Drift Counteraction Optimal Control: Theory and Applications to Autonomous Cars and Spacecraft

by

Robert A. E. Zidek

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2017

Doctoral Committee:

Professor Ilya V. Kolmanovsky, Chair
Professor Ella M. Atkins
Associate Professor Anouck R. Girard
Professor Jing Sun

Robert A. E. Zidek

robzidek@umich.edu

ORCID iD: 0000-0002-8684-5121

To my grandmother, Thea Weß.

ACKNOWLEDGMENTS

First and foremost, I sincerely thank my advisor Prof. Ilya Kolmanovsky. Words cannot describe how grateful I am for his steady support, help, and guidance over the years, which have been invaluable for my personal and professional growth. He is truly a remarkable person and one of the smartest people I know, who can always tell a joke and put a smile on your face. I couldn't have wished for a better advisor and mentor.

I am grateful to Prof. Ella Atkins, Prof. Anouck Girard, and Prof. Jing Sun for joining my dissertation committee and offering useful feedback and suggestions about my research. I especially thank Prof. Ella Atkins. Her classes on Aerospace Information Systems and Robotics have made me a better engineer and researcher.

I also acknowledge the National Science Foundation (Award Number EECS 1404814) for funding this project.

Many thanks go to Prof. Alberto Bemporad. I have learned a lot about MPC and mathematical programming from him. His help has been crucial in the development of the MPC approaches in this work. I also thank Dr. Rohit Gupta for helping me understand and apply advanced mathematical concepts and Dr. Chris "Crispy" Petersen for his valuable insights into space flight applications and feedback about my work. I also thank my other labmates and colleagues for their feedback that has helped me improve this work.

Finally, thank you to my family and friends for supporting me throughout my life and laughing with me every day. Special thanks go to Julia Hlavac for her tremendous love and support, and to my wonderful parents, Edeltraud and Walter, for everything.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

ACC adaptive cruise control

ADP approximate/adaptive dynamic programming

CW Clohessy-Wiltshire

DCOC drift counteraction optimal control

DP dynamic programming

ECI Earth-centered inertial

E-W East-West

GEO geostationary Earth orbit

HJB Hamilton-Jacobi-Bellman

iff if and only if

LEO low Earth orbit

LP linear program

LQ linear quadratic

MILP mixed-integer linear program

MINLP mixed-integer nonlinear program

MPC model predictive control

NLP nonlinear program

NN neural network

N-S North-South

ODE ordinary differential equation

PDE partial differential equation

RW reaction wheel

SMPC stochastic model predictive control

SRP solar radiation pressure

TPBVP two point boundary value problem

UAV unmanned aerial vehicle

USC upper semi-continuous

VDP van der Pol

VI value iteration

VSI variable specific impulse

ABSTRACT

Many engineering systems are subject to persistent disturbances or dynamics that cause the process variables to drift. This dissertation studies the problem of how to control such systems in order to maximize the time or total yield before prescribed constraints are violated. This problem is referred to as drift counteraction optimal control (DCOC) since the controller may be viewed as counteracting drift in order to delay constraint violation.

The first part of this dissertation focuses on deterministic DCOC problems, i.e., problems where the system behavior is described by a deterministic model. Conditions for the existence of a solution are derived and an optimal control strategy is characterized in terms of the value function. Moreover, properties of the first exit-time, i.e., the first time instant at which constraint violation occurs, and of the value function are studied, where both are shown to be upper semi-continuous (USC) with respect to the state variables under suitable conditions. New algorithms based on dynamic programming (DP), approximate dynamic programming (ADP), and model predictive control (MPC) are developed to obtain solutions or good-quality approximations of solutions. In terms of DP, an enhanced version of the value iteration (VI) algorithm is proposed that converges to the value function faster than conventional VI in a numerical setting. Based on the enhanced VI algorithm, an ADP approach is obtained to mitigate the curse of dimensionality. Another DP-based algorithm, referred to as base-trajectory VI, is proposed, which converges to the value function by gradually connecting pieces of an optimal control policy. A mixed-integer nonlinear program is derived that obtains open-loop solutions to deterministic DCOC problems and good-quality suboptimal solutions are obtained with a similar nonlinear program without

integer variables. In the linear systems case, a mixed-integer linear program (MILP) and a standard linear program (LP) are formulated to obtain open-loop solutions and good-quality approximations of solutions, respectively. Based on linear model approximation, the MILP and LP are used to implement an MPC strategy that can be effective in DCOC of nonlinear systems. New applications of deterministic DCOC are proposed with a focus on spacecraft control, where the effectiveness of the developed approaches in delaying constraint violation is demonstrated in several numerical case studies.

In the second part of this dissertation, the assumption of perfect information is relaxed and the developments for the case of deterministic DCOC are extended to the case of stochastic systems. Conditions are derived under which the average first exit-time and value function are bounded from above, which are necessary conditions for the existence of a solution to stochastic DCOC problems. Further conditions are provided that guarantee the existence of a solution and an optimal control policy is characterized. Moreover, the average first exit-time is shown to be USC with respect to the state variables under suitable assumptions. The enhanced VI algorithm for the deterministic case is extended to stochastic systems, where the proof of convergence requires a different approach. Similar to the deterministic case, an ADP approach is presented that mitigates the curse of dimensionality. Another contribution is a novel tree-based stochastic MPC (SMPC) approach to solve stochastic DCOC problems. A scenario tree with a specified number of tree nodes is used to encode the most likely system behavior, where each path on the tree corresponds to a distinct disturbance scenario. For linear discrete-time systems with an additive random disturbance, an MILP obtains solutions arbitrarily close to the optimal solution for a sufficient number of tree nodes. In order to compensate for an incomplete scenario tree and/or unmodeled effects, feedback is provided by recomputing the MILP solution over a receding time horizon based on the current disturbance and state variables. New applications of stochastic DCOC are introduced with a focus on automated and autonomous driving and a variety of numerical case studies of such DCOC problems are treated in this dissertation.

CHAPTER 1

Introduction

1.1 Problem Statement

The problem addressed in this dissertation can be stated as follows: given a deterministic (stochastic) system and a set of prescribed constraints on the system's process and control variables, find a control strategy that maximizes the (expected value of the) time or total yield before at least one of the constraints is violated. For each admissible control strategy, it is assumed that constraint violation occurs in finite time (with probability one), so that the problem may be defined. Such problems are referred to as drift counteraction optimal control (DCOC) problems since the solution may be viewed as counteracting drift imposed by disturbances or system dynamics in order to delay constraint violation and maximize (expected) total yield.

