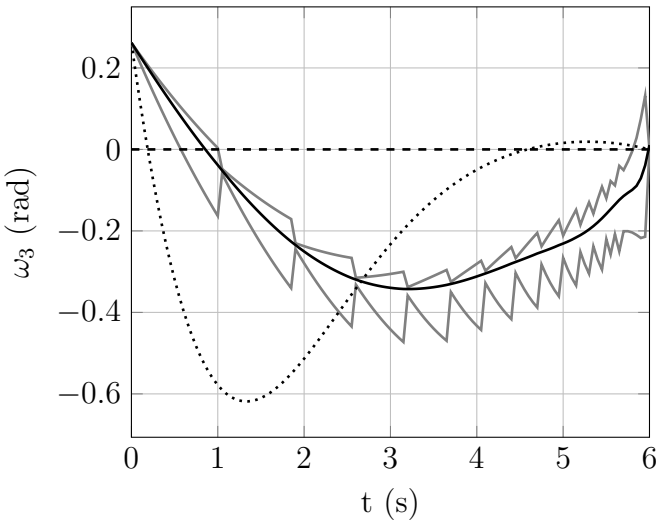


Fig. 4.22: Angular velocity of the spacecraft. The black curves are the angular velocity of the actual nonlinear system whereas the gray curves are the angular velocities of the auxiliary linear systems. The dashed line is the desired angular velocity and the dotted line a solution using a non-convex solver.

4.4 Computational Method of Controlling Convex Polytope Bounded Nonlinear Systems with Comparison to Feedback Linearization

This section describes an algorithm to control nonlinear systems whose nonlinearity is known to take values in a convex polytope. The algorithm relies upon solving a finite set of linear programs and reconstructing the nonlinear control from these solutions. Though motivated by continuous-time applications, the algorithm relies upon a time discretization and computes controls at those discrete time instances. The algorithm bears resemblance to feedback linearization and model predictive control (MPC). Examples indicate that it performs better in cases when the feedback linearization domain does not coincide with the spatial domain of interest. Furthermore, the algorithm uses optimization, but the resulting nonlinear control solutions are not optimal in any pre-defined sense.

The following is a brief outline of the section. Section 4.4.1 explains some of the mathematical nomenclature used throughout the rest of the section. Section 4.4.2 states the problem of interest and required assumptions. Section 4.4.3 gives requisite background on feedback linearization. Section 4.4.4 presents the algorithm of this section, which involves the solution of a finite set of linear programming problems. Section 4.4.5 provides two examples analyzed in MATLAB [91] and compares the results to feedback linearization.

The primary contribution of this section is Algorithm 3 on page 105. The algorithm is a technique for controlling nonlinear systems whose nonlinearity takes values in a convex polytope. It solves a linear program analytically at each discrete time instance.

4.4.1 Nomenclature for the Section

The following is a partial list of notation used throughout this chapter. $\mathbb{R}$ is the set of real numbers. $\mathbb{R}^n$ is the set of real n-tuples. The time derivative of a function is denoted with an over-dot, i.e. $dx(t)/dt = \dot{x}(t)$. Identity matrix with appropriate dimension is denoted by I. An element-wise inequality is denoted by $a \leq b$. The transpose is denoted with a superscript $\top$. A map x is identically equal to constant on an interval T if $\forall t \in T, x(t) = c$ where c is a constant.. This is denoted by $x \equiv c$. The gradient of a map $f : \mathbb{R}^n \to \mathbb{R}^m$ with respect to x is denoted by $\frac{\partial f}{\partial x} : \mathbb{R}^n \to \mathbb{R}^{m \times n}$. The Lie derivative is denoted by $L_f g\,(x) =$

$\frac{\partial g}{\partial x} f(x)$. The following notations are introduced for Lie derivatives $L_f^k g(x) = \frac{\partial L_f^{k-1} g}{\partial x} f(x)$ and $L_f^0 g(x) = g(x)$. The lie bracket is denoted as $[f,g](x) = L_f g(x) - L_g f(x)$ with the following additional notation $ad_f^0 g(x) = g(x)$ and $ad_f^k g(x) = \left[f, ad_f^{k-1} g\right](x)$ for $k \geq 1$. For discrete dynamical systems the upperscipt denotes the sampling occurrence (e.g. x^{k+1} is sampled one sampling period after x^k).

4.4.2 Problem Statement

Throughout this section a nonlinear system of the form

$$\dot{x}(t) = Ax(t) + Bu(t) + f\big(x(t)\big) \tag{4.97}$$

is considered. The system state is $x : [t_0, t_f] \to \mathcal{D} \subset \mathbb{R}^n$. The control input is $u : [t_0, t_f] \to \mathbb{R}^m$ and belongs to the set of piecewise continuous functions. The linear system matrix is $A \in \mathbb{R}^{n \times n}$. The linear control matrix is $B \in \mathbb{R}^{n \times m}$. The nonlinear element of the system is captured in the function $f : \mathcal{D} \to \mathbb{R}^n$. It is assumed that the system is Lipschitz in x $\forall t \in [t_0, t_f]$ and $x \in \mathcal{D}$, the linear portion (A, B) of the system is controllable and the nonlinearity $f\big(x(t)\big)$ takes values in a convex polytope. These dynamics are discretized to get

$$x^{k+1} = \bar{A}x^k + \bar{B}u^k + \bar{D}f\left(x^k\right) \tag{4.98}$$

where $k = 1, 2, \ldots, N-1$, $N = (t_f - t_0)/T$, $\bar{A} = e^{AT}$, $\bar{B} = \int_0^T e^{A\tau} d\tau B$ and $\bar{D} = \int_0^T e^{A\tau} d\tau$. Here T is sampling period and a zero-order hold for u and $f(x)$ is assumed. Since $f(x)$ is continuous in time the zero-order hold is invalid. However, this assumption is made in practical feedback linearization as measurements and estimates of state variables are received discretely in time. The control objective is to regulate the state to the origin in finite time.

For the continuous dynamics in (4.97) with $f \equiv c$ where c is a consant in $\mathbb{R}^n$, a continuous-time control law can be synthesized by solving the following finite horizon linear-quadratic regulator (LQR) problem.

$$\begin{aligned} \min_{u} \quad & \frac{1}{2}\int_{t_0}^{t_f}\left(e^{\top}(t)\,Qe(t)+u^{\top}(t)\,Ru(t)\right)dt \\ & +\frac{1}{2}e^{\top}(t_f)\,Se(t_f) \\ \text{s.t.} \quad & \text{Equation (4.97)} \\ & e(t)=x(t)-x_f \\ & x(t_0)=x_0 \end{aligned} \tag{P4.0}$$

The fixed initial time is t_0 and the known initial state is x_0. The fixed final time is t_f and desired final state x_f. The matrices $Q, S \in \mathbb{R}^{n\times n} \succeq 0$ and the matrix $R \in \mathbb{R}^{m\times m} \succ 0$.

In similar fashion with $f \equiv c$, a discrete-time control can be synthesized by solving the following LQR problem.

$$\begin{aligned} \min_{u^k} \quad & \frac{1}{2}\sum_{k=1}^{N-1}\left(\left(e^k\right)^{\top}Qe^k+\left(u^k\right)^{\top}Ru^k\right) \\ & +\frac{1}{2}\left(e^N\right)^{\top}Se^N \\ \text{s.t.} \quad & \text{Equation (4.98)} \\ & e^k=x^k-x_f, \quad k=1,2,\ldots,N \\ & x^1=x_0 \end{aligned} \tag{P4.1}$$

The resulting LQR laws will be used in Section 4.4.4 to construct a nonlinear control for certain nonlinearities.

4.4.3 Feedback Linearization

An input-state feedback linearization technique for a single input system is discussed here. Feedback linearization is presented because it serves as the primary comparison case for the proposed algorithm.

Definition 4.4. *[2] Let f_1 and f_2 be vector fields. A distribution Σ is involutive if*

$$f_1(x) \in \Sigma \quad \textit{and} \quad f_2(x) \in \Sigma \quad \Longrightarrow [f_1, f_2](x) \in \Sigma$$

Lemma 4.3. *[2] A nonsingular distribution $\Sigma = span\{f_1, f_2, \ldots, f_n\}$ on domain $\mathcal{D}$ is involutive if and only if $[f_i, f_j] \in \Sigma, \quad \forall 1 \leq i, j \leq n$.*

Proof. Necessity is obvious as it follows directly from Definition 4.4. For sufficiency, let g_1 and g_2 be any two vector fields in Σ. Then they can be expressed as

$$g_1(x) = \sum_{i=1}^{n} \alpha_i(x) f_i(x), \quad g_2(x) = \sum_{i=1}^{n} \beta_i(x) f_i(x)$$

The Lie bracket of the vector fields g_1 and g_2 is

$$\begin{aligned}
[g_1, g_2](x) &= \left[\sum_{i=1}^{n} \alpha_i f_i, \sum_{i=1}^{n} \beta_i f_i\right](x) \\
&= \sum_{i=1}^{n}\sum_{j=1}^{n} \Big(\alpha_i(x)\beta_j(x)[f_i, f_j](x) + \\
&\qquad \alpha_i(x) L_{f_i} f_j(x) - \beta_j(x) L_{f_j} f_i(x)\Big)
\end{aligned}$$

Since $[f_i, f_j](x) \in \Sigma$, it follows that $[g_1, g_2](x) \in \Sigma$. □

As mentioned earlier, feedback linearization is one of the most common techniques to design controllers for nonlinear systems. One of the shortcomings of feedback linearization is that it is not always possible or it may only be applicable on a restricted domain.

Theorem 4.4. *[2] For (4.97), let $g(x) = Ax + f(x)$ and the control input u to be a scalar. Then the system in (4.97) is feedback linearizable on $\mathcal{D}_0$ if and only if there exist a domain $\mathcal{D}_0 \subset \mathcal{D}$ such that*

1. *matrix $G(x) = \left[B, \quad ad_g B(x), \quad \ldots, \quad ad_g^{n-1} B(x)\right]$ is non-singular $\forall x \in \mathcal{D}_0$.*
2. *distribution $\mathcal{E} = span\left\{B, ad_g B, \ldots, ad_g^{n-2} B\right\}$ is involutive in $\mathcal{D}_0$.*

If Theorem 4.4 holds, then it is possible to find a transformation $z = T(x)$ such that $T(x)$ is diffeomorphism on $\mathcal{D}_0$ and

$$\dot{z} = A_z z + b_z v \tag{4.99}$$

where $A_z \in \mathbb{R}^{n\times n}$, $b_z \in \mathbb{R}^n$ and $v = \gamma(x)(u + \alpha(x))$ or equivalently $u = v\gamma^{-1}(x) - \alpha(x)$. The transformation $T(x)$ is

$$T(x) = \begin{bmatrix} h(x) \\ L_g h(x) \\ L_g^2 h(x) \\ \vdots \\ L_g^{n-1} h(x) \end{bmatrix}$$

so $z_i = L_g^{i-1} h(x)$ for $i = 1, 2, \ldots, n$. This makes $\alpha(x) = L_g^n h(x) \left(L_B L_g^{n-1} h(x)\right)^{-1}$ and $\gamma(x) = L_B L_g^{n-1} h(x)$. The mapping $h(x)$ is chosen such that $L_B L_g^{i-1} h(x) = 0$, $\forall i = 1, 2, \ldots, n-1$, $\gamma(x) \neq 0$ and $h(x_f) = 0$ where x_f is a desired final state.

4.4.4 Proposed Linearization Technique

This subsection describes the proposed linearization technique which takes advantage of convex combination of vertices of a polytope that bounds the nonlinearity in (4.97). Let the columns of matrix $\Delta = \begin{bmatrix} \Delta_1, & \Delta_2, & \ldots, & \Delta_q \end{bmatrix}$, $\Delta_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, q$ be the vertices of a bounding polytope for the nonlinearity in (4.97). The nonlinear system can then be represented as a linear equation

$$\dot{x}(t) = Ax(t) + Bu(t) + \Delta\lambda(t) \tag{4.100}$$

where $\lambda \in \mathbb{R}^q$ such that

$$\Delta\lambda(t) = f\big(x(t)\big),\ 0 \leq \lambda(t) \leq 1,\ \sum_{i=1}^{q} \lambda_i(t) = 1 \tag{4.101}$$

This means that $f\big(x(t)\big)$ can be represented as a convex combination of the extreme points of its bounding polytope. It should be noted that the vertices of the polytope are assumed to be constant vectors, which makes the matrix Δ constant. The vector λ is time varying. Equation (4.100) can be written in discrete form to get

$$x^{k+1} = \bar{A}x^k + \bar{B}u^k + \bar{D}\Delta\lambda^k \tag{4.102}$$

where $\Delta\lambda^k = f\left(x^k\right)$. If the state x^k is known, then each subsystem can be represented based on (4.102) as

$$x_i^{k+1} = \bar{A}x^k + \bar{B}u_i^k + \bar{D}\Delta_i \quad \forall i = 1, 2, \ldots, q \tag{4.103}$$

each of which is linear in the state. Since x^k is known, (4.101) can be solved for λ^k. By taking convex combinations of the control inputs from (4.103), the resulting control is $u^k = \begin{bmatrix} u_1^k, & u_2^k, & \ldots, & u_q^k \end{bmatrix} \lambda^k = U^k\lambda^k$. This gives

$$x^{k+1} = \bar{A}x^k + \bar{B}U^k\lambda^k + \bar{D}\Delta\lambda^k$$

What remains is to determine the control sequences u_i^k. A subproblem based on Problem P4.1 for each subsystem (4.103) is proposed as

$$\begin{aligned}
\min_{u_i^k} \quad & \frac{1}{2}\sum_{k=1}^{N-1}\left(\left(e_i^k\right)^\top Q e_i^k + \left(u_i^k\right)^\top R u_i^k\right) \\
& + \frac{1}{2}\left(e_i^N\right)^\top S e_i^N \\
\text{s.t.} \quad & \text{Equation (4.103)} \\
& e_i^k = x_i^k - x_f, \quad k = 1, 2, \ldots, N \\
& x_i^1 = x_0
\end{aligned} \tag{P4.2}$$

Now let

$$\bar{F} = \begin{bmatrix} \bar{B} & 0 & \cdots & 0 \\ \bar{A}\bar{B} & \bar{B} & & \vdots \\ \vdots & & \ddots & 0 \\ \bar{A}^{N-2}\bar{B} & \bar{A}^{N-3}\bar{B} & \cdots & \bar{B} \end{bmatrix} \tag{4.104}$$

$$\hat{A} = \begin{bmatrix} \bar{A} \\ \bar{A}^2 \\ \vdots \\ \bar{A}^{N-1} \end{bmatrix} \tag{4.105}$$

$$\hat{D} = \begin{bmatrix} \bar{D} \\ \bar{A}\bar{D} + \bar{D} \\ \vdots \\ \sum_{k=0}^{N-2} \bar{A}^k \bar{D} \end{bmatrix} \tag{4.106}$$

$$\bar{Q} = \text{diag}\,(Q, \ldots, Q, S) \tag{4.107}$$

$$\bar{R} = \text{diag}\,(R) \tag{4.108}$$

$$\bar{I} = \begin{bmatrix} I, & I, & \cdots, & I \end{bmatrix}^\top \tag{4.109}$$

Then a solution to Problem P4.2 is given by

$$\begin{bmatrix} u_i^1 \\ u_i^2 \\ \vdots \\ u_i^{N-1} \end{bmatrix} = \left(\bar{F}^\top \bar{Q} \bar{F} + \bar{R}\right)^{-1} \left(\bar{F}^\top \bar{Q} \left(\hat{A}x_0 + \hat{D}\Delta_i - \bar{I}x_f\right)\right) \tag{4.110}$$

With these solutions now available analytically, the following algorithm is proposed to give an approximate solution to Problem P4.0. The algorithm is shown in Algorithm 3. Step 5 requires the solution of a linear feasibility (programming) problem.