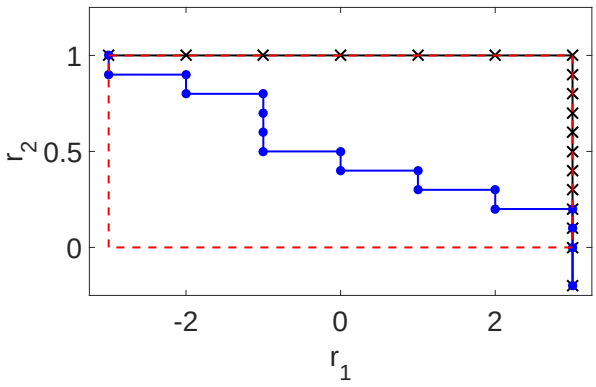

Figure 1.1: DCOC problem subject to system (1.1): state trajectories in r_1-r_2 plane of two different solutions. The dashed red line indicates the constraints.

As an example, consider the deterministic discrete-time system,

$$
\begin{aligned}
r_{1,t+1} &= r_{1,t} - u_t + 1 \\
r_{2,t+1} &= r_{2,t} - 0.1u_t,
\end{aligned}
\tag{1.1}
$$

where the states at a time instant $t \in \mathbb{Z}_{\geq 0}$ are given by $r_{1,t}$ and $r_{2,t}$, and $u_t \in [0, 1]$ denotes the control input. The objective is to find a control strategy that maximizes the time that $r_{1,t} \in [-3, 3]$ and $r_{2,t} \in [0, 1]$. The solution to this DCOC problem is not unique. For initial states $r_{1,0} = -3$ and $r_{2,0} = 1$, the optimal number of time steps until constraint violation is 17. One possible solution is given by the control sequence $0, 0, ..., 0, 1, 1, ..., 1$, the corresponding state trajectory to which is plotted in the r_1-r_2 plane in Figure 1.1 (black line). In addition to the constraints (dashed red line), the state trajectory of another solution is also plotted in Figure 1.1 (blue line).

The objective of the dissertation is to develop methods to systematically compute such optimal control policies, provide insight into the solution properties, and to apply the developed methods to solve engineering problems.

1.2 Motivation

DCOC problems can be found in many engineering systems, in particular, those with

- large persistent disturbances (e.g., wind gusts or drag),
- limited control authority (e.g., underactuated systems),
- finite resources (fuel, energy, component life, etc.),

causing the process variables of the system to drift. In order to maintain the operation of the system in a safe and/or efficient region for as long as possible, control algorithms need to be designed that counteract the drift in an optimal way.

For example, consider a satellite in geostationary Earth orbit (GEO), which is the most frequently used Earth orbit. Typical position windows for GEO satellites are around ± 0.05 deg in longitude and latitude. The satellite is subject to orbital perturbations (gravity of the Moon and Sun as well as other planets, solar radiation pressure (SRP), etc.), requiring regular thrusting for station keeping. Eventually, the satellite runs out of fuel and position constraints are violated. The objective of DCOC is to find a thrust strategy that delays this event in order to increase the operational time of the satellite.

This is illustrated in Figure 1.2, which shows the trajectory of an example GEO satellite when using a DCOC strategy. The left plot in Figure 1.2 shows the satellite position relative to its target orbit in the plane defined by the North-South (N-S) and East-West (E-W) directions. The available fuel (the fuel mass is normalized here) can be seen in the right plot. The dashed red lines in Figure 1.2 indicate the prescribed constraints on the satellite position and fuel. In this example, the satellite thrusters use all available fuel in the beginning, which is followed by a long coasting phase. Constraints are violated for the first time after about seven days (see lower right corner in left plot of Figure 1.2) and the satellite reenters the prescribed position window before constraints are violated again.

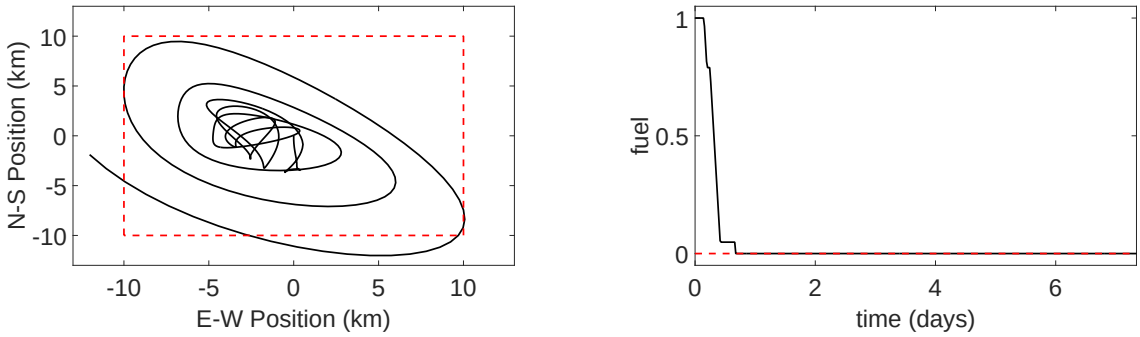

Figure 1.2: DCOC application: GEO satellite station keeping. Left: satellite position relative to target orbit in N-S / E-W plane. Right: normalized fuel mass over time. The dashed red lines indicate the constraints.

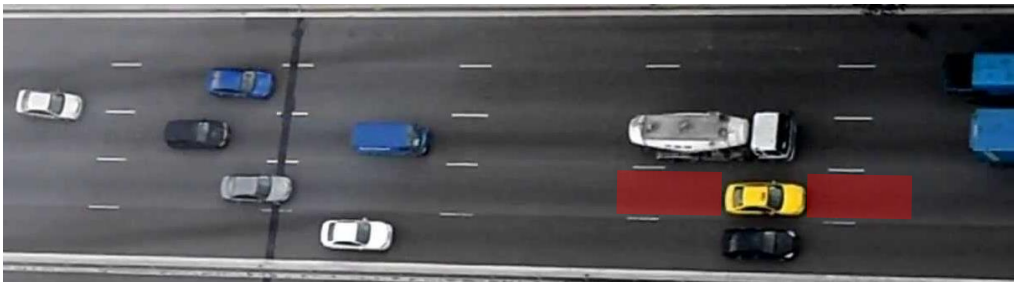

Figure 1.3: DCOC application: driving policy that maximizes the average time that other cars stay outside the red area.

Another area of DCOC applications is automated and autonomous driving. Automated and autonomous vehicles may improve road safety, provide greater convenience for humans, and lower cost of transportation in the future. Car following or adaptive cruise control (ACC) can be formulated as a DCOC problem, where the objective is to control the

acceleration of the follower vehicle such that the distance to the lead vehicle (with the lead vehicle velocity modeled as a random disturbance) stays within a prescribed range for as long as possible. This car following problem can be extended by taking into account lateral motion as well, i.e., allowing lane changes. In this case, the objective of DCOC may be to generate a driving policy that maximizes the expected time that none of the surrounding cars (treated as random disturbances) enters a prescribed safe zone around the ego car as illustrated in Figure 1.3.

Besides satellite station keeping and autonomous driving, other DCOC applications are, for example:

- Energy management of hybrid electric vehicles: there are two or more distinct power sources, e.g., an engine and an electric motor with a battery. The objective is to control the power flows in this system in order to maximize the vehicle's range. The problem may be formulated as a stochastic control problem where, e.g., the wheel power demand is treated as a random disturbance.

- Spacecraft attitude control: the spacecraft orientation is subject to drift caused, e.g., by SRP, and the objective is to generate counteracting torques, e.g., using reaction wheels or thrusters, such that the orientation stays within prescribed bounds for as long as possible. The control authority may be limited due to underactuation or a saturated reaction wheel (RW).

- Glider flight management: consider the flight of a glider airplane with uncertain lift conditions, where the objective is to find an optimal flight path that maximizes the expected time of flight (time in air). An additional constraint may be imposed requiring the glider to stay within a prescribed range.

- Hover control of an unmanned aerial vehicle (UAV): given random wind disturbances, the DCOC problem is formulated with the objective of satisfying tight constraints on the UAV position for as long as possible. Similarly, the objective may be to maximize the expected time of following a moving target (which is an additional random disturbance), i.e., find a control strategy such that the UAV stays within a certain range of the target for as long as possible.

- Dynamic positioning of offshore vessels: find a control strategy that maintains the vessel position and heading within prescribed bounds for as long as possible by using its propellers and thrusters, where environmental forces (wind, current, waves, etc.) act as random disturbances.

- Cancer treatment: find a treatment strategy (e.g., medication dosage and intervals) that optimally counteracts cancer growth in order to maximize the expected lifetime of a cancer patient.

1.3 Literature Review

The DCOC problem investigated in this dissertation is an optimal exit-time control problem. Similar problems were studied in the past for both deterministic and stochastic systems. The work by Lions [1] investigates optimal exit-time control problems in continuous-time, where it is shown that, under suitable assumptions, the optimal control and its corresponding value function satisfy the Hamilton-Jacobi-Bellman (HJB) equation in the viscosity/weak sense. The notion of viscosity solutions to the HJB equation, which is a second-order partial differential equation (PDE) in the stochastic case, is introduced in [2, 3]. Viscosity solutions to the HJB equation and properties of the value function of optimal exit-time control problems for continuous-time stochastic systems are also discussed in the book by Fleming and Soner [4] as well as in [5–10].

Similar to the stochastic case, viscosity solutions to the HJB equation and properties of the value function in the deterministic case (in which the HJB equation is a first-order PDE) are discussed in the book by Bardi and Capuzzo-Dolcetta [11] as well as in [12–21]. A different approach can be found in [22], where Lipschitz continuity and semi-concavity results for the value function are obtained based on the maximum principle. Moreover, Rungger and Stursberg [23] investigate the continuity of the value function for hybrid systems with continuous-time dynamics.

However, most of the previous results on optimal exit-time control problems (both in the deterministic and stochastic case) for continuous-time systems rely on conservative assumptions, in particular: non-negative cost values (i.e., minimization of non-negative cost functions) and/or cost functions with discount factors. In contrast, DCOC involves maximization problems with undiscounted non-negative yield functions such as the the problems envisioned in the previous section (Section 1.2). For continuous-time systems with Wiener-Poisson inputs, a stochastic optimal control problem similar to the DCOC problem is investigated by Kolmanovsky and Maizenberg [24], who obtain an explicit solution to the HJB equation only for a simple scalar system. Similarly, in [25] a game-theoretic approach is used to obtain an explicit expression for the value function of a (continuous-time) flow control time maximization problem with discount factor.

Explicit solutions to the HJB equation, however, only exist for special problems. Otherwise, a solution can only be obtained approximately using numerical methods. The numer-

ical methods developed by Kushner et al. [26, 27] and their extensions, for example, [28], are based on iteratively solving discretized versions of the problem using the value iteration (VI) algorithm [29]. Under certain conditions, the sequence of solutions to the discrete problems approaches the solution of the continuous problem [27, 30]. Another numerical approach is to represent the value function of the discretized problem as a tensor and approximate the solution to the HJB equation by employing, for example, the alternating least squares algorithm [31, 32] or tensor-based VI [33]. Raffard et al. [34] propose an adjoint-based method to solve exit-time problems for hybrid systems with continuous-time dynamics.

Instead of solving a PDE numerically, the formulation of optimal exit-time control problems in discrete-time appears to be computationally more tractable for determining a solution [29, 35]. Kolmanovsky et al. [35] consider the DCOC problem for stochastic discrete-time systems and show that VI converges to the value function if an optimal solution exists. They also propose a numerical implementation of VI based on equidistant state space discretization, a typical approach used in applications [36], and solve two DCOC examples of hybrid electric vehicle powertrain management and oil extraction control. The approach by Kolmanovsky et al. is furthermore used to solve other stochastic DCOC problems of hybrid electric vehicle energy management and adaptive cruise control [37], energy management for a fuel cell and battery powered mini air vehicle [38], and glider flight management [39, 40].

1.4 Contributions and Dissertation Outline

The contributions of this dissertation are advancements in theory, methodology, and applications of DCOC for systems with discrete-time dynamics, extending the initial discrete-time DCOC framework proposed by Kolmanovsky et al. [35]. The contributions can be categorized into two parts: DCOC for deterministic systems (first part) and DCOC for stochastic systems (second part). In particular, new approaches based on dynamic programming (DP), approximate/adaptive dynamic programming (ADP), and model predictive control (MPC) as well as approaches to obtain open-loop solutions are developed and analyzed, and several new DCOC applications are proposed and investigated. The majority of the contents of this dissertation has been originally published or submitted to scientific journals [41, 42] or conference proceedings [43–51]. The individual contributions of this dissertation and their corresponding chapters for the first part (deterministic systems) are:

Chapter 2 (DP approaches):

- Section 2.2 [41]: using DP techniques, properties of deterministic DCOC problems are derived. Under suitable assumptions, the objective function and the first exit-time [see (2.2)] are shown to be upper semi-continuous (USC). Furthermore, conditions are derived under which a control policy is optimal and additional conditions are provided that guarantee the existence of a solution to the DCOC problem.