Algorithm 3 Algorithm for controller design using a combination of convex polytope vertices to represent nonlinearities

Input: x^k, x_f, t^k, t_f, T, A, B, $f(x)$, Δ
Output: u^k
Initialization:
1: Discretize the continuous system given by (4.97) to get (4.98)
Subproblems:
2: **for** $i = 1$ to q **do**
3: Use Equation (4.110) to get u_i^k
4: **end for**
Convex Combination:
5: Solve for λ^k such that $f(x^k) = \Delta\lambda^k$
6: Use λ^k to get $u^k = U^k\lambda^k$
7: **return** u^k

4.4.5 Examples

This section gives numerical examples which compare the controller introduced in Section 4.4.4 to a controller based on feedback linearization from Section 4.4.3. For feedback linearization based controller the following is introduced. After linearizing the system given in (4.97) using a feedback linearization, the control input v is computed as

$$\begin{bmatrix} v^1 \\ v^2 \\ \vdots \\ v^{N-1} \end{bmatrix} = \left(\bar{F}_z^\top \bar{Q} \bar{F}_z + \bar{R}\right)^{-1} \left(\bar{F}_z^\top \bar{Q} \left(\hat{A}_z z_0 - \bar{I} z_f\right)\right) \tag{4.111}$$

where $\bar{F}_z$ and $\hat{A}_z$ follow $\bar{F}$ and $\hat{A}$ introduced in (4.104) and (4.105) respectively with the exception that $\bar{A}$ is replaced with $\bar{A}_z$ and $\bar{B}$ with $\bar{b}_z$, which are the discrete versions of A_z and b_z. This gives a feedback linearization based controller which is comparable to the controller introduced in Section 4.4.4.

In both examples let $t_0 = 0$, $t_f = 1$, $T = .005$, $Q = 0$, $R = 1/N$, $S = NI$, and $x_f = [0, 0]^\top$.

Example 1: Consider the following nonlinear system

$$\dot{x}_1 = \sin(x_2)$$
$$\dot{x}_2 = -x_1^2 + u$$

on the domain

$$\mathcal{D} = \left\{ x \in \mathbb{R}^2 \middle| |x_1| \leq 1,\ 0 \leq x_2 \leq 2\pi \right\}$$

Note that the "linear part" of the system is not controllable. However, by rewriting the system such that

$$A = \begin{bmatrix} 0 & 0.1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \text{ and } f(x) = \begin{bmatrix} \sin(x_2) - 0.1x_2 \\ x_1^2 \end{bmatrix}$$

the (A, B) pair is controllable. Using Theorem 4.4 with

$$g(x) = \begin{bmatrix} \sin(x_2) \\ -x_1^2 \end{bmatrix}$$

it can be confirmed that the system is feedback linearizable with $\mathcal{D}_0 = \{x \in \mathcal{D} | \cos(x_2) \neq 0\}$. Note that the linearizable domain does not coincide with the domain of interest. Therefore, the control obtained by feedback linearization is not guaranteed to achieve the control objective.

The control inputs calculated using Algorithm 3 are shown in Figure 4.23 and corresponding state trajectories shown in Figure 4.24 as functions of time. In similar fashion, the control inputs as a function of time using the controller from Section 4.4.3 are shown in Figure 4.25 and the corresponding state trajectories are shown in Figure 4.26. It is easy to see that the feedback linearization controller fails when $x_2 = \pi/2$ with the control input becoming unbounded since $\gamma(x) = 0$.

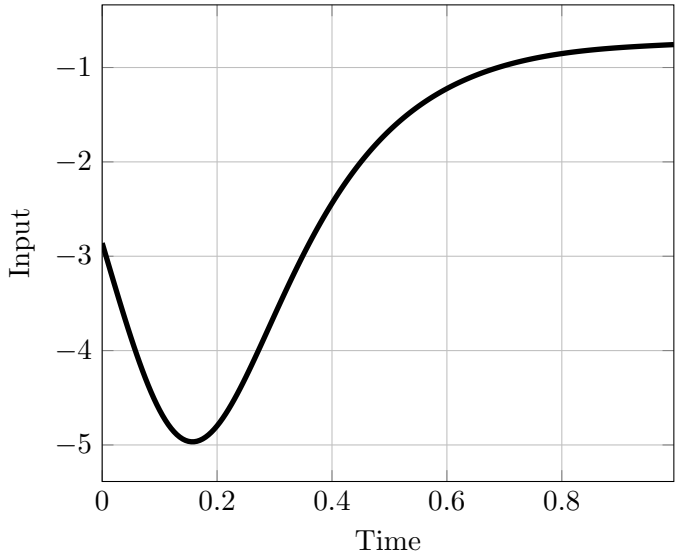


Fig. 4.23: Control input as a function of time for Example I using controller from Algorithm 3.

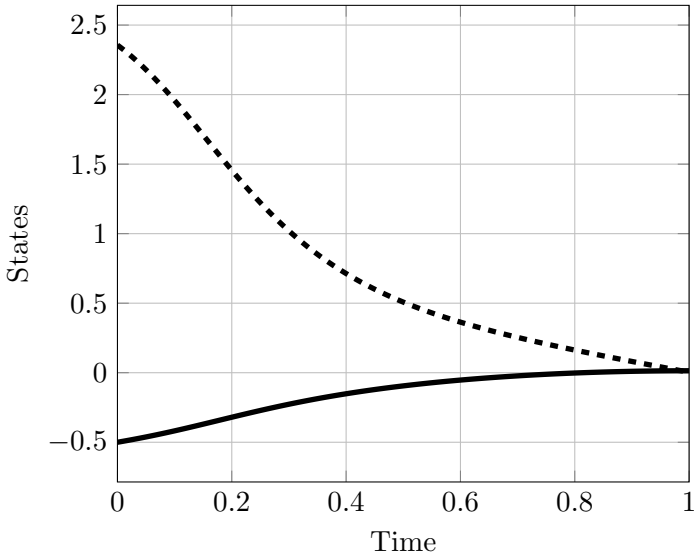


Fig. 4.24: State trajectories as a function of time for Example I using controller from Algorithm 3. Solid line represents x_1 and dashed line x_2.

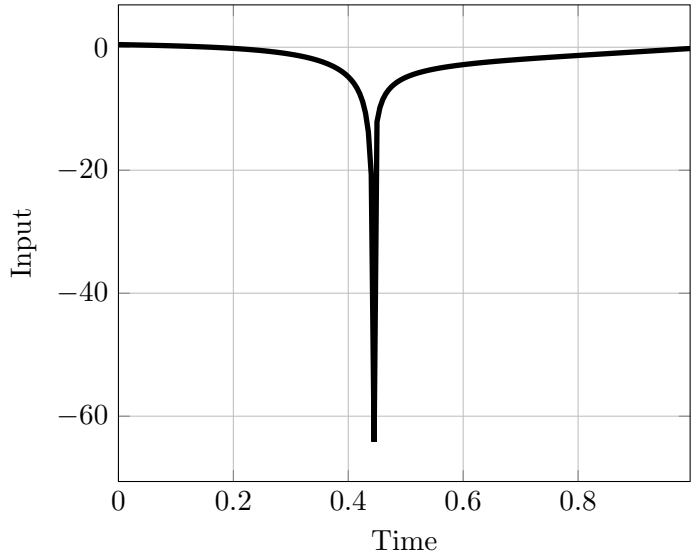


Fig. 4.25: Control input as a function of time for Example I using feedback linearization controller.

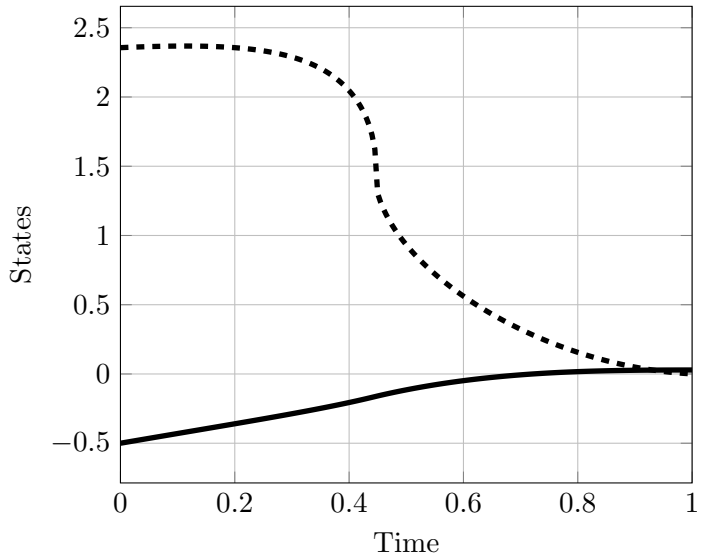


Fig. 4.26: State trajectories as a function of time for Example I using feedback linearization controller. Solid line represents x_1 and dashed line x_2.

Example 2: Consider the nonlinear system

$$\dot{x}_1 = x_1 + x_2^3$$

$$\dot{x}_2 = x_1 + x_2 + u$$

on the domain $\mathcal{D} = \mathbb{R}^2$. Note, again, that the "linear part" of the system is not controllable. However, by rewriting the system such that

$$A = \begin{bmatrix} 1 & 0.1 \\ 1 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \text{ and } f(x) = \begin{bmatrix} -0.1x_2 + x_2^3 \\ 0 \end{bmatrix}$$

the (A, B) pair is controllable. Using Theorem 4.4 with

$$g(x) = \begin{bmatrix} x_1 + x_2^3 \\ x_1 + x_2 \end{bmatrix}$$

the system is feedback linearizable on $\mathcal{D}_0 = \mathcal{D} \backslash \{x_2 = 0\}$. Note that the linearizable domain excludes the target state. Therefore, the control obtained by feedback linearization is not guaranteed to achieve the control objective.

With an initial state of $x_0 = [0, 0.1]^\top$, the control inputs calculated using Algorithm 3 are shown in Figure 4.27 and corresponding state trajectories are shown in Figure 4.28 as functions of time. In similar fashion, the control inputs as a function of time using a controller based on feedback linearization are shown in Figure 4.29 and the corresponding state trajectories are shown in Figure 4.30. Once again, it is easy to see that the feedback linearization controller fails when $x_2 = 0$ with the control input becoming unbounded which is caused by $\gamma(x) = 0$.

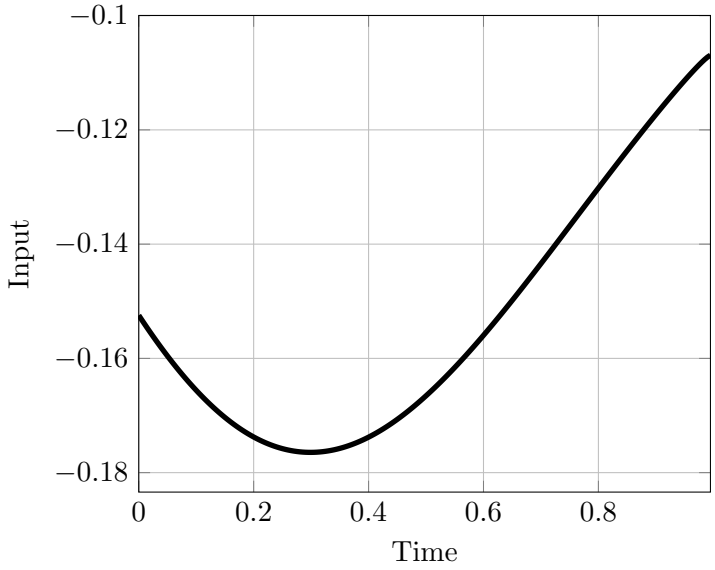


Fig. 4.27: Control input as a function of time for Example II using controller from Algorithm 3.

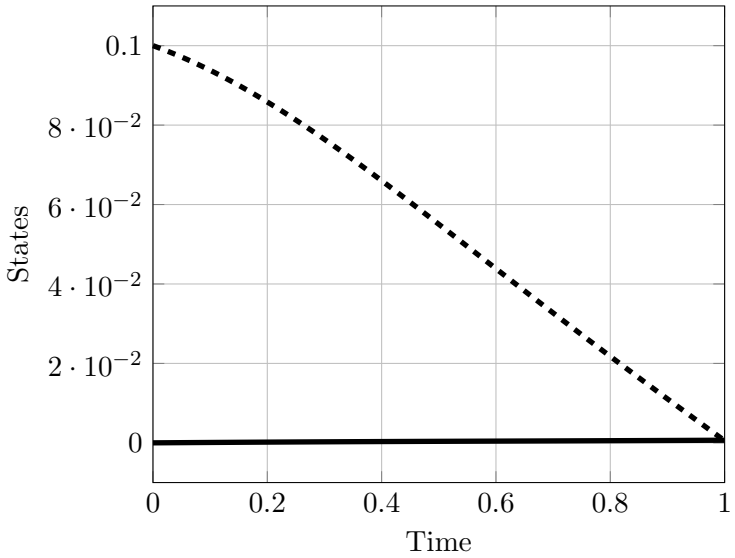


Fig. 4.28: State trajectories as a function of time for Example II using controller from Algorithm 3. Solid line represents x_1 and dashed line x_2.