- Section 2.3 [41, 43]: an enhanced version of the VI algorithm is developed that obtains the value function faster than conventional VI (which is a special case of the enhanced VI algorithm) in a numerical setting. The idea behind the enhanced algorithm is to consider VI as a control problem with the objective of driving the error in the DP equation (also referred to as the Bellman equation) to zero. The proposed algorithm updates the value function in proportion to the error and an extension using adaptive proportional gains is considered.

- Section 2.4 [41]: based on proportional feedback VI (Section 2.3), an ADP method is proposed. The ADP method uses a function approximator such as a neural network (NN) or Gaussian process to approximate the value function. Compared to conventional DP techniques that approximate the value function on a mesh of discrete points (and using interpolation to approximate the value function between the grid points), the ADP method mitigates the curse of dimensionality and is computationally advantageous for higher-dimensional problems.

- Section 2.5 [44, 49]: a new algorithm, referred to as base-trajectory VI, is developed. The new algorithm converges to the value function by gradually connecting pieces of an optimal control policy. This is achieved by traversing a base-trajectory until deviating from it provides an improvement. It is shown that, in a numerical setting, base-trajectory VI is more accurate (i.e., achieves better performance in terms of longer exit-times) than conventional VI.

- Section 2.6 [41, 43–45, 49]: using the methods developed in Sections 2.3–2.5, several numerical case studies are presented for the following DCOC problems: time maximization of a van der Pol (VDP) oscillator (Sections 2.6.2.1 and 2.6.3.1), life extension of a low Earth orbit (LEO) satellite (Sections 2.6.1 and 2.6.2.2) as well as of a GEO satellite (Section 2.6.3.2), and spacecraft attitude control (Section 2.6.4).

Chapter 3 (open-loop solutions and MPC approach):

- Section 3.2 [42, 48]: for deterministic time maximization problems (i.e., maximize the time before prescribed constraints are violated), a mixed-integer linear program (MILP) is developed that obtains an open-loop solution for problems subject to linear systems. The MILP is relaxed, yielding a standard linear program (LP) that generates good-quality suboptimal solutions. In addition, an iterative procedure is proposed to obtain a proper time horizon for the LP and MILP, respectively. For problems subject to nonlinear systems, a similar mixed-integer nonlinear program (MINLP) is developed, including a nonlinear program (NLP) as its relaxed version.

- Section 3.3 [42,45,48]: the LP and MILP (Section 3.2) are used to formulate an MPC scheme, where, at each time instant, the LP or MILP is solved over a receding time horizon based on the current state of the system. The first element of the obtained open-loop control sequence is applied to the system, providing state feedback in order to compensate for unmodeled effects.

- Section 3.4 [42, 45, 48]: using the developed MPC scheme, several numerical case studies of drift counteraction for a VDP oscillator (Section 3.4.1.1), GEO satellite station keeping (Section 3.4.2.1), and spacecraft attitude control (Section 3.4.2.2) are treated. The results are compared against the respective open-loop solutions.

In the second part of this dissertation, the developments for deterministic systems are extended to stochastic systems. The individual contributions for the second part (stochastic systems) are listed as follows:

Chapter 4 (DP approaches):

- Section 4.2: conditions are obtained under which the expected value of the first exit-time (first time instant of constraint violation) and the value function are bounded from above.

- Section 4.3 [46]: based on Section 4.2, the results on upper semi-continuity of the first exit-time and objective function of deterministic DCOC problems (Section 2.2) are extended to the stochastic case. Moreover, conditions are derived that characterize an optimal control policy and the existence of a solution is analyzed.

- Section 4.4 [46]: proportional feedback VI for deterministic DCOC problems (Section 2.3) is extended to the stochastic case. The proof of convergence requires a

procedure different from the deterministic case. The algorithm is extended using adaptive proportional gains and a damping factor is introduced to prevent the iterations from diverging in the adaptive setting.

- Section 4.5 [50]: the DCOC framework is used to generate driving policies for autonomous vehicles. The traffic around the ego car is described by a stochastic hybrid model, which is developed in Section 4.5.1. In Section 4.5.2, the DCOC framework is extended to take into account such hybrid models. An ADP method similar to the method for deterministic problems (Section 2.4) is presented in Section 4.5.3, where NNs are used to approximate the value function. A numerical case study is considered where the ADP method is compared against a conventional DP approach (Section 4.5.4).

- Section 4.6 [46]: other numerical case studies of stochastic DCOC problems are treated, including control of a pendulum (Section 4.6.1), glider flight management (Section 4.6.2), and a car following problem (Section 4.6.3).

Chapter 5 (Tree-based SMPC approach):

- Section 5.2 [51]: a stochastic model predictive control (SMPC) approach is developed in this chapter to solve stochastic DCOC problems with the objective of maximizing the expected time before constraint violation. In order to optimize over a subset of all possible scenarios, an algorithm is proposed to construct a scenario tree that encodes the most likely system behavior for a specified number of tree nodes.

- Section 5.3 [51]: an MILP is proposed, the solution of which, assuming a linear system model with additive random disturbances modeled by Markov chains, approaches a solution to the stochastic DCOC problem as the number of tree nodes approaches infinity. In contrast to the open-loop solutions provided by the MILP and LP in the deterministic case (Section 3.2), the MILP in the stochastic case takes into account state feedback at future time instants.

- Section 5.4 [51]: the MILP from Section 5.3 is used to implement the tree-based SMPC strategy. At each time instant, based on the current states and disturbances, a new scenario tree is constructed with a specified number of nodes (Section 5.2) and the corresponding MILP solution is computed. The first element of the solution, i.e., the control associated with the root node of the scenario tree, is applied to the system. This procedure is repeated at the next time instant, introducing feedback to compensate for unmodeled effects as well as for not taking into account all possible

scenarios. If computing the MILP solution requires longer than available time, a relaxed version without integer variables, which is a standard LP, is solved instead.

- Section 5.5 [51]: the developed tree-based SMPC scheme is applied to several numerical case studies, including applications to ACC and driving policies for autonomous cars, where the results of the SMPC scheme are compared to DP-based results (Section 4.6). For the autonomous driving problem in Section 5.5.3, a special version of the MILP is provided to account for potential lane changes. In addition, an effective hybrid SMPC strategy is derived for drift counteraction by combining DCOC-based lane change decision making with a DCOC-based car following controller.