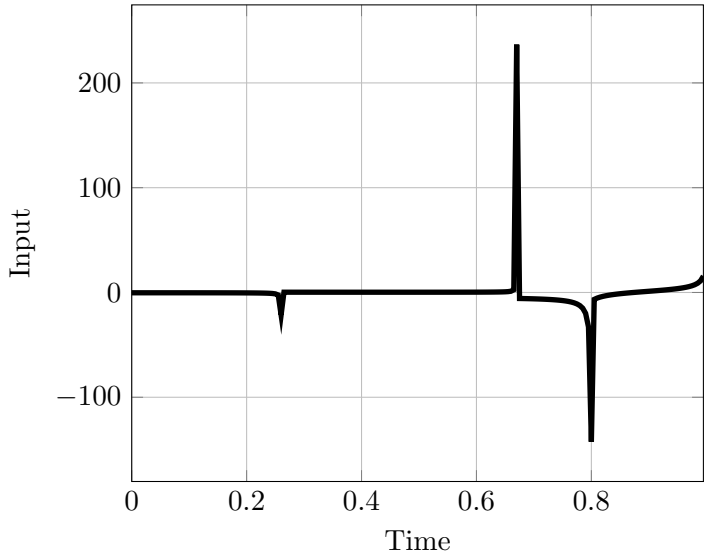


Fig. 4.29: Control input as a function of time for Example II using feedback linearization.

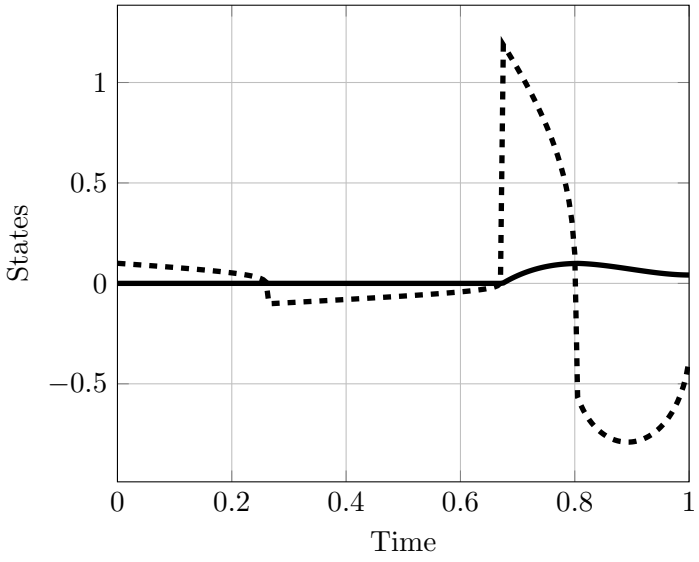


Fig. 4.30: State trajectories as a function of time for Example II using feedback linearization. Solid line represents x_1 and dashed line x_2.

These two examples show how feedback linearization controllers fail when the matrix G in Theorem 4.4 approaches singularity. When G becomes singular, it also means that $\gamma(z)$ approaches zero and the control input magnitude becomes unbounded. The controller

proposed in Algorithm 3 of Section 4.4.3 remains bounded throughout the time horizon and outperforms the feedback linearization based controller. These examples show that the proposed controller is an alternative for feedback linearization when the linearization domain does not coincide with the domain of interest.

4.5 Summary and Conclusions

This chapter consider a problem of trajectory design and control of systems with additive nonlinearities. First, systems with additive scalar nonlinearity were considered and a sufficient condition on when such system can be controlled by bounding linear systems was derived. This was done for both continuous-time and discrete-time systems with control constraints and convex state constraints. The scalar results were then generalized for discrete systems with additive nonlinearities and convex state and control constraints. A computational method for the multi-dimensional case similar to feedback linearization was also discussed in the last section, and compared against the classic feedback linearization technique through numerical examples.

CHAPTER 5
LINEAR PURSUIT-EVASION GAMES

5.1 Introduction

This chapter studies time-optimal pursuit-evasion games with linear dynamics and convex state and control constraints. Cases with a single pursuer and a single evader, multiple pursuers and a single evader, as well as a single pursuer and multiple evaders are considered. A geometrical approach is taken in solving these games by setting them up as reachable set problems as first introduced in [47]. By using modern, convex optimization techniques for reachable set construction of constrained systems, it is now possible to solve constrained, multi-player games not considered in [47].

Pursuit-evasion games commonly arise in aerospace engineering as well as in economics. The history of differential pursuit-evasion games begins with the seminal work by Isaacs [110]. His work also describes multiple interesting problems that can be set as differential pursuit-evasion games. An overview of the development of differential pursuit-evasion games is given in [111] including seminal work done in the 1970s as well as the current state of the art.

Aerial warfare is one of the typical problems that can be modeled as a pursuit-evasion game [112, 113]. Multiple assumptions are often made to simplify dynamical complexity. Some common assumptions restrict the motion to be planar, model the aircraft as a point-mass, linearize the dynamics, and require instantaneous control response [114]. Examples of different aerial warfare problems modeled as pursuit-evasion games consist of missile versus aircraft [44, 112, 114, 115], aircraft versus aircraft interception [116, 117], and fighter maneuvering [43].

Pursuit-evasion games have also been studied in the context of astrodynamics, particularly collision avoidance and interception [118,119]. Rather than use a nonlinear two-body

formulation, linearization and the use of the Clohessy-Wiltshire (CW) equations simplifies the problem so that minimax formulations lead to open-loop and closed-loop control strategies based on kriging [120]. Pursuit-evasion strategies have been used for tracking of space objects and selection of sensor management strategies [121]. A hybrid global-local technique has been developed for a two phase (long distance and short distance) game [122].

A pursuit-evasion game can be seen as an optimal control problem where the pursuer(s) and evader(s) have different objectives. The addition of multiple agents with opposite objectives generally makes the optimal control problem more difficult to solve than those of an individual pursuer's or evader's would be. It should be noted that sometimes the single agent optimal control problems can be complicated to solve as is. A technique utilizing calculus of variation may be used to to solve differential games. It requires setting an optimal control problem for all the players with the underlying necessary conditions and then finding a solution simultaneously for all the players. Linear-quadratic as well as time-optimal scalar games were solved using this methodology in [40]. A case of one of the players not playing the game optimally based on the variational approach solution is also included in [40] as is a short discussion on games with stochastic behavior.

Because of their complexity, it is often difficult to solve differential games using the variational theory. This is particularly true when the governing differential equations are nonlinear and there are pointwise constraints on the states and controls. The pointwise constraints introduce switches as trajectories enter and leave control boundaries, which may violate smoothness assumptions required for a Newton method. Consequently, direct methods [45, 46] and semi-direct methods have been introduced to leverage finite-dimensional optimization packages. These methods involve the use of genetic algorithms and neural networks [41–44].

In contrast to the optimization-based approaches, Mizukami [47] has shown that a two-player time-optimal pursuit-evasion game terminates when the evader's reachable set is contained in the pursuer's reachable set – motivating a geometric approach to solving the game based on reachable set computation. More precisely, the termination point in state

space lies on the boundary of the players' reachable sets. For problems with linear dynamics and integral-constrained (rather than pointwise-constrained) controls, analytical solutions can be derived for the termination time and optimal trajectories. Motivated by this work, reachable set analyses have been used in dynamic flowfields [48], coordinate control [49], missile/sensor trade studies [51], and other game scenarios [50, 52].

In general, it is difficult to find analytical solutions for reachable sets, and consequently, numerical methods are utilized to approximate them. Multiple numerical algorithms to calculate a reachable set for control affine systems are provided in [123]. An algorithm for the computation of reachable sets for linear systems with bounded inputs, which approaches the problem by finding inner and outer approximations, is introduced in [124]. Optimal control has also been used to calculate reachable sets [125–128]. A polytopic approximation of the actual reachable set is computed in them. In [127], a single semidefinite programming (SDP) problem and multiple sequential second-order cone programming (SOCP) problems are solved to find an inner approximation of the reachable set for linear dynamics and convex state and control constraints. An outer approximation is added in [126] as a heuristic to determine the precision of the approximation. Non-convex control constraints are considered in [128]. The approach from [127] is used as a basis for solving multi-player games in this chapter.

The primary contribution of this chapter is the construction of reachable set approximations to solve constrained pursuit-evasion games. The inclusion of pointwise, convex constraints on the state and control make the variational approach difficult, while such constraints add minimal additional complexity to the geometric approach. Inclusion of multiple pursuers and evaders adds complexity to the problem as well, especially if variational techniques were to be used. It is shown that with reachable set methods, solving pursuit-evasion problems with multiple pursuers and evaders stays tractable. It is assumed that the pursuers and evaders have perfect information about the other players and everyone behaves optimally. Pursuit-evasion problems with 1) a single pursuer and a single evader, 2) multiple pursuers and a single evader, and 3) a single pursuer and multiple evaders are

considered. Numerical aerospace-related examples of all the cases are given to demonstrate the capability of the method. In these examples, the agents are assumed to be in close proximity to each other and in low earth orbit.

The remainder of this chapter is structured as follows. Section 5.2 describes a numerical method based in convex optimization for approximating reachable sets for both deterministic and stochastic systems. The algorithm is based on the one introduced in [127]. The constrained pursuit-evasion game is introduced in Section 5.3 and an explanation on how reachable set theory can be used to solve it for varying number of pursuers and evaders is explained. Section 5.5 applies the method to aerospace related single-pursuer and single-evader problems whose dynamics and constraints are explained in Section 5.4. Examples with multiple pursuers or evaders are shown in Section 5.6. Results of this chapter are summarized and conclusions drawn in Section 5.7.

5.2 Reachable Sets

An introduction to reachable sets is given in this section. Reachable sets for linear time-varying (LTV) dynamics with second-order cone control constraints and linear state constraints are considered. This system of interest is given as

$$\dot{x} = A(t)x + B(t)u \tag{5.1}$$

$$y = C(t)x \tag{5.2}$$

$$x \in \mathcal{X}(t) = \{x \in \mathbb{R}^n : D(t)x \leq b(t)\} \tag{5.3}$$

$$u \in \mathcal{U}(t) \tag{5.4}$$

$$x(t_0) = x_0 \tag{5.5}$$

The state vector is $x \in \mathbb{R}^n$, the control vector is $u \in \mathbb{R}^m$, the output vector is $y \in \mathbb{R}^p$, the system matrix is $A(t) : \mathbb{R}^n \to \mathbb{R}^n$, the control influence matrix is $B(t) : \mathbb{R}^m \to \mathbb{R}^n$, and the output matrix is $C(t) : \mathbb{R}^n \to \mathbb{R}^p$. The quantities $D(t) \in \mathbb{R}^{q \times n}$ and $b(t) \in \mathbb{R}^q$ define linear

state inequality constraints for the set $\mathcal{X}(t)$. The second-order cone control constraint is $\mathcal{U}(t)$. The initial state vector is x_0 at initial time t_0.

Definition 5.1. *For the system in* (5.1)-(5.5)*, the reachable set* $\mathcal{R}(t)$ *is the set of all outputs the system can reach at time* t *from the initial state* x_0 *using feasible controls, i.e.,*

$$\mathcal{R}(t) = \{y \in \mathbb{R}^p \ : \ y = C(t)\Phi(t,t_0)x_0 + C(t)\int_{t_0}^{t} \Phi(t,\tau)B(\tau)u(\tau)\, d\tau \text{ for all feasible } u\} \tag{5.6}$$

where $\Phi(t_2, t_1)$ *is the state-transition matrix.*

Remark 5.1. *Because the dynamics of the system are linear and the state and control constraints are convex, the reachable set is convex [129].*

5.2.1 Algorithm for Reachable Set Calculation

An algorithm to calculate an inner polytopic approximation of the system described in (5.1)-(5.5) is given next. The algorithm first solves a SDP optimization problem to compute the largest possible size simplex that fits inside the reachable set. The algorithm then solves a sequence of SOCP problems to compute a better inner approximation of the reachable set based on the initial simplex. A more detailed explained of the algorithm can be found in [127].

First, consider a discretized version of the continuous system given in (5.1)-(5.5). The discrete-time system is

$$x_{k+1} = \bar{A}_k x_k + \bar{B}_k u_k, \ k = 0, ..., N-1 \tag{5.7}$$

$$y_k = C_k x_k, \ k = 0, ..., N \tag{5.8}$$

$$x_k \in \mathcal{X}_k = \{x \in \mathbb{R}^n : D_k x \leq b_k\} \tag{5.9}$$

$$u_k \in \mathcal{U}_k \tag{5.10}$$

$$x_{k=0} = x_0 \tag{5.11}$$

where N represents the number of time intervals used in discretization. The barred quantities are discretized matrices based on their continuous counterparts and discretization time step with the discretization performed as explained in Section 2.3.1. Discretization allows the use of numerical optimization. The reachable set definition given in Definition 5.1 for continuous system is analogous to that for discrete system by replacing t with t_k.

Initial Simplex

The computation of the initial simplex is explained first. Computing the initial simplex requires solving a single SDP problem. Let $\mathcal{Z}$ be the set containing the $p+1$ vertices of an initial simplex in $\mathbb{R}^p$. Furthermore, define a matrix Q as

$$Q = \begin{bmatrix} (z_2 - z_1) & \dots & (z_{p+1} - z_1) \end{bmatrix} : z_i \in \mathcal{Z} \tag{5.12}$$

The volume of the simplex may then be calculated as

$$v = \frac{1}{p!} |\det Q| \tag{5.13}$$

With an expression for a simplex volume, maximizing (5.13) gives us an initial simplex with maximum volume. This also leads to the vertices of the simplex being on the boundary of the system's reachable set.

$$z_i \in \partial\mathcal{R} \subset \mathcal{R} \tag{5.14}$$

This maximum volume simplex is used as the initial simplex. The volume maximization problem can be written as a SDP.

$$\begin{aligned} \min_{u_i} \quad & -\operatorname{logdet}(Q) \\ s.t. \quad & Q = Q^T = \begin{bmatrix} (z_2 - z_1) & \dots & (z_{p+1} - z_1) \end{bmatrix} \succeq 0 \\ & z_i = y_i(t_f), i = 1, ..., p+1 \\ & \text{Eqs. (5.7)-(5.11)} \end{aligned} \tag{5.15}$$

The log-determinant is denoted logdet, which acts on square symmetric matrices. The initial simplex requires calculation of $p+1$ trajectories emanating from the initial point, x_0, to $p+1$ final states. The initial simplex is constructed from the $p+1$ final state vectors which form the $z_1, \ldots, z_{p+1}$ vertices of the initial simplex.

Growing Simplices

Following construction of the initial simplex, improved approximations are achieved by computing additional simplices out of the open faces of the current polytopic approximation. Where the computation of the initial simplex required a solution to a SDP problem, the computation of additional simplices is achieved by solving a SOCP for each additional vertex in the polytopic approximation. This is possible because maximizing the volume of an additional simplex is analogous to maximizing the length of a vector that connects the open face to a point on the boundary of the reachable set and is orthogonal to the face. The point on the boundary of the reachable set is then added as a new vertex to the current polytopic approximation of the reachable set.

$$\begin{aligned} \min_{\alpha,\lambda,u} \quad & -\alpha \\ \text{s.t.} \quad & \nu = Z_i\lambda, \quad 1^\top\lambda = 1, \quad \lambda \geq 0 \\ & \nu + \alpha h_i = z \\ & z = y(t_f) \\ & \text{Eqs. (5.7)-(5.11)} \end{aligned} \tag{5.16}$$

The quantity to be maximized is the scaling factor α. The point on the open face ν is such that the minimum distance between it and the reachable set boundary is maximized. The matrix Z_i contains the p vertices corresponding to the open face i. The vector $\lambda \in \mathbb{R}^p$ identifies the point ν from the vertices in Z_i. A vector normal to the i^{th} open face is given by h_i. A point on the boundary of the reachable set is z. Figure 5.1 gives a graphical representation of these variables with ν being represented as a vector from a vertex on an open face to the actual point ν.

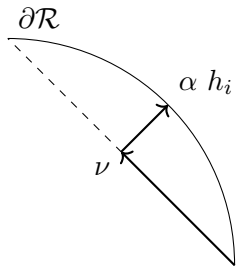


Fig. 5.1: Illustration of variables used for polytopic approximation in (5.16).

Overall, the algorithm requires a solution to a single SDP defined in (5.15) to get an initial approximation of the reachable set. A tighter approximation is then reached by solving a sequence of SOCPs given in (5.16) with each SOCP adding a new vertex to the current polytopic approximation. The first three reachable set approximations are illustrated in Figure 5.2 along with the actual reachable set. Further explanation and illustration of the approximation process may be found in [127].

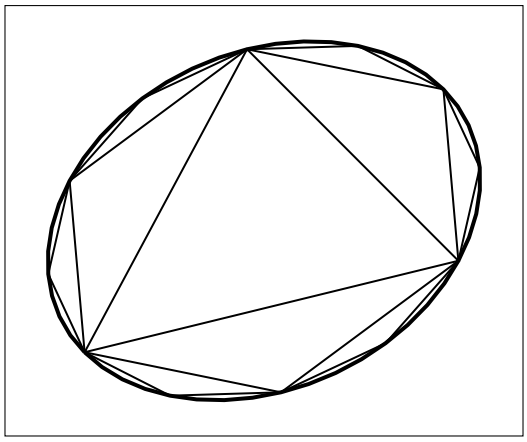


Fig. 5.2: The initial simplex is the innermost triangle. The next two generations share vertices with the triangle and give better approximations of the reachable set, which appears as the ellipse with a thicker line.

5.2.2 Stochastic System

In this section, the algorithm is extended for dynamical systems with stochastic uncertainty. Consider the following discrete, stochastic, LTV control system with state and

control constraints.

$$x_{k+1} = \bar{A}_k x_k + \bar{B}_k u_k + W_k w_k \tag{5.17}$$

$$y_k = \bar{C}_k x_k \tag{5.18}$$

$$\delta \geq \text{Prob}\left(D_k x_k > b_k\right) \tag{5.19}$$

$$||u_k|| \leq \rho_k \tag{5.20}$$

$$x_{k=0} = x_0 \tag{5.21}$$

where w is an unknown stochastic disturbance and δ is an upper bound on the probability of not satisfying the constraint. The uncertainties vary as follows:

$$w_k \sim \mathcal{N}(0, Q_k) \tag{5.22}$$

$$x_0 \sim \mathcal{N}(\bar{x}_0, P_0) \tag{5.23}$$

$$P_{k+1} = \bar{A}_k P_k \bar{A}_k^T + W_k Q_k W_k^T \tag{5.24}$$

where Q_k, P_0 and P_k are positive definite covariance matrices and $\bar{x}_0$ is the expected value of the initial state. The stochastic problem described in (5.17)-(5.21) with the uncertainties stated in (5.22)-(5.24) can be written in a deterministic form by taking a conservative approximation of (5.19). This approximation is given as [130]:

$$\sum_{k=0}^{N} \big(\text{Prob}\left(D_k x_k > b_k\right)\big) \leq \delta \tag{5.25}$$

where the probability inequality is equal to

$$\text{Prob}\left(D_{i,k} x_k > b_{i,k}\right) = \frac{1}{2} - \frac{1}{2}\text{erf}\left(\frac{b_{i,k} - D_{i,k}\bar{x}_k}{\sqrt{2\sigma_{i,k}}}\right) \tag{5.26}$$

with $\sigma_{i,k} = D_{i,k} P_k D_{i,k}^T$, $\bar{x}_k$ is an expected value of x_k, $i = 1, ..., q$ specifies the row of vector or matrix and erf$(\cdot)$ is the error function. Using (5.25) and (5.26), a conservative but

deterministic approximation of the stochastic problem given in (5.17)-(5.21) is below.

$$x_{k+1} = \bar{A}_k x_k + \bar{B}_k u_k + W_k w_k \tag{5.27}$$

$$y_k = \bar{C}_k x_k \tag{5.28}$$

$$D_{i,k}\bar{x}_k \leq b_{i,k} - \sqrt{2\sigma_{i,k}}\text{erf}^{-1}(1 - 2\Delta_{i,k}) \tag{5.29}$$

$$\delta \geq \sum_{k=0}^{N} \Delta_k \tag{5.30}$$

$$||u_k|| \leq \rho_k \tag{5.31}$$

$$x(k=0) = x_0 \tag{5.32}$$

where Δ_k are risk allocation variables that can be included as solution variables or preset, and $\text{erf}^{-1}(\cdot)$ is the inverse error function. The inverse error function cannot be included within an SOCP, but a conservative, quadratic approximation can. The quadratic approximation is given as

$$\text{erf}^{-1}(\xi) = a + b\xi + c\xi^2,\ \xi \in [1 - \delta_i, 1 - \delta_i/\alpha] \tag{5.33}$$

where $i = 1, ..., q$, $\alpha >> 1$ and a, b, and c are coefficients that can be solved using least squares method. Since we now have a deterministic SOCP, an approximation for its reachable set can be calculated using the algorithm provided in the previous section by switching the constraint equations from (5.7)-(5.11) to (5.27)-(5.32) and using (5.33) to evaluate the inverse error function.

5.3 Game Theory

This section introduces the pursuit-evasion game of interest. Three pursuit-evasion games considered in this chapter are games with 1) a single pursuer and a single evader, 2) multiple pursuers and a single evader, and 3) a single pursuer and multiple evaders. The pursuers' objective is to minimize the capture time of all the evaders. The evaders' objective is to maximize the time it takes for the pursuers to capture all the evaders. It is assumed

that a solution exists, i.e. capture occurs, such that the game considered here is a game of degree rather than a game of kind [110].

5.3.1 Single Pursuer and Single Evader

First, consider the simplest case of the game with a single pursuer and a single evader. The game may be rewritten as two time-optimal control problems. The pursuer's problem is:

$$\begin{aligned}
\min_{u_P(\cdot)} \quad & t^* \\
\text{s.t.} \quad & \dot{x}_P = A_P(t)x_P + B_P(t)u_P \\
& y_P = C_P(t)x_P \\
& x_P \in \mathcal{X}_P(t) = \{x \in \mathbb{R}^n : D_P(t)x \leq b_P(t)\} \\
& u_P \in \mathcal{U}_P(t) \\
& x_P(t_0) = x_{0,P} \\
& y_P(t^*) = y_E(t^*)
\end{aligned} \tag{P5.1}$$

The evader's problem is:

$$\begin{aligned}
\max_{u_E(\cdot)} \quad & t^* \\
\text{s.t.} \quad & \dot{x}_E = A_E(t)x_E + B_E(t)u_E \\
& y_E = C_E(t)x_E \\
& x_E \in \mathcal{X}_E(t) = \{x \in \mathbb{R}^n : D_E(t)x \leq b_E(t)\} \\
& u_E \in \mathcal{U}_E(t) \\
& x_E(t_0) = x_{0,E} \\
& y_E(t^*) = y_P(t^*)
\end{aligned} \tag{P5.2}$$

The subscript P refers to the pursuer, the subscript E refers to the evader, and t^* is the termination time of the game. It is assumed that both pursuer and evader have perfect information about the other.

It was shown by Mizukami [47] that such time-optimal games can be recast as reachable set inclusion problems.

Theorem 5.1 ([47]). *If a solution to the game exists, then the game terminates in the least time such that*

$$\mathcal{R}_E(t) \subseteq \mathcal{R}_P(t) \tag{5.34}$$

The question of existence is not explored in this dissertation. It is assumed that a solution exists, and the theorem is used to find the solution by comparing the players' reachable sets. The termination time is the first time at which the evader's reachable set is contained in the pursuer's reachable set. The game is therefore reduced to a one-dimensional search for t^*. In state space, the game terminates at a point on the boundary of both players' reachable sets, i.e., $y^* \in \partial R_E(t^*) \cap \partial R_P(t^*)$, where y^* is the capture point.

The simplest situation occurs when $C(t)\Phi(t, t_0)x_0$ is zero (so that the reachable sets do not translate) and the pursuer's control set is larger than the evader's (so that it enlarges at a faster rate). Prior to the termination time, the evader's reachable set is not contained in the pursuer's. After the termination time, the evader's reachable set is in the interior of the pursuer's. At the termination time, the evader's set is contained in the pursuer's set and they share a boundary point. These relationships are demonstrated in Figure 5.3.

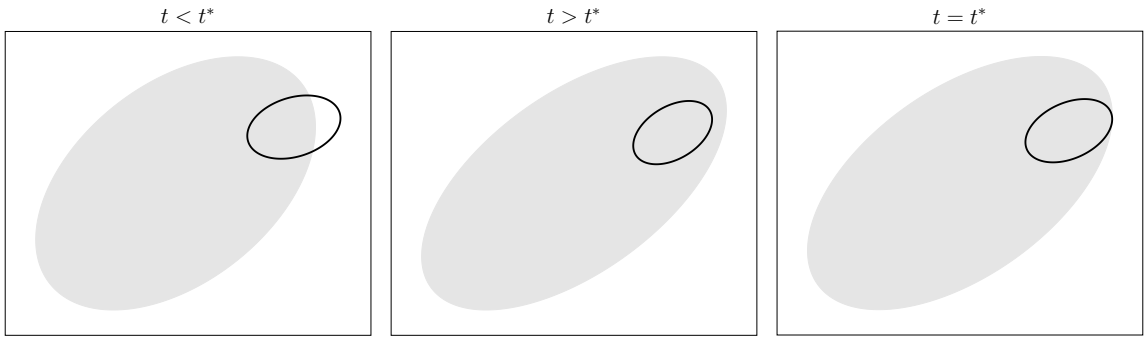


Fig. 5.3: Pursuer's (shaded region) and evader's (solid line) reachable sets before capture ($t < t^*$), after capture ($t > t^*$), and at capture ($t = t^*$).

5.3.2 Multiple Pursuers and Single Evader

When there are multiple pursuers and a single evader, the game terminates when the evader's reachable set is contained in the union of the pursuers' reachable sets [48], i.e.,

$$\mathcal{R}_E(t) \subseteq \left\{ \bigcup_{i=1}^{N_P} \mathcal{R}_{P_i}(t) \right\} \tag{5.35}$$

where N_P is the number of pursuers. If the game has only one pursuer, (5.35) reduces to (5.34) as expected. Some of the possible capture scenarios are shown in Figures 5.4 and 5.5. In Figure 5.4, the capture happens at the boundary of the two pursuers' and evader's sets. In Figure 5.5, the capture happens at the boundary of the four pursuers' sets but is in the interior of the evader's reachable set. In both cases, however, the capture happens at the first time instance when the evader's reachable set is contained in the union of the pursuers' reachable sets.

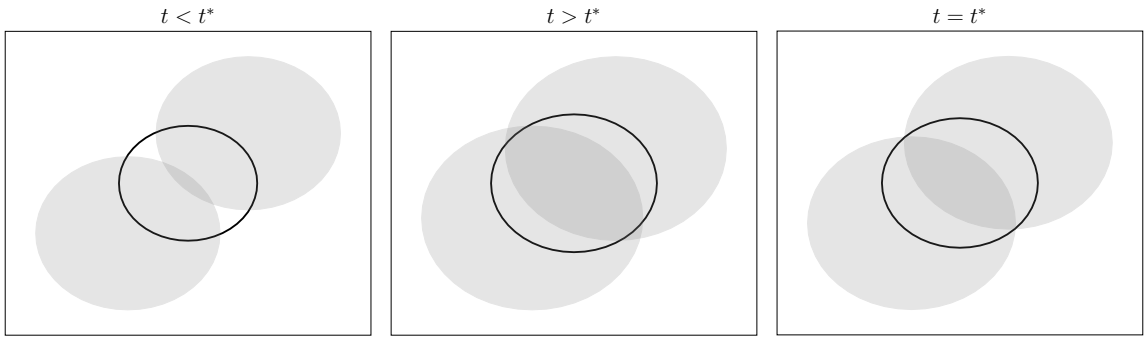


Fig. 5.4: Pursuers' (shaded region) and evader's (solid line) reachable sets before capture, after capture and at capture.

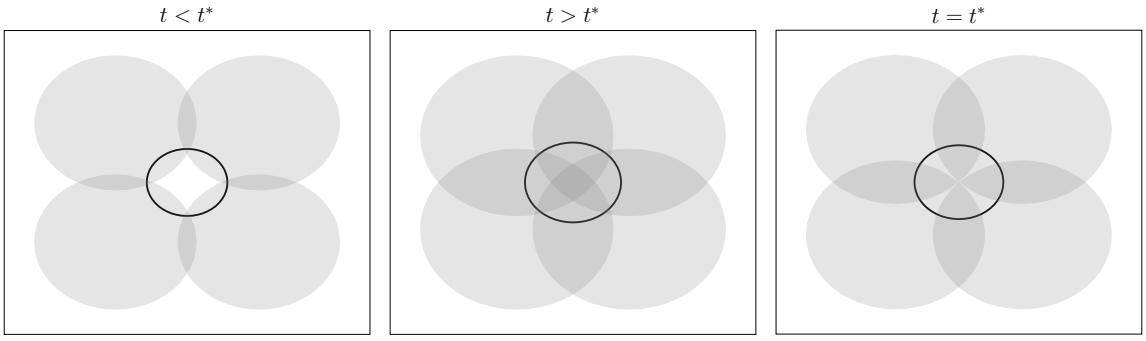


Fig. 5.5: Pursuers' (shaded region) and evader's (solid line) reachable sets before capture, after capture and at capture.