In addition to DCOC, other developments on related topics of optimal control for systems with disturbances have appeared in a journal publication [52] and are presented in Chapter 6. In particular, a closed-form approximate solution to a linear quadratic (LQ) optimal control problem with known time-varying disturbance term is developed in Chapter 6. The approach is demonstrated in Section 6.4 for a spacecraft orbital maneuver taking into account J_2 and J_3 perturbation as well as atmospheric drag perturbation.

CHAPTER 2

Deterministic DCOC – DP Approaches

2.1 Problem Formulation

In this chapter, discrete-time nonlinear systems of the form

$$x_{t+1} = f(x_t, u_t), \tag{2.1}$$

are considered, where $x_t \in \mathbb{R}^n$ denotes the state vector at a discrete time instant $t \in \mathbb{Z}_{\geq 0}$ and f is a general nonlinear function (no restrictions on f are made at this point). The control input vector at a time instant t is given by $u_t = \pi(x_t)$, where $\pi : \mathbb{R}^n \to U \subset \mathbb{R}^p$ is an admissible control policy and the set of admissible control policies is denoted by Π. The first exit-time from a prescribed set $G \subset \mathbb{R}^n$, given the initial state vector $x_0 \in G$ and control policy $\pi \in \Pi$, is defined as follows

$$\tau(x_0, \pi) = \inf \left\{ t \in \mathbb{Z}_{\geq 0} : x_t \notin G \right\}, \tag{2.2}$$

where x_t evolves according to (2.1). The DCOC problem is given by

$$J(x_0, \pi) = \sum_{t=0}^{\tau(x_0,\pi)-1} g(x_t, u_t) \to \max_{\pi \in \Pi}, \tag{2.3}$$

where $x_0 \in G$ and $g : G \times U \to \mathbb{R}_+$ is the instantaneous yield. Note that if $g \equiv 1$ in (2.3), the objective is to maximize the first exit-time from G.

2.2 Theoretical Results

The approach in this chapter for solving the DCOC problem (2.3) is based on DP, where the optimal control policy is characterized by the value function V, which is defined by

$$V(x) = \sup_{\pi \in \Pi} J(x, \pi). \tag{2.4}$$

The following assumption about g is made.

Assumption 2.1. There exists a real-valued $\bar{g} > 0$ such that $g(x, u) \leq \bar{g}$ for all $(x, u) \in G \times U$.

Theorem 2.1 provides conditions under which the total yield and the value function are bounded. It is based on the following assumption about the first exit-time $\tau(x, \pi)$.

Assumption 2.2. There exists an integer $\bar{T} > 0$ such that $\tau(x, \pi) \leq \bar{T}$ for all $x \in G$ and $\pi \in \Pi$.

This assumption is reasonable in DCOC problems in which every trajectory will eventually violate the constraints and the objective is either to delay this event or to maximize yield before it happens. This is the case, for example, in applications where resources such as fuel are limited, see Section 1.2, or where insufficient control authority is available.

Theorem 2.1. *Suppose Assumptions 2.1 and 2.2 hold. Then there exists $\bar{V} > 0$ such that $J(x, \pi) \leq V(x) \leq \bar{V}$ for all $x \in G$ and $\pi \in \Pi$.*

Proof. Let $x = x_0 \in G$ be a given state and $\pi \in \Pi$. Using Assumptions 2.1 and 2.2,

$$J(x, \pi) = \sum_{t=0}^{\tau(x,\pi)-1} g(x_t, u_t) \leq \sum_{t=0}^{\tau(x,\pi)-1} \bar{g} \leq \bar{T}\bar{g}. \tag{2.5}$$

This and (2.4) imply that $V(x) \leq \bar{V} = \bar{T}\bar{g}$. □

Remark 2.1. *Theorem 2.1 guarantees the existence of a maximizing sequence for all $x \in G$, i.e., a sequence $\{\pi_n\}$ in Π such that $J(x, \pi_n) \to \sup_{\pi \in \Pi} J(x, \pi)$ or, equivalently, $J(x, \pi_n) \to V(x)$ for all $x \in G$.*

The next theorem provides sufficient conditions for a control policy to be optimal.

Theorem 2.2. *Let $L^{\pi}V(x) = V(x) - V(f(x,\pi(x)))$ and suppose Assumptions 2.1 and 2.2 hold. Then $\pi^* \in \Pi$ satisfies*

$$\begin{aligned} &L^{\pi^*}V(x) = g(x,\pi^*(x)), \ \text{ if } x \in G, \\ &L^{\pi}V(x) \geq g(x,\pi(x)), \ \text{ if } x \in G, \ \pi \neq \pi^*, \\ &V(x) = 0, \ \text{ if } x \notin G, \end{aligned} \tag{2.6}$$

for all $x \in \mathbb{R}^n$ and $\pi \in \Pi$ if and only if (iff) π^* maximizes $J(x,\pi)$ for all $x \in G$. Furthermore, $V(x) = J(x,\pi^*)$ and

$$\pi^*(x) \in \Pi^*(x) = \arg\max_{u \in U} \left\{ g(x,u) + V(f(x,u)) \right\}. \tag{2.7}$$

Proof. Since $J(x,\pi) = 0$ for all $x \notin G$, $V(x) = 0$ for all $x \notin G$. Now let $x = x_0 \in G$ be a given state and $\pi \in \Pi$. For the first part of the proof, assume π^* satisfies (2.6). Thus,

$$\begin{aligned} J(x,\pi) &= \sum_{t=0}^{\tau(x,\pi)-1} g(x_t, \pi(x_t)) \\ &\leq \sum_{t=0}^{\tau(x,\pi)-1} L^{\pi}V(x_t) \\ &= V(x), \end{aligned} \tag{2.8}$$

since $V(x_{\tau(x,\pi)}) = 0$ due to $x_{\tau(x,\pi)} \notin G$. Similarly,

$$\begin{aligned} J(x,\pi^*) &= \sum_{t=0}^{\tau(x,\pi^*)-1} g(x_t, \pi^*(x_t)) \\ &= \sum_{t=0}^{\tau(x,\pi^*)-1} L^{\pi^*}V(x_t) \\ &= V(x). \end{aligned} \tag{2.9}$$

Equations (2.8) and (2.9) can be compared because V is bounded by Theorem 2.1, which shows that $J(x,\pi^*) \geq J(x,\pi)$. It immediately follows from (2.6) that $\pi^*(x) \in \Pi^*(x)$ according to (2.7). For the second part of the proof, assume that π^* maximizes $J(x,\pi)$ for all $x \in G$. Then, by (2.4), $V(x) = J(x,\pi^*)$ for all $x \in G$. This implies

$$\begin{aligned} V(x) &= g(x,\pi^*(x)) + J(f(x,\pi^*(x)),\pi^*) \\ &= g(x,\pi^*(x)) + V(f(x,\pi^*(x))). \end{aligned} \tag{2.10}$$