A line search is done to find the capture time. An algorithm to perform this line search is shown in Algorithm 4. The algorithm uses a method from Section 5.2.1 to generate inner approximations of the pursuers' and evader's reachable sets and checks if (5.35) is satisfied or not. The algorithm uses a bisection method to find the capture time. The algorithm uses notation z_X for the vertices of the agent X's reachable set approximation where $\mathcal{R}_X$ would be agent X's reachable set. N_P is the number of pursuer's in the game. Subscript $\cup P$ means the union over all pursuers.

Algorithm 4 Line Search for Capture Time in Multiple Pursuer and Single Evader Game

Input: Dynamics for evader and pursuers, initial states, ϵ, t_{min}, t_{max}

Output: t^*

Set $t_0 = (t_{min} + t_{max})/2$ and $i = 0$.
while $\|t_i - t_{i-1}\| < \epsilon$ **do**
  $i = i + 1$
  Calculate reachable sets for the evader and each pursuer.
  $z_{\cup P} = \cup_{j=1}^{N_P} z_{P,j} \notin \mathcal{R}_{P,k}$, $k \in \{1, \ldots, N_P\} \setminus \{j\}$
  if All $z_E \in \mathcal{R}_{\cup P}$ or any $z_{\cup P} \in \mathcal{R}_E$ **then**
    $t_i = (t_{min} + t_{i-1})/2$, $t_{max} = t_{i-1}$.
  else
    $t_i = (t_{max} + t_{i-1})/2$, $t_{min} = t_{i-1}$
  end if
end while
$t^* = t_i$

5.3.3 Single Pursuer and Multiple Evaders

Lastly, a pursuit-evasion game with a single pursuer and multiple evaders is considered. This game is split into multiple single-pursuer and single-evader games where the pursuer's reachable set is reset at the capture point when a capture of one of the evaders happens. Figure 5.6 demonstrates the evolution of pursuer's and evaders' reachable sets where the pursuer's reachable set is reset after capturing an evader.

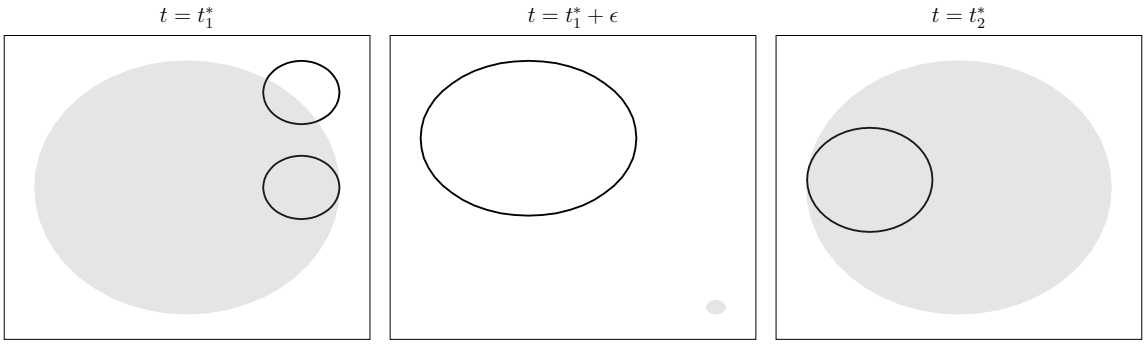


Fig. 5.6: Pursuer's (shaded region) and evaders' (solid line) reachable sets at capture of the first evader, shortly after the capture of the first evader and at capture of the second evader. Notice that shortly after capture of the first evader, the pursuer's reachable set is reset at the capture point whereas the second evader's set continues to grow.

To consider all the possible strategies for the pursuer to capture the evaders, it is in general required to consider $N_E!$ possibilities for capturing sequences where N_E is the number of the evaders. This means that in a case of three evaders, the pursuer could capture the evaders in six different orders, 1,2,3; 1,3,2; 2,1,3; 2,3,1; 3,1,2 or 3,2,1. Let's consider a case where catching four evaders in an order 1,2,3,4 takes $t_{1,2,3,4}$ time units. If catching only evaders 4 and 3 in this order takes $t_{4,3}$ time units such that $t_{1,2,3,4} < t_{4,3}$, it is not optimal for the pursuer to capture the evaders starting with evaders 4 and 3 as catching these two evaders in this order takes more time than catching all the evaders in order 1,2,3,4. This logic leads to a pruning strategy that may allow finding the termination time for the game without considering all the capture strategies completely.

Figure 5.7 shows the pursuer's possibilities for capture order in a three evader game. t with subscripts represents the capture time of the evaders in the order of numeric values in the subscript. t^* is the termination time of the game associated with the shallowest branch, i.e., the branch offering the least time compared to all other branches. In this example, the selected order for capturing the evaders is 2,1,3. The nodes beyond this capture time are not considered.

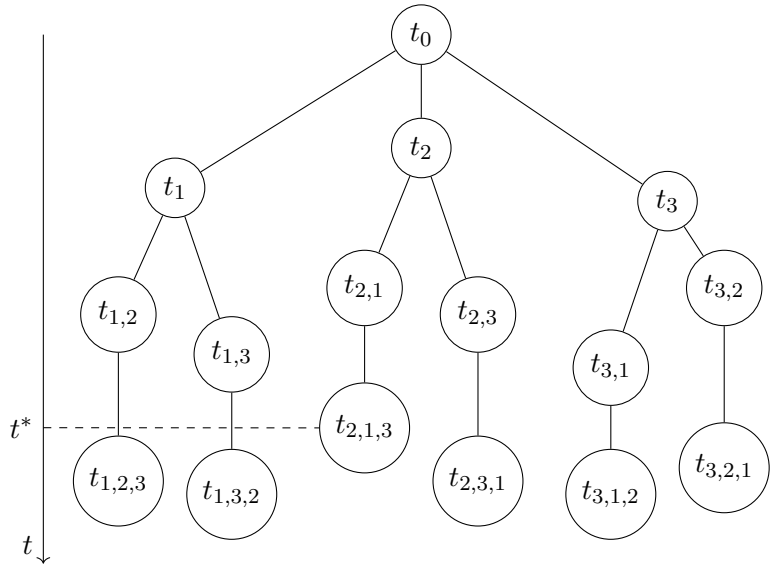


Fig. 5.7: Tree graph showing the different possibilities (branches) on the order pursuer could capture the evaders. t^* is the termination time for the pursuer to capture all the evaders.

An algorithm to find the termination time t^* for the multiple evader and a single pursuer game is provided in 5. The algorithm begins by finding the termination time of the game with a greedy approach when choosing the next evader being captured. The other capturing sequences are then compared to this. If the capture time of an evader is larger than the current estimate for the termination time of the game $t^*_{current}$, that capturing sequence is excluded from further analysis, and the branch is pruned. On the other hand, if the game termination time using that sequence is less than the current estimate for the game termination time, the current estimate for the game termination time is updated to that.

Algorithm 5 Termination Time in Single Pursuer and Multiple Evader Game

Input: Dynamics for evaders and pursuer, initial states

Output: t^*

Set $t^*_{current} = \infty$
for $i = 1$ **to** $N_E!$ **do**
  while Free evaders left **do**
    Find capture time, t of the next evader such that this capture order is not yet considered.
    Add the captured evader in the capture sequence.
    if $t \geq t^*_{current}$ **then**
      Set this capture sequence considered.
      break while
    end if
    Reset pursuer's reachable set at capture point.
  end while
  if $t < t^*_{current}$ **then**
    $t^*_{current} = t$
  end if
end for
$t^* = t^*_{current}$

5.4 Dynamics

The dynamics used for numerical examples in Sections 5.5 and 5.6 are explained in this section. A planar constellation of satellites near a circular orbit is considered in the

numerical examples. The dynamics can be modeled as linear by the use of the Clohessy-Wiltshire (CW) equations [106]

$$\begin{aligned} \dot{x} &= Ax + Bu = \begin{bmatrix} 0 & I \\ M_1 & M_2 \end{bmatrix} x + \begin{bmatrix} 0 \\ I \end{bmatrix} u \\ y &= Cx = \begin{bmatrix} I & 0 \end{bmatrix} x \end{aligned} \tag{5.36}$$

where $x \in \mathbb{R}^4$ is a state vector with the first two elements corresponding to a satellite's relative position in a local vertical local horizontal (LVLH) frame and the last two to a satellite's relative velocity in the LVLH frame. The external control acceleration is $u \in \mathbb{R}^2$. The matrices M_1 and M_2 are given by

$$M_1 = \begin{bmatrix} 3\omega^2 & 0 \\ 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 2\omega \\ -2\omega & 0 \end{bmatrix} \tag{5.37}$$

where ω is the mean motion of the circular reference orbit and is set to 4 rad/hr for the remainder of this section. This corresponds to a low earth orbit. In all the examples, the output vector y is to be the relative position of the player, which means that the output matrix is $C = \begin{bmatrix} I & 0 \end{bmatrix}$ with $I, 0 \in \mathbb{R}^2$. The relative position is chosen as the output because capture occurs in position space. The first element of the y vector corresponds to the player's radial position (altitude) in the LVLH frame and the second element to the player's along-track position in LVLH frame.

5.5 Examples with Single Pursuer and Single Evader

In this section, examples of several time-optimal pursuit-evasion games with a single pursuer and a single evader are provided, and the techniques discussed in Section 5.3 are applied to find solutions to these examples. All the agents evolve according to CW dynamics as explained in Section 5.4. Each agent's control set is of the form

$$\mathcal{U}(t) = \{u \in \mathbb{R}^2 \ : \ ||u|| \leq \rho\} \tag{5.38}$$

with ρ a prescribed upper bound on the control magnitude. This is a second-order cone constraint. Additional constraints consistent with those in Equations (5.1)-(5.5) are also present and described in the following subsections.

5.5.1 Problem without State Constraints

Consider a single-pursuer and single-evader pursuit-evasion problem with the players' dynamics given in (5.36). The initial state of the pursuer is $x_{0,P} = [0, 1, 0, 0]^\top$ and of the evader is $x_{0,E} = [0, 0, 0, 0]^\top$. The control of the pursuer is bounded by $\rho_P = 1$ and the control of the evader by $\rho_E = 0.5$. All the units of distance in these problems are km and the units of time hr.

Figure 5.8 shows the reachable sets of the pursuer and evader when the capture happens. The trajectories that the pursuer and evader take to reach the capture point are also shown in the figure. The capture happens at $t^* = 1.32$ hr. It is evident from the figure that the capture happens at the boundary of both pursuer's and evader's reachable sets.

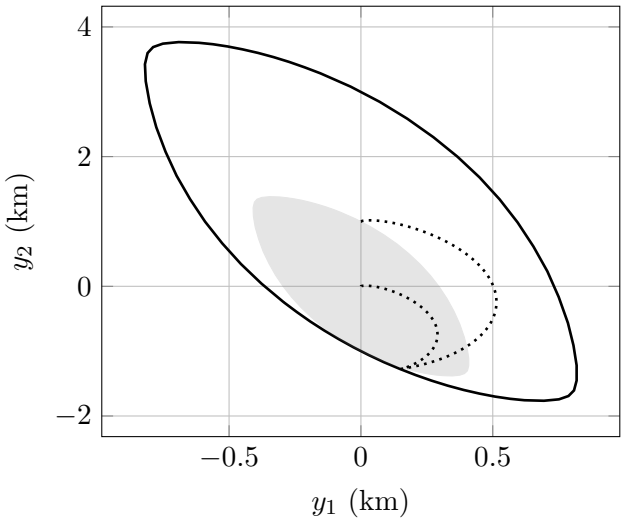


Fig. 5.8: Pursuer's (shaded) and evader's (solid) reachable sets at capture time with the trajectories of the pursuer and evader from their initial outputs to the capture point shown with dotted lines.

5.5.2 Problem with State Constraints

Now consider the same problem setup with state constraints. Data for the state constraints in (5.3) for both pursuer and evader is given as

$$D = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{5.39}$$

Because of these constraints, the previous solution is now infeasible. Figure 5.9 shows the reachable sets of the pursuer and evader when the capture happens. The trajectories that the pursuer and evader take to reach the capture point are also shown in the figure. The capture happens at $t^* = 1.42$ hr. Once again, the capture happens at the boundary of both pursuer's and evader's reachable sets. The black regions demonstrate the state constraints.

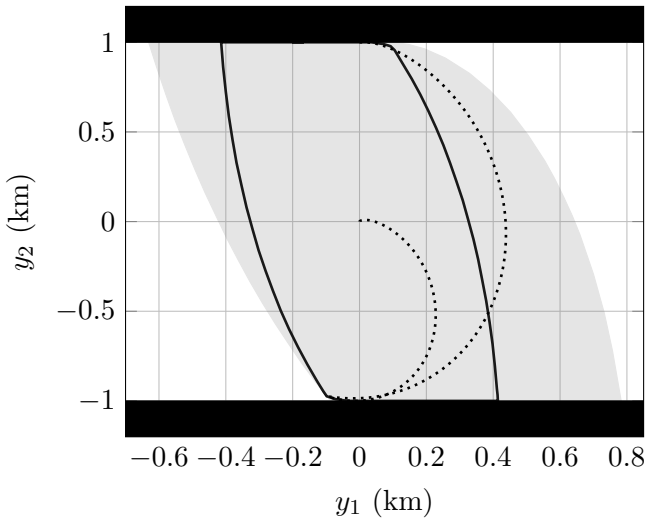


Fig. 5.9: Pursuer's (shaded) and evader's (solid) reachable sets at capture time with the trajectories of the pursuer and evader from their initial outputs to the capture point shown with dotted lines. The black areas represent the linear state constraints or keep-out zones.

5.5.3 Stochastic Problem

The problem in Section 5.5.2 is now changed by making the state constraint probabilistic, introducing a stochastic input into the dynamics, and considering an uncertain initial

point. The formulation follows that given in (5.17)-(5.24) where

$$\delta_P = \delta_E = .05 \tag{5.40}$$

$$W_P = W_E = I_{4\times 4} \tag{5.41}$$

$$Q_P = Q_E = 10^{-6} I_{4\times 4} \tag{5.42}$$

$$P_{P,0} = P_{E,0} = 10^{-9} I_{4\times 4} \tag{5.43}$$

$$N = 40 \tag{5.44}$$

and all units are consistent with the data specified in the previous examples. The initial condition of the pursuer is changed to $x_{0,P} = [0, 0.75, 0, 0]^\top$ due to the stochastic nature of the problem. By conducting a one-dimensional search, the optimal capture time is found to be $t^* = 1.2$ hr. The reachable sets as well as the trajectories of both the pursuer and evader are shown in Figure 5.10. It is evident from the plot that both, pursuer and evader stay away from the keep-out zones. Due to the uncertainty in both of their systems, the agents do not go on the boundary of the keep-out zones like they did in the deterministic example given in Section 5.5.2.

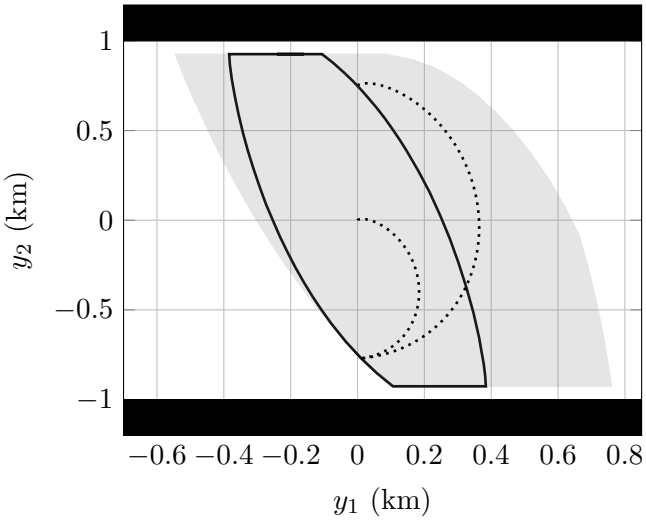


Fig. 5.10: Pursuer's (shaded) and evader's (solid) reachable sets at capture time with the trajectories of the pursuer and evader from their initial outputs to the capture point shown with dotted lines. The black areas represent the linear state constraints or keep-out zones.

5.5.4 Sun Blocking

As a final example of a single-pursuer and single-evader game, a sun blocking problem is considered. In this problem, the pursuer's goal is to place himself between the evader and sun. Reasons for doing so include depriving the evader of solar power and positioning the pursuer for capture of images of the evader while depriving the evader of the capability to image the pursuer. In this example, the pursuer and evader have the same deterministic CW dynamics as in Sections 5.5.1 and 5.5.2, and there is no state constraint. In the Hill frame, a vector pointing to the Sun rotates with a constant angular velocity. Thus, the Sun pointing vector, $s(t)$, is given by

$$s(t) = \begin{bmatrix} \cos(\omega t + \theta) \\ -\sin(\omega t + \theta) \end{bmatrix} \tag{5.45}$$

where ω is angular velocity and is the same quantity that is used in the CW equations as mean motion, and θ is an offset value corresponding to the initial position of the players along the orbit. A value of $-\pi/2$ is used for θ in this example. The game termination criterion must now be changed to

$$\beta s(t) + \mathcal{R}_E(t) \subseteq \mathcal{R}_P(t) \tag{5.46}$$

where β is a non-negative scalar. Conceptually, this means that all points in the evader's reachable set can be projected along the Sun vector into the pursuer's reachable set. Once again by conducting the one-dimensional search, the optimal "capture" time and corresponding Sun vector are

$$t^* = 0.65 \text{ hr}$$
$$s(t^*) = \begin{bmatrix} 0.52 & -0.85 \end{bmatrix}^T$$

The reachable sets at the moment when Sun blocking occurs are shown in Figure 5.11. The area shaded by the pursuer's reachable set at that time instant is shown in gray. It can be

observed from the figure that the evader's reachable set (solid line) is completely shaded by the pursuer's reachable set (black) indicating that the pursuer is capable of blocking the Sun from any position the evader could achieve. It should be noted that the previous statement may only be true at a single instant in time. A future research consideration would be to assure that the pursuer can block the Sun from the evader for an extended period of time.

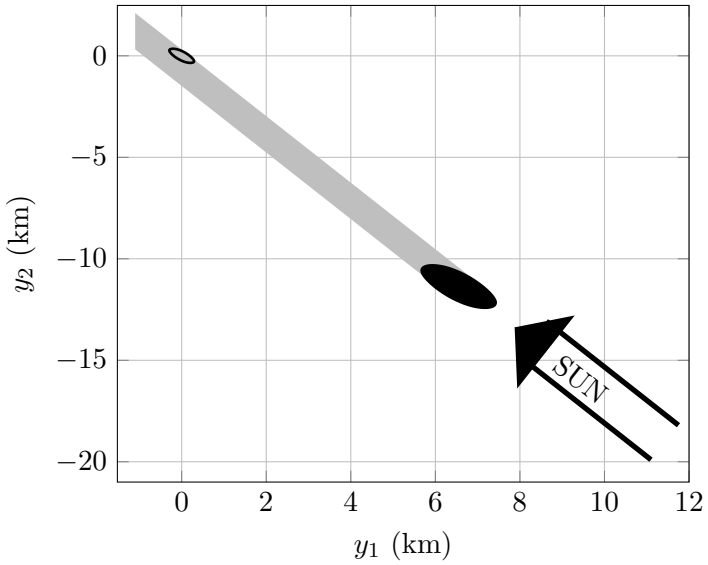


Fig. 5.11: Reachable sets for sun blocking problem: Pursuer's reachable set is marked with filled black, evader's reachable set with solid line and the area shaded by pursuer's reachable set in gray. The arrow represents the direction on which the Sun is shining on the players.

5.6 Examples with Multiple Pursuers or Evaders

This section considers numerical examples with either multiple pursuers or evaders. As in Section 5.5, the agents evolve according to CW dynamics as explained in Section 5.4 in this section as well. Each agent's control set is of the form

$$\mathcal{U}(t) = \{u \in \mathbb{R}^2 \ : \ ||u|| \leq \rho\} \tag{5.47}$$

with ρ a prescribed upper bound on the control magnitude. Possible additional constraints are consistent with those in Equations (5.1)-(5.5).

5.6.1 Multiple Pursuers and Single Evader

Consider a pursuit-evasion problem with 4 pursuers and a single evader. The dynamics of the pursuers and the evader are given by the CW equations (5.36). The initial states of the players are $x_{0,P_1} = [0, 0.5, 0, 0]^\top$, $x_{0,P_2} = [0, -0.5, 0, 0]^\top$, $x_{0,P_3} = [\omega^{-2}, 0.5, 0, 0]^\top$, $x_{0,P_4} = [-\omega^{-2}, -0.5, 0, 0]^\top$ and $x_{0,E} = [0, 0, 0, 0]^\top$. The states of all the players are constrained with

$$D = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{5.48}$$

which are the same D and b specified in (5.39). The control input of all the pursuers is constrained by $\rho_P = 1$ and the control input of the evader by $\rho_E = 0.5$.

Figure 5.12 shows the reachable sets of all the pursuers as well as the evader. The capture happens at the origin. The evader could be captured by either pursuer 1 or pursuer 2 but only pursuer 1's trajectory from its initial point to the capture point is shown. Interestingly the capture happens at the evaders initial location as that is the last point covered by the union of the pursuers' reachable sets. The capture happens at $t^* = 0.84$ hr.

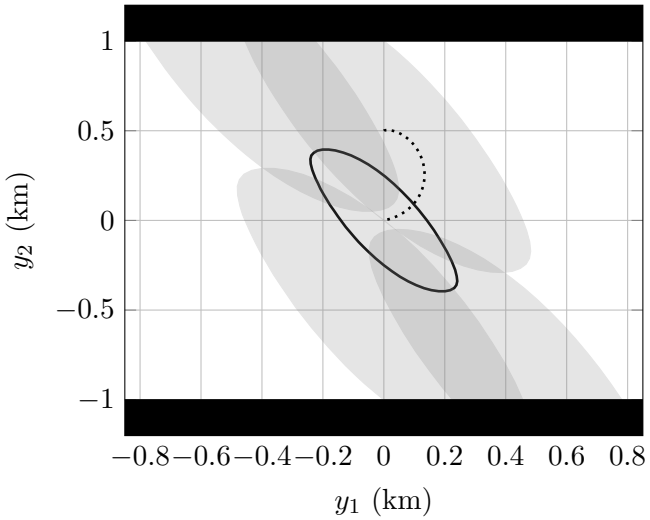


Fig. 5.12: Pursuers' (shaded) and evader's (solid) reachable sets at capture time with the trajectory of the capturing pursuer from its initial output to the capture point shown with dotted line. The black areas represent the linear state constraints or keep-out zones.

5.6.2 Single Pursuer and Multiple Evaders

Lastly, a game with three evaders and a single pursuer is considered. The dynamics of all the agents are again given by the CW equations. The initial state of the pursuer is $x_{0,P} = [0, 0, 0, 0]^\top$ whereas the evaders' initial states are given as $x_{0,E_1} = [0, 0.5, 0, 0]^\top$, $x_{0,E_2} = [0, -0.75, 0, 0]^\top$ and $x_{0,E_3} = [\omega^{-2}, 0, 0, 0]^\top$. There are no state constraints for the agents but the control inputs are constrained with $\rho_P = 1$ for the pursuer and $\rho_E = 0.25$ for the evaders.

Figure 5.13 shows the reachable sets of all the agents at the time instance when the first evader is captured. Figure 5.14 shows the reachable sets of the remaining two evaders as well as the pursuer's set when the second evader is captured. Figure 5.15 shows the reachable sets of the pursuer and the last evader at the time of the last capture which is also the termination time of the game. The captures happen at $t_1 = 0.92$ hr, $t_{1,2} = 2.32$ hr and $t_{1,2,3} = t^* = 6.24$ hr.

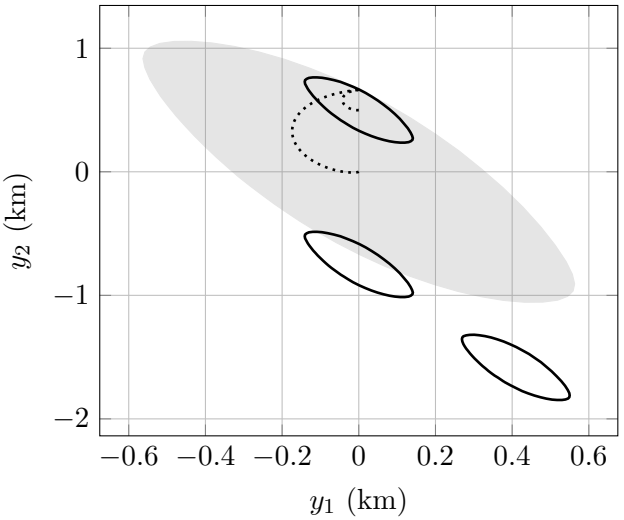


Fig. 5.13: Pursuer's (shaded) and evaders' (solid) reachable sets at the time of the first capture with the trajectories of the pursuer and the captured evader from their initial outputs to the capture point shown with dotted lines.

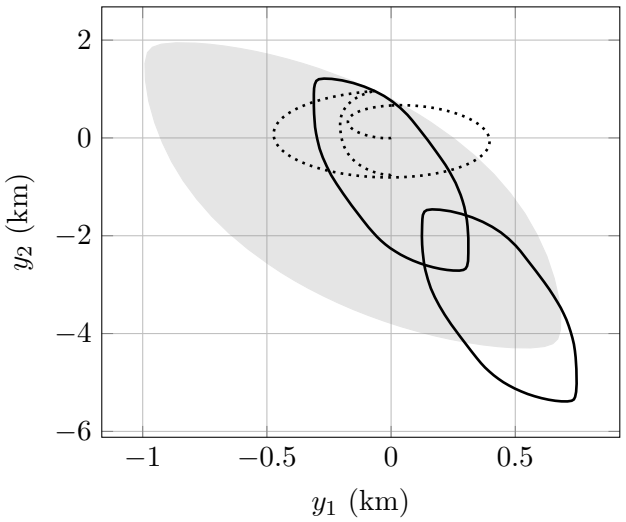


Fig. 5.14: Pursuer's (shaded) and evaders' (solid) reachable sets at the time of the second capture with the trajectories of the pursuer and the captured evader from their initial outputs to the capture point shown with dotted lines.

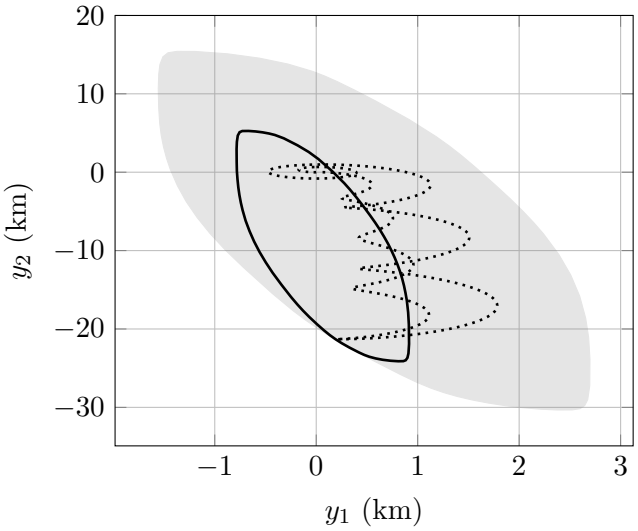


Fig. 5.15: Pursuer's (shaded) and evader's (solid) reachable sets at the time of the third capture with the trajectories of the pursuer and the captured evader from their initial outputs to the capture point shown with dotted lines.

5.7 Summary and Conclusions

This chapter demonstrated a technique using convex optimization to numerically construct reachable sets and to solve time-optimal pursuit-evasion games when the dynamics are linear and all the constraints are convex. The games with 1) a single pursuer and a single evader, 2) multiple pursuers and a single evader, and 3) a single pursuer and multiple evaders were considered. Traditional formulations, such as those based on variational equations, are not tractable in the presence of practical actuator and state constraints. On the contrary, such constraints add minimal complexity to the reachable set method described herein. This is true because construction of the reachable set has been reduced to a sequence of convex programs. Multiple aerospace-related numerical examples were given to demonstrate the method for a constellation of satellites in close proximity in low earth orbit. In conclusion, the reachable set approach leads to a tractable formulation of the constrained, multi-player game problem that leverages modern convex solvers.