Since $V(x)$ is the optimal value, it follows that, for any admissible policy $\pi \neq \pi^*$,

$$\begin{aligned} V(x) &\geq g(x, \pi(x)) + J(f(x, \pi(x)), \pi^*) \\ &= g(x, \pi(x)) + V(f(x, \pi(x))). \end{aligned} \tag{2.11}$$

□

Remark 2.2. *The optimal control policy π^* to* (2.3)*, if it exists, may not be unique (see, for example, Figure 1.1). In case of non-uniqueness, additional criteria, for instance, minimizing the 2-norm, may be used for selecting the control from the set of maximizers in* (2.7)*.*

Theorem 2.3. *If a solution to* (2.3) *exists, V is the unique solution to* (2.6)*.*

Proof. Suppose $\pi^* \in \Pi$ is a solution to (2.3). Furthermore, suppose that, in addition to V, another function $\hat{V}$ satisfies (2.6). It follows from the proof of Theorem 2.2 and (2.9) that, for all $x \in G$, $V(x) = J(x, \pi^*)$ and $\hat{V}(x) = J(x, \pi^*)$, which implies $\hat{V} = V$. □

The existence of a solution to (2.3) can be studied using the set $\Pi^*(x)$.

Theorem 2.4. *A solution $\pi^* \in \Pi$ to the DCOC problem* (2.3) *exists for all $x \in G$ iff the set $\Pi^*(x)$ defined in* (2.7) *is nonempty for all $x \in G$.*

Proof. For the first part of the proof, assume that $\Pi^*(x)$ is nonempty for all $x \in G$. Then there exists $\pi^*(x) = u^* \in U$ such that

$$g(x, u^*) + V(f(x, u^*)) \geq g(x, u) + V(f(x, u)), \tag{2.12}$$

for all $x \in G$ and $u \in U$. Therefore, according to Theorem 2.2, π^* is a solution to (2.3). For the second part of the proof, assume that a solution π^* exists for all $x \in G$. This and (2.4) imply that $V(x) = J(x, \pi^*)$ for all $x \in G$. Consequently, by denoting $u^* = \pi^*(x) \in U$,

$$\begin{aligned} J(x, \pi^*) &= g(x, u^*) + J(f(x, u^*), \pi^*) \\ &\geq g(x, u) + J(f(x, u), \pi^*), \end{aligned} \tag{2.13}$$

for all $x \in G$ and $u \in U$. Using $V(x) = J(x, \pi^*)$, (2.13) implies that, for all $x \in G$, there exists $u^* = u^*(x) \in U$ such that $g(x, u^*) + V(f(x, u^*)) \geq g(x, u) + V(f(x, u))$ for all $u \in U$. □

Using Theorem 2.4, in order to guarantee the existence of a solution to the DCOC problem, conditions under which $\Pi^*(x)$ is nonempty need to be found. Three separate

conditions under which this holds are provided in Theorem 2.5, where condition 2 relies on Lemma 2.1. The following definition of upper semi-continuity adapted from [53] is used.

Definition 2.1. A real-valued function $f(x)$ is USC at $x \in G$ if, for all $\varepsilon > 0$, there exists $\delta > 0$ such that $y \in G$ and $\|x - y\| < \delta$ imply $f(y) < f(x) + \varepsilon$. A real-valued function $f(x)$ is USC on G (i.e., with respect to $x \in G$) if it is USC at all $x \in G$.

Lemma 2.1. *If $V(x)$ is USC on G and $f(x, u)$ is continuous with respect to $u \in U$ for all $x \in G$, then $V(f(x, u))$ is USC with respect to $u \in U$ for all $x \in G$.*

Proof. Define $h^x(u) = V(f(x, u))$. It needs to be shown that, for every $\varepsilon > 0$, there exists $\delta > 0$ such that $v, u \in U$ and $\|v - u\| < \delta$ imply $h^x(v) < h^x(u) + \varepsilon$ for all $x \in G$, see Definition 2.1. Since $f(x, u)$ is continuous with respect to $u \in U$ for all $x \in G$, for every $\varepsilon_1 > 0$, there exists $\delta_1 > 0$ such that

$$\|v - u\| < \delta_1 \Rightarrow \|f(x, v) - f(x, u)\| < \varepsilon_1, \tag{2.14}$$

for all $x \in G$ [54]. Moreover, since V is USC, for any $\varepsilon_2 > 0$, there exists $\delta_2 > 0$ such that, for all $x \in G$,

$$\|y - f(x, u)\| < \delta_2 \Rightarrow V(y) < V(f(x, u)) + \varepsilon_2. \tag{2.15}$$

For $\varepsilon > 0$, using (2.15) with $y = f(x, v)$, there exists $\delta_2 > 0$ such that

$$\|f(x, v) - f(x, u)\| < \delta_2 \Rightarrow h^x(v) < h^x(u) + \varepsilon.$$

Thus, by taking $\varepsilon_1 = \delta_2$ and $\delta_1 = \delta > 0$ in (2.14), $\|v - u\| < \delta$ implies $h^x(v) < h^x(u) + \varepsilon$. □

Theorem 2.5. *Suppose either*

1. *U is finite and Assumptions 2.1 and 2.2 hold.*
2. *U is compact, $f(x, u)$ and $g(x, u)$ are continuous and USC with respect to $u \in U$, respectively, for all $x \in G$, and $V(x)$ is USC on G.*
3. *Assumption 2.2 holds and $g \equiv 1$.*

Then a solution to the DCOC problem (2.3) *exists for all $x \in G$.*

Proof. It needs to be shown that $\Pi^*(x)$ in (2.7) is nonempty for all $x \in G$ since, due to Theorem 2.4, this implies the existence of a solution to (2.3). Assume that 1 holds. By

Assumptions 2.1 and 2.2 and Theorem 2.1, both V and g are bounded for all $x \in G$ and $u \in U$. Consequently, their sum is bounded. Since U is finite and the maximum of a bounded function over a finite set exists, $\Pi^*(x)$ is nonempty for all $x \in G$. Now suppose 2 holds. By Lemma 2.1, $V(f(x,u))$ is USC with respect to $u \in U$ for all $x \in G$. Since the sum of two USC functions is USC, the sum of g and V in (2.7) is USC with respect to $u \in U$ for all $x \in G$. Because U is compact, it follows from the extension of the Weierstrass theorem to USC functions [55] that $\Pi^*(x)$ is nonempty for all $x \in G$. Finally, suppose that 3 holds. Since $g \equiv 1$, the objective function is integer-valued and bounded (by Assumption 2.2). Hence, according to (2.4), $V(x)$ is integer-valued and bounded for all $x \in G$. Because any bounded collection of integers has a maximum, $\Pi^*(x)$ is nonempty for all $x \in G$. □

In the following, conditions are derived under which the first exit-time τ and the objective function J are USC.