CHAPTER 6

CONCLUSIONS AND SUMMARY

This dissertation focused in the application of convexity for control problems; specifically, single-agent problems with linear or nonlinear dynamics and multi-agent problems with linear dynamics were considered. A mixture of convex and non-convex constraints for optimal control problems was considered. The main contributions of this dissertation included: 1) a convexification of linear single-agent problems with annular control constraint, 2) a technique for controlling bounded nonlinear single-agent systems, and 3) a technique for solving multi-agent pursuit-evasion games with linear dynamics and convex control and state constraints. Altogether, the theoretical and computational results demonstrated the significance of convex analysis in solving non-convex control problems.

Chapter 3 presented new convexification results including a sufficient condition for the standard convexification to hold for both free and fixed final time problems, a sufficient condition for the standard convexification to hold for all final times between the minimum feasible time and the optimal time, and a perturbation technique to solve the general fixed time problem as a sequence of convex programs when the final time is greater than the optimal time. In short, the perturbation technique works as follows: perturb the initial point, solve a feasibility problem to the final point, and repeat until convexification works, which is guaranteed to happen. This results in a globally minimizing control. The perturbation technique has practical applications as demonstrated in the Mars landing example. In each call to guidance, the problem was solved in less than one second without customization, suggesting that the new perturbation technique is suitable for real-time guidance applications.

Chapter 4 considered a problem of trajectory design and control of systems with additive nonlinearities. First, systems with additive scalar nonlinearity were considered, and a sufficient condition on when such system can be controlled by bounding linear systems was

derived. This was done for both continuous-time and discrete-time systems with control constraints and convex state constraints. More specifically, the ability to solve problems with non-convex control constraints was demonstrated with an example with quantized control. The scalar results were then generalized for discrete-time systems with additive nonlinearities and convex state and control constraints. Two different methods for solving nonlinear problems of this type were derived using the sufficient condition. These methods were demonstrated with spacecraft trajectory design examples. A computational method for multi-dimensional case was also discussed in the last section of Chapter 4. The computational method was compared against the classic feedback linearization technique via numerical examples. The introduced method outperformed the feedback linearization in the given examples. The future research in the area of control of nonlinear systems using their bounding linear ones should try to find answers to the following questions: Is it possible to control nonlinear systems by bounding linear ones when the control inputs are not affine? Can the infinite-dimensional continuous-time approach be used in the control applications instead of the finite-dimensional discrete-time one? Is there a less conservative sufficient condition, or even a necessary condition for systems that can be found?

A technique using convex optimization to numerically construct reachable sets and to solve time-optimal pursuit-evasion games when the dynamics are linear and all the constraints are convex was discussed in Chapter 5. The games with 1) a single pursuer and a single evader, 2) multiple pursuers and a single evader, and 3) a single pursuer and multiple evaders were considered. Traditional formulations, such as those based on variational equations, are not tractable in the presence of practical actuator and state constraints. On the contrary, such constraints add minimal complexity to the reachable set method described in the chapter. This is true because construction of the reachable set was reduced to a sequence of convex programs. Multiple aerospace-related numerical examples were given to demonstrate the method for a constellation of satellites in close proximity in low earth orbit. In conclusion, the reachable set approach leads to a tractable formulation of the constrained, multi-player game problem that leverages modern convex solvers. The com-

putation of the reachable sets is currently not fast enough using the algorithm given in the chapter to consider this method real-time capable. Applying a method for faster reachable set computation is a possible future research topic if onboard computing is desired.

REFERENCES

[1] Lowe, G., and Zohdy, M. A., 2009, "A technique for using H2 and H-infinity robust state estimation on nonlinear systems," In 2009 IEEE International Conference on Electro/Information Technology, IEEE, pp. 109–115.

[2] Khalil, H. K., 2001, *Nonlinear Systems* Pearson.

[3] Rugh, W. J., 1981, *Nonlinear System Theory* The Johns Hopkins University Press.

[4] Vidyasagar, M., 2002, *Nonlinear systems analysis* SIAM.

[5] Açıkmeşe, B., and Ploen, S., 2007, "Convex programming approach to powered descent guidance for Mars landing," *AIAA Journal of Guidance, Control and Dynamics,* **30**, pp. 1353–1366.

[6] Açıkmeşe, B., Carson, J., and Blackmore, L., 2013, "Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem," *IEEE Transactions on Control Systems Technology,* **21**, pp. 2104–2113.

[7] Açıkmeşe, B., and Blackmore, L., 2011, "Lossless convexification for a class of optimal control problems with nonconvex control constraints," *Automatica,* **47**, pp. 341–347.

[8] Harris, M. W., and Açıkmeşe, B., 2014, "Lossless convexification of non-convex optimal control problems for state constrained linear systems," *Automatica,* **50**, pp. 2304–2311.

[9] Harris, M. W., 2021, "Optimal control on disconnected sets using extreme point relaxations and normality approximations," *IEEE Transactions on Automatic Control,* **66**(12).

[10] Malyuta, D., and Açıkmeşe, B., 2020, "Lossless convexification of optimal control problems with semi-continuous inputs," In IFAC World Congress 2020, IFAC.

[11] Harris, M. W., 2014, *Lossless Convexification of Optimal Control Problems* The University of Texas at Austin, Austin, TX.

[12] Açıkmeşe, B., and Ploen, S. R., 2005, "A powered descent guidance algorithm for Mars pinpoint landing," In AIAA Guidance, Navigation, and Control Conference (San Francisco, California), AIAA.

[13] Ploen, S. R., Açıkmeşe, B., and Wolf, A., 2006, "A comparison of powered descent guidance laws for Mars pinpoint landing," In AIAA Guidance, Navigation, and Control Conference (Keystone, Colorado), AIAA.

[14] Blackmore, L., Açıkmeşe, B., and Scharf, D., 2010, "Minimum landing error powered descent guidance for Mars landing using convex optimization," *Journal of Guidance, Control and Dynamics,* **33**, pp. 1161–1171.

[15] Blackmore, L., 2016, "Autonomous precision landing of space rockets," *The Bridge : Linking Engineering and Society (National Academy of Engineering),* **26**.

[16] Blackmore, L., Açıkmeşe, B., and Carson, J. M., 2012, "Lossless convexification of control constraints for a class of nonlinear optimal control problems," *Systems and Control Letters,* **61**, pp. 863–871.

[17] Kunhippurayil, S., and Harris, M. W., 2022, "Strong observability as a sufficient condition for non-singularity and lossless convexification in optimal control with mixed constraints," *Control Theory and Technology*, pp. 1–13.

[18] Korda, M., and Mezic, I., 2018, "Linear redictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica,* **93**.

[19] Brunton, S. L., Brunton, B. W., Proctor, J. L., and Kutz, J. N., 2016, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PLoS ONE,* **11**(2).

[20] Sename, O., Gaspar, P., and Bokor, J., 2013, *Robust control and linear parameter varying approaches: application to vehicle dynamics*, Vol. 437 Springer.

[21] Robert, D., Sename, O., and Simon, D., 2007, "A reduced polytopic LPV synthesis for a sampling varying controller: experimentation with a T inverted pendulum," In 2007 European Control Conference (ECC), IEEE, pp. 4316–4323.

[22] Atoui, H., Sename, O., Milanés, V., and Martinez, J. J., 2021, "LPV-based autonomous vehicle lateral controllers: A comparative analysis," *IEEE Transactions on Intelligent Transportation Systems.*

[23] Do, A. L., 2011, "LPV approach for the vehicles dynamics robust control: joint comfort and safety improvement.," PhD thesis, PhD thesis. Université de Grenoble.

[24] He, T., 2019, "Smooth switching LPV control and its applications," PhD thesis, Michigan State University.

[25] Pylorof, D., and Bakolas, E., 2015, "Nonlinear control under polytopic input constraints with application to the attitude control problem," In 2015 American Control Conference (ACC), pp. 4555–4560.

[26] Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mischenko, E. F., 1986, *The Mathematical Theory of Optimal Processes* Gordon and Breach Science Publishers.

[27] Liberzon, D., 2012, *Calculus of Variations and Optimal Control theory* Princeton University Press.

[28] Hull, D. G., 1997, "Conversion of optimal control problems into parameter optimization problems," *Journal of Guidance, Control and Dynamics,* **20**, pp. 57–60.

[29] Reynolds, T. P., and Mesbahi, M., 2020, "The crawling phenomenon in sequential convex programming," In 2020 American Control Conference (ACC), pp. 3613–3618.

[30] Rawlings, J. B., Mayne, D. Q., and Diehl, M. M., 2019, *Model Predictive Control: Theory, Computation, and Design* Nob Hill Publishing.

[31] Harris, M. W., and Açıkmeşe, B., 2013, "Maximum divert for planetary landing using convex optimization," *Journal of Optimization Theory and Applications,* **162**, pp. 975–995.

[32] Kunhippurayil, S., Harris, M. W., and Jansson, O., 2021, "Lossless convexification of optimal control problems with annular control constraints," *Automatica,* **133**.

[33] Woodford, N. T., and Harris, M. W., 2022, "Geometric properties of time optimal controls with state constraints using strong observability," *IEEE Transactions on Automatic Control,* **67**(12).

[34] Rugh, W. J., 1993, *Linear System Theory* Prentice Hall.

[35] Trentelman, H. L., Stoorvogel, A. A., and Hautus, M., 2001, *Control Theory for Linear Systems* Springer.

[36] Berkovitz, L. D., 1975, *Optimal Control Theory* Springer-Verlag.

[37] Lewis, A. D., 2001, "A brief on controllability of nonlinear systems," *Preprint.*

[38] Peng, J., Roos, C., and Terlaky, T., 2001, *Self-regularity: a new paradigm for primal-dual interior-point algorithms* Princeton Series in Applied Mathematics.

[39] Nesterov, Y., and Nemirovsky, A., 1994, *Interior-point polynomial methods in convex programming* SIAM.

[40] Bryson, A. E., and Ho, Y.-C., 1975, *Applied optimal control: optimization, estimation and control* CRC Press.

[41] Johnson, P. A., 2009, "Numerical solution methods for differential game problems," Master's thesis, Massachusetts Institute of Technology.

[42] Horie, K., and Conway, B. A., 2004, "Genetic algorithm preprocessing for numerical solution of differential games problems," *Journal of guidance, control, and dynamics,* **27**(6), pp. 1075–1078.

[43] Horie, K., and Conway, B. A., 2006, "Optimal fighter pursuit-evasion maneuvers found via two-sided optimization," *Journal of guidance, control, and dynamics,* **29**(1), pp. 105–112.

[44] Carr, R. W., Cobb, R. G., Pachter, M., and Pierce, S., 2018, "Solution of a pursuit–evasion game using a near-optimal strategy," *Journal of Guidance, Control, and Dynamics,* **41**(4), pp. 841–850.

[45] Basar, T., 1987, "Relaxation techniques and asynchronous algorithms for online computation of non-cooperative equilibria," *Journal of Economic Dynamics and Control,* **11**(4), pp. 531–549.

[46] Uryasev, S., and Rubinstein, R., 1994, "On relaxation algorithms in computation of noncooperative equilibria," *IEEE Transactions on Automatic Control,* **39**(6), pp. 1263–1267.

[47] Mizukami, K., and Eguchi, K., 1977, "A geometrical approach to problems of pursuit-evasion games," *Journal of the Franklin Institute,* **303**(4), pp. 371–384.

[48] Sun, W., Tsiotras, P., Lolla, T., Subramani, D. N., and Lermusiaux, P. F., 2017, "Multiple-pursuer/one-evader pursuit–evasion game in dynamic flowfields," *Journal of guidance, control, and dynamics,* **40**(7), pp. 1627–1637.

[49] Chung, C. F., Furukawa, T., and Goktogan, A. H., 2006, "Coordinated control for capturing a highly maneuverable evader using forward reachable sets," In Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., IEEE, pp. 1336–1341.

[50] Chung, C. F., and Furukawa, T., 2008, "A reachability-based strategy for the time-optimal control of autonomous pursuers," *Engineering Optimization,* **40**(1), pp. 67–93.

[51] Salmon, D. M., and HElNE, W., 1973, "Reachable sets analysis-an efficient technique for performing missile/sensor tradeoff studies," *AIAA Journal,* **11**(7), pp. 927–931.

[52] Zanardi, C., Hervé, J.-Y., and Cohen, P., 1995, "Escape strategy for a mobile robot under pursuit," In 1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century, Vol. 4, IEEE, pp. 3304–3309.

[53] Jansson, O., and Harris, M. W., 2022, "Nonlinear control algorithm for systems with convex polytope bounded nonlinearities," In 2022 Intermountain Engineering, Technology, and Computing Conference.

[54] Jansson, O., and Harris, M., 2022, "A Technique for Constrained and Quantized Control of Nonlinear Systems using Second-order Cone Programming," *ASME Letters in Dynamic Systems and Control*, 12, pp. 1–11.

[55] Jansson, O., and Harris, M. W., 2023, "Convex optimization-based techniques for trajectory design and control of nonlinear systems with polytopic range," *Aerospace,* **10**(1).

[56] Jansson, O., Harris, M., and Geller, D., 2021, "A parallelizable reachable set method for pursuit-evasion games using interior-point methods," In 2021 IEEE Aerospace Conference (50100), IEEE, pp. 1–9.

[57] Jansson, O., and Harris, M. W., 2022, "A geometrical, reachable set approach for constrained pursuit-evasion games with multiple pursuers and evaders," *Aerospace* Submitted.

[58] Boyd, S., and Vandenberghe, L., 2004, *Convex Optimization* Cambridge University Press.

[59] Moon, T. K., and Stirling, W. C., 2000, *Mathematical methods and algorithms for signal processing* No. 621.39: 51 MON.

[60] Hespanha, J. P., 2018, *Linear systems theory* Princeton university press.

[61] Elnagar, G., Kazemi, M. A., and Razzaghi, M., 1995, "The pseudospectral Legendre method for discretizing optimal control problems," *IEEE transactions on Automatic Control,* **40**(10), pp. 1793–1796.

[62] Boyd, J. P., 2001, *Chebyshev and Fourier spectral methods* Courier Corporation.

[63] Klumpp, A. R., 1974, "Apollo lunar descent guidance," *Automatica,* **10**, pp. 133–146.