Theorem 2.6. *Suppose Assumption 2.2 holds, G is compact, and $f(x,u)$ is continuous on $G \times U$. Then $\tau(x,\pi)$ is USC with respect to $x \in G$ for all $\pi \in C_G(\Pi)$, where*

$$C_G(\Pi) = \{\pi \in \Pi | \pi \textit{ is continuous on } G\} \tag{2.16}$$

is the set of admissible control policies that are continuous on G.

Proof. Let $\pi \in C_G(\Pi)$ be a given admissible control policy and $x = x_0 \in G$ be a given state. Denote the trajectory that results from the control policy π by $\{x_t\}_\pi$. Consider another trajectory $\{\tilde{x}_t\}_\pi$ with $\tilde{x}_0 \in G$ that results from π. Using Definition 4.5 in [54], the continuity of f, G being compact, and $\pi \in C_G(\Pi)$ imply that for all $\tilde{\varepsilon} > 0$, there exists $\delta > 0$ such that

$$\|\tilde{x}_0 - x_0\| < \delta \;\Rightarrow\; \left\|\tilde{x}_{\tau(x_0,\pi)} - x_{\tau(x_0,\pi)}\right\| < \tilde{\varepsilon}, \tag{2.17}$$

where $\tau(x_0,\pi)$ is defined due to Assumption 2.2. Assumption 2.2 and the compactness of G imply that there exists $\varepsilon_G > 0$ such that

$$\left\|\tilde{x}_{\tau(x_0,\pi)} - x_{\tau(x_0,\pi)}\right\| < \varepsilon_G \;\Rightarrow\; \tilde{x}_{\tau(x_0,\pi)} \notin G, \tag{2.18}$$

where $x_{\tau(x_0,\pi)} \notin G$ by the definition of τ. Take $\tilde{\varepsilon} = \varepsilon_G$ in (2.17) and note that $\tilde{x}_{\tau(x_0,\pi)} \notin G$ implies $\tau(\tilde{x}_0,\pi) \leq \tau(x_0,\pi)$. It then follows from (2.17) and (2.18) that for any $\varepsilon > 0$, there exists $\delta > 0$ such that $\|\tilde{x}_0 - x_0\| < \delta \;\Rightarrow\; \tau(\tilde{x}_0,\pi) < \tau(x_0,\pi) + \varepsilon$, i.e., $\tau(x,\pi)$ is USC with respect to $x \in G$ for all $\pi \in C_G(\Pi)$. □

Note that, in contrast to the results by Lions [1], who proves upper semi-continuity of the first exit-time of continuous-time systems under the assumption of open-loop control sequences, Theorem 2.6 assumes continuous state feedback control policies. It is straightforward to show that the result by Lions (continuous-time systems with open-loop control) also holds for discrete-time systems. In this regard, let $\tau(x_0, \{u_t\})$ be the first exit-time of x_0 when using the open-loop control sequence $\{u_t\}$. Since f is continuous with respect to $u \in U$ and G is compact, by analogy to (2.17), it follows that, for all $\tilde{\varepsilon} > 0$, there exists $\delta > 0$ such that

$$\|\tilde{x}_0 - x_0\| < \delta \;\Rightarrow\; \left\|\tilde{x}_{\tau(x_0,\{u_t\})} - x_{\tau(x_0,\{u_t\})}\right\| < \tilde{\varepsilon}, \tag{2.19}$$

for every admissible open-loop control sequence $\{u_t\}$. Following the arguments of the proof of Theorem 2.6, one obtains that there exists $\varepsilon_G > 0$ such that

$$\left\|\tilde{x}_{\tau(x_0,\{u_t\})} - x_{\tau(x_0,\{u_t\})}\right\| < \varepsilon_G \;\Rightarrow\; \tilde{x}_{\tau(x_0,\{u_t\})} \notin G. \tag{2.20}$$

Eventually, it follows that, for any $\varepsilon > 0$, there exists $\delta > 0$ such that

$$\|\tilde{x}_0 - x_0\| < \delta \;\Rightarrow\; \tau(\tilde{x}_0, \{u_t\}) < \tau(x_0, \{u_t\}) + \varepsilon,$$

i.e., the first exit-time is also USC under the assumption of open-loop control sequences.

The next theorem extends the results on the first exit-time to the objective function J.

Theorem 2.7. *Suppose Assumption 2.2 holds and G is compact. Furthermore, suppose that $f(x,u)$ and $g(x,u)$ are continuous and USC on $G \times U$, respectively. Then $J(x,\pi)$ is USC with respect to $x \in G$ for all $\pi \in C_G(\Pi)$.*

Proof. Let $\pi \in C_G(\Pi)$ be a given admissible control policy, $x = x_0 \in G$ be a given initial state, and $\{x_t\}_\pi$ be the corresponding trajectory. Moreover, let $\{\tilde{x}_t\}_\pi$ be another trajectory with initial state $\tilde{x}_0 \in G$. Since $\tau(x,\pi)$ is integer-valued and USC with respect to $x \in G$ for all $\pi \in C_G(\Pi)$ (by Theorem 2.6), there exists $\delta_1 > 0$ such that

$$\|\tilde{x}_0 - x_0\| < \delta_1 \;\Rightarrow\; \tau(\tilde{x}_0, \pi) \leq \tau(x_0, \pi). \tag{2.21}$$