[64] Cherry, G. W., 1997, "A general, explicit, optimizing guidance law for rocket-propelled spaceflight," In Guidance, Navigation, and Control Conference, AIAA Paper 1997-3709, AIAA.

[65] D'Souza, C. S., 1964, "An optimal guidance law for planetary landing," In Astrodynamics, Guidance and Control Conference, AIAA Paper 1964-0638.

[66] Lu, P., 2020, "Theory of fraction-polynomial powered descent guidance," *Journal of Guidance, Control, and Dynamics,* **43**, pp. 398–409.

[67] Martin, D. T., Sievers, R. F., O'Brien, R. M., and Rice, A. L., 1967, "Saturn V guidance, navigation, and targeting," *Journal of Spacecraft and Rockets,* **4**, pp. 891–898.

[68] Chandler, D. C., and Smith, I. E., 1967, "Development of the iterative guidance mode with its applications to various vehicles and missions," *Journal of Spacecraft and Rockets,* **4**, pp. 898–903.

[69] McHenry, R. L., Brand, T. J., Long, A. D., Cockrell, B. F., and Thibodeau, J. R., 1979, "Space shuttle ascent guidance, navigation, and control," *Journal of the Astronautical Sciences,* **27**, pp. 1–38.

[70] Hull, D., 2011, "Optimal guidance for quasi-planar lunar descent with throttling," In AAS/AIAA Space Flight Mechanics Meeting, AAS 11-169, AAS/AIAA.

[71] Lu, P., and Liu, X., 2013, "Autonomous trajectory planning for rendezvous and proximity operations by conic optimization," *Journal of Guidance, Control, and Dynamics,* **36**, pp. 375–389.

[72] Lu, P., 2019, "Augmented Apollo powered descent guidance," *Journal of Guidance, Control, and Dynamics,* **42**, pp. 447–457.

[73] Lu, P., 2017, "Introducing computational guidance and control," *Journal of Guidance, Control, and Dynamics,* **40**.

[74] Liu, X., Lu, P., and Pan, B., 2017, "Survey of convex optimization for aerospace applications," *Astrodynamics,* **1**, pp. 23–40.

[75] Liu, X., and Lu, P., 2014, "Solving nonconvex optimal control problems by convex optimization," *Journal of Guidance, Control, and Dynamics,* **37**, pp. 750–765.

[76] Pinson, R., and Lu, P., 2015, "Rapid generation of optimal asteroid powered descent trajectories," In AAS/AIAA Astrodynamics Specialist Conference, AAS 15-616, AAS/AIAA.

[77] Pinson, R., and Lu, P., 2016, "Trajectory design employing convex optimization for landing on irregularly shaped asteroids," In AAS/AIAA Astrodynamics Specialist Conference, AIAA 2016-5378, AAS/AIAA.

[78] Bixby, R. E., 2012, "A brief history of linear and mixed-integer programming computation," *Documenta Mathematica*, pp. 107–121.

[79] Harris, M. W., and Açıkmeşe, B., 2013, "Lossless convexification for a class of optimal control problems with linear state constraints," In IEEE Conference on Decision and Control (Florence, Italy), IEEE.

[80] Harris, M. W., and Açıkmeşe, B., 2013, "Lossless convexification for a class of optimal control problems with quadratic state constraints," In American Control Conference (Washington, D.C.).

[81] Mattingley, J., and Boyd, S., 2012, "CVXGEN-a code generator for embedded convex optimization," *Optimization and Engineering,* **13**, pp. 1–27.

[82] Chu, E., Parikh, N., Domahidi, A., and Boyd, S., 2013, "Code generation for embedded second-order cone programming," In European Control Conference (Zurich, Switzerland).

[83] Domahidi, A., Chu, E., and Boyd, S., 2013, "ECOS: An SOCP solver for embedded systems," In European Control Conference (Zurich, Switzerland).

[84] Dueri, D., Açıkmeşe, B., Scharf, D., and Harris, M. W., 2017, "Customized real-time interior-point method for onboard powered descent guidance," *Journal of Guidance, Control and Dynamics,* **40**, pp. 197–212.

[85] Açıkmeşe, B., Aung, M., Casoliva, J., Mohan, S., Johnson, A., Scharf, D., Masten, D., Scotkin, J., Wolf, A., and Regehr, M. W., 2013, "Flight testing of trajectories computed by G-FOLD: Fuel optimal large divert guidance algorithm for planetary landing," In AAS/AIAA Spaceflight Mechanics Meeting, AAS/AIAA.

[86] Scharf, D., Regehr, M. W., Dueri, D., Açıkmeşe, B., Vaughan, G. M., and Benito, J., 2014, "Adapt: Demonstrations of onboard large-divert guidance with a resuable launch vehicle," In IEEE Aerospace Conference, IEEE.

[87] Hartl, R., Sethi, S., and Vickson, R., 1995, "A survey of the maximum principles for optimal control problems with state constraints," *SIAM Review,* **37**(2), June, pp. 181–218.

[88] Hermes, H., and LaSalle, J. P., 1969, *Functional analysis and time optimal control* Academic Press.

[89] Jurdjevic, V., 1997, *Geometric Control Theory* Cambridge University Press.

[90] Gurobi Optimization, LLC, 2019, Gurobi Optimizer Reference Manual.

[91] , 2021, *MATLAB 2021a* The Mathworks, Inc.

[92] Löfberg, J., 2004, "Yalmip : A toolbox for modeling and optimization in MATLAB," In In Proceedings of the CACSD Conference.

[93] Ahmed, H., Rios, H., Ayalew, B., and Wang, Y., 2018, "Robust output tracking control for Van der Pol oscillator: A sliding-mode differentiator approach," In 2018 American Control Conference (ACC), pp. 5350–5355.

[94] Jin, L., Mei, J., and Li, L., 2014, "Chaos control of parametric driven duffing oscillators," *Applied Physics Letters,* **104**.

[95] Lowe, G., and Zohdy, M., 2010, "Modeling nonlinear systems using multiple piecewise linear equations," *Nonlinear Analysis: Modelling and Control,* **15**(4), pp. 451–458.

[96] LaValle, S. M., 2006, *Planning Algorithms* Cambridge University Press.

[97] Flores-Abad, A., Ma, O., Pham, K., and Ulrich, S., 2014, "A review of space robotics technologies for on-orbit servicing," *Progress in aerospace sciences,* **68**, pp. 1–26.

[98] Li, W.-J., Cheng, D.-Y., Liu, X.-G., Wang, Y.-B., Shi, W.-H., Tang, Z.-X., Gao, F., Zeng, F.-M., Chai, H.-Y., Luo, W.-B., et al., 2019, "On-orbit service (OOS) of spacecraft: A review of engineering developments," *Progress in Aerospace Sciences,* **108**, pp. 32–120.

[99] Tsuda, Y., Yoshikawa, M., Abe, M., Minamino, H., and Nakazawa, S., 2013, "System design of the Hayabusa 2—asteroid sample return mission to 1999 JU3," *Acta Astronautica,* **91**, pp. 356–362.

[100] Gaudet, B., Linares, R., and Furfaro, R., 2020, "Terminal adaptive guidance via reinforcement meta-learning: Applications to autonomous asteroid close-proximity operations," *Acta Astronautica,* **171**, pp. 1–13.

[101] D'Amico, S., Benn, M., and Jørgensen, J. L., 2014, "Pose estimation of an uncooperative spacecraft from actual space imagery," *International Journal of Space Science and Engineering,* **2**(2), pp. 171–189.

[102] Stastny, N. B., and Geller, D. K., 2008, "Autonomous optical navigation at Jupiter: a linear covariance analysis," *Journal of Spacecraft and Rockets,* **45**(2), pp. 290–298.

[103] Bradley, N., Olikara, Z., Bhaskaran, S., and Young, B., 2020, "Cislunar navigation accuracy using optical observations of natural and artificial targets," *Journal of Spacecraft and Rockets,* **57**(4), pp. 777–792.

[104] Curtis, H., 2009, *Orbital mechanics for engineering students*, second ed. Butterworth-Heinemann.

[105] Harris, M. W., and Woodford, N. T., 2022, "Equilibria, periodicity, and chaotic behavior in spherically constrained relative orbital motion," *Nonlinear Dynamics*.

[106] Clohessy, W. H., and Wiltshire, R. S., 1960, "Terminal guidance system for satellite rendezvous," *Journal of Aerospace Systems,* **27**, pp. 653–658.

[107] Shuster, M. D., 1993, "The kinematic equation for the rotation vector," *IEEE Transactions on Aerospace and Electronic Systems,* **29**(1), pp. 263–267.

[108] Markley, F. L., and Crassidis, J. L., 2014, *Fundamentals of spacecraft attitude determination and control*, Vol. 1286 Springer.

[109] Bortz, J. E., 1971, "A new mathematical formulation for strapdown inertial navigation," *IEEE transactions on aerospace and electronic systems*(1), pp. 61–66.

[110] Isaacs, R., 1999, *Differential games : a mathematical theory with applications to warfare and pursuit, control and optimization* Dover Publications, Mineola, N.Y.

[111] Weintraub, I. E., Pachter, M., and Garcia, E., 2020, "An introduction to pursuit-evasion differential games," In 2020 American Control Conference (ACC), IEEE, pp. 1049–1066.

[112] Shinar, J., and Gutman, S., 1979, "Recent advances in optimal pursuit and evasion," In 1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes, IEEE, pp. 960–965.

[113] Shinar, J., Glizer, V. Y., and Turetsky, V., 2009, "A pursuit-evasion game with hybrid pursuer dynamics," *European Journal of Control,* **15**(6), pp. 665–684.

[114] Shinar, J., 1981, "Solution techniques for realistic pursuit-evasion games," In *Control and Dynamic Systems*, Vol. 17. Elsevier, pp. 63–124.

[115] Imado, F., and Kuroda, T., 2005, "A method to solve missile-aircraft pursuit-evasion differential games," *IFAC Proceedings Volumes,* **38**(1), pp. 176–181.

[116] Greenwood, N., 1992, "A differential game in three dimensions: The aerial dogfight scenario," *Dynamics and Control,* **2**(2), pp. 161–200.

[117] Calise, A. J., and Yu, X.-m., 1985, "An analysis of a four state model for pursuit-evasion games," In 1985 24th IEEE Conference on Decision and Control, IEEE, pp. 1119–1121.

[118] Shen, D., Pham, K., Blasch, E., Chen, H., and Chen, G., 2011, "Pursuit-evasion orbital game for satellite interception and collision avoidance," In Sensors and Systems for Space Applications IV, Vol. 8044, International Society for Optics and Photonics, p. 80440B.

[119] Blasch, E. P., Pham, K., and Shen, D., 2012, "Orbital satellite pursuit-evasion game-theoretical control," In 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), IEEE, pp. 1007–1012.

[120] Stupik, J., Pontani, M., and Conway, B., 2012, "Optimal pursuit/evasion spacecraft trajectories in the Hill reference frame," In AIAA/AAS astrodynamics specialist conference, p. 4882.

[121] Shen, D., Jia, B., Chen, G., Pham, K., and Blasch, E., 2017, "Game optimal sensor management strategies for tracking elusive space objects," In 2017 IEEE Aerospace Conference, IEEE, pp. 1–8.

[122] Zeng, X., Yang, L., Zhu, Y., and Yang, F., 2020, "Comparison of two optimal guidance methods for the long-distance orbital pursuit-evasion game," *IEEE Transactions on Aerospace and Electronic Systems*.

[123] Colonius, F., and Szolnoki, D., 2001, "Algorithms for computing reachable sets and control sets," *IFAC Proceedings Volumes,* **34**(6), pp. 723–728.

[124] Girard, A., Le Guernic, C., and Maler, O., 2006, "Efficient computation of reachable sets of linear time-invariant systems with inputs," In International Workshop on Hybrid Systems: Computation and Control, Springer, pp. 257–271.

[125] Varaiya, P., 2000, "Reach set computation using optimal control," In *Verification of Digital and Hybrid Systems*. Springer, pp. 323–331.

[126] Dueri, D., Raković, S. V., and Açıkmeşe, B., 2016, "Consistently improving approximations for constrained controllability and reachability," In 2016 European Control Conference (ECC), IEEE, pp. 1623–1629.

[127] Dueri, D., Açıkmeşe, B., Baldwin, M., and Erwin, R. S., 2014, "Finite-horizon controllability and reachability for deterministic and stochastic linear control systems with convex constraints," In 2014 American Control Conference, IEEE, pp. 5016–5023.

[128] Yang, R., and Liu, X., 2022, "Reachable set computation of linear systems with nonconvex constraints via convex optimization," *Automatica,* **146**, p. 110632.

[129] Pecsvaradi, T., and Narendra, K. S., 1971, "Reachable sets for linear dynamical systems," *Information and control,* **19**(4), pp. 319–344.

[130] Blackmore, L., and Ono, M., 2009, "Convex chance constrained predictive control without sampling," In AIAA Guidance, Navigation, and Control Conference, p. 5876.

CURRICULUM VITAE

Olli N. Jansson

Published Journal Articles

- S. Kunhippurayil, M. Harris, **O. Jansson**, "Lossless Convexification of Optimal Control Problems with Annular Control Constraints," *Automatica*, vol. 133, 2021: 109848.
- **O. Jansson**, M. Harris, "A Technique for Constrained and Quantized Control of Nonlinear Systems Using Second-Order Cone Programming," *ASME Letters in Dynamic Systems and Control*, 2022. doi: https://doi.org/10.1115/1.4056551
- **O. Jansson**, M. Harris, "Convex Optimization-based Techniques for Trajectory Design and Control of Nonlinear Systems with Polytopic Range," *Aerospace*, vol. 10(1), 2023: 71.
- **O. Jansson**, M. Harris, "A Geometrical, Reachable Set Approach for Constrained Pursuit-Evasion Games with Multiple Pursuers and Evaders," *Aerospace*, In Review 2022.

Published Conference Papers

- P. Karra, **O. Jansson**, "A Cost-effective Laboratory Setup for Engine and Chassis-Dynamometer," In 2019 ASEE Annual Conference & Exposition.
- **O. Jansson**, M. Harris, D. Geller, "A Parallelizable Reachable Set Method for Pursuit-Evasion Games Using Interior-Point Methods," In 2021 IEEE Aerospace Conference.

- **O. Jansson**, M. Harris, "Nonlinear Control Algorithm for Systems with Convex Polytope Bounded Nonlinearities," In 2022 Intermountain Engineering, Technology, and Computing Conference.