In addition, for any $\varepsilon > 0$, the continuity of f and the upper semi-continuity of g imply that there exists $\delta_2 > 0$ such that

$$\|\tilde{x}_0 - x_0\| < \delta_2 \Rightarrow \sum_{t=0}^{\tau(\tilde{x}_0,\pi)-1} g(\tilde{x}_t, \tilde{u}_t) < \sum_{t=0}^{\tau(\tilde{x}_0,\pi)-1} \left(g(x_t, u_t) + \frac{\varepsilon}{\tau(\tilde{x}_0, \pi)} \right), \tag{2.22}$$

where $\tilde{u}_t = \pi(\tilde{x}_t)$ and $u_t = \pi(x_t)$. Take $\delta = \delta_1 = \delta_2$ sufficiently small such that (2.21) and (2.22) hold. Then,

$$\begin{aligned} J(x_0, \pi) &= \sum_{t=0}^{\tau(x_0,\pi)-1} g(x_t, u_t) \\ &= \sum_{t=0}^{\tau(\tilde{x}_0,\pi)-1} (g(x_t, u_t) - g(\tilde{x}_t, \tilde{u}_t)) + J(\tilde{x}_0, \pi) \\ &+ \sum_{t=\tau(\tilde{x}_0,\pi)}^{\tau(x_0,\pi)-1} g(x_t, u_t) > -\varepsilon + J(\tilde{x}_0, \pi), \end{aligned} \tag{2.23}$$

since g is positive. Consequently, $J(\tilde{x}_0, \pi) < J(x_0, \pi) + \varepsilon$, which proves upper semi-continuity of $J(x, \pi)$ with respect to $x \in G$ for all $\pi \in C_G(\Pi)$. □

Upper semi-continuity of the value function V, however, cannot be inferred from Theorem 2.7 since the supremum of infinitely many USC functions, see (2.4), may not be USC. On the other hand, if $g \equiv 1$ and Assumption 2.2 holds, then $V(x) = \tau(x, \pi^*)$, where π^* is a solution to the DCOC problem, which exists by Theorem 2.5. Hence, assuming continuous control policies, i.e., $\pi \in C_G(\Pi)$, $V(x)$ is USC with respect to $x \in G$ in this case (by Theorem 2.6).

2.3 Proportional Feedback VI

2.3.1 Theoretical Results

According to (2.7), an optimal control policy π^* is defined by the value function V, which may be computed using conventional VI [35]. In the following, a modification of the VI algorithm is developed that may provide faster convergence in a numerical setting. In this regard, based on Theorem 2.2, the error at iteration n is defined as follows

$$e_n(x) = \max_{u \in U} \{V_n(f(x, u)) + g(x, u)\} - V_n(x), \tag{2.24}$$

where $e_n = 0$ iff $V_n = V$. Then $V_n(x)$ is updated in proportion to $e_n(x)$ according to

$$\begin{aligned} &V_{n+1}(x) = V_n(x) + k e_n(x), \text{ if } x \in G, \\ &V_{n+1}(x) = 0, \text{ if } x \notin G, \end{aligned} \tag{2.25}$$

with $k \in \mathbb{R}$ as the proportional gain. Note that the conventional VI algorithm is a special case of (2.24) and (2.25) for $k = 1$. The sequence of functions in (2.24) and (2.25) is referred to as proportional feedback VI. It converges to V under the conditions stated in Theorem 2.8. In order to prove Theorem 2.8, the sets

$$\mathcal{H} = \{x \in G : \exists u \in U \text{ s.t. } f(x,u) \in G\}, \tag{2.26}$$

$$\mathcal{K}_0 = G \cap \mathcal{H}^{\mathrm{c}}, \tag{2.27}$$

$$\mathcal{K}_m = \left\{ x \in \mathcal{H} : f(x,u) \in \bigcup_{k=0}^{m-1} \mathcal{K}_k \cup G^{\mathrm{c}}, \forall u \in U \right\}, \tag{2.28}$$

are defined, where $\mathcal{K}_m$, $m \in \mathbb{Z}_{\geq 0}$, are the sets of states that lead to trajectories exiting G in at most $m+1$ steps and $\mathcal{S}^{\mathrm{c}}$ is the complement of a set $\mathcal{S}$. Moreover, Lemma 2.2 is used for the proof of Theorem 2.8.

Lemma 2.2. *Suppose Assumption 2.2 holds. Then there exists an integer $\bar{m} \geq 0$ such that the sets $\mathcal{K}_m$ are nonempty for all $m \leq \bar{m}$ and $G = \bigcup_{m=0}^{\bar{m}} \mathcal{K}_m$.*

Proof. Let $\bar{m} \in \mathbb{Z}_{\geq 0}$. By Assumption 2.2, there exists $x^* \in G$ and $\pi^* \in \Pi$ such that $\bar{m} + 1 = \tau(x^*, \pi^*) \geq \tau(x, \pi)$ for all $x \in G$ and $\pi \in \Pi$. Denote the trajectory corresponding to x^* and π^* by $\{x_t^*\}_{\pi^*}$, where $x_0^* = x^*$. Then $x_t^* \in \mathcal{K}_{\bar{m}-t}$ for $t \in \{0, 1, ..., \bar{m}\}$. Furthermore, by Assumption 2.2, for each $x \in G$, there exists an integer $\tilde{m}(x) \leq \bar{m}$ such that $x \in \mathcal{K}_{\tilde{m}(x)}$. Consequently, $G = \bigcup_{m=0}^{\bar{m}} \mathcal{K}_m$. □

Theorem 2.8. *Suppose there exists a solution π^* to* (2.3) *for all $x \in G$. Furthermore, suppose $k \in (0, 2)$ and let $V_0(x) \in \mathbb{R}$ be defined for all $x \in G$. Then the sequence of functions defined by* (2.24) *and* (2.25) *converges pointwise to $V(x)$ for all $x \in G$.*

Proof. The existence of a solution to (2.3) for all $x \in G$ implies Assumptions 2.1 and 2.2. Thus, by Lemma 2.2, there exists an integer $\bar{m} \geq 0$ such that the sets $\mathcal{K}_m$ defined by (2.27) and (2.28) are nonempty and each $x \in G$ can be assigned to one of the sets $\mathcal{K}_m$ for $m \leq \bar{m}$. Then proceed by induction. For all $x \in \mathcal{K}_0$, the existence of an optimal control policy π^* implies that for $n \geq 1$,

$$e_n(x) = g(x, \pi^*(x)) - V_n(x) = V(x) - V_n(x), \tag{2.29}$$

because $V_n(f(x,u)) = 0$ according to (2.25) due to $f(x,u) \notin G$ for all $u \in U$. Hence, by (2.25), $V_{n+1}(x) = V_n(x) + k[V(x) - V_n(x)]$, which can be written as

$$V_{n+1}(x) - V(x) = [1-k][V_n(x) - V(x)]. \tag{2.30}$$

This and $k \in (0,2)$ imply that $V_n(x) \to V(x)$ for all $x \in \mathcal{K}_0$. Now assume that for some $m < \bar{m}$, $V_n(x) \to V(x)$ for all $x \in \mathcal{K}_\zeta$ and $\zeta \in \mathcal{C} = \{0,1,...,m\}$. Then for all $x \in \mathcal{K}_{m+1}$, it follows from $f(x,\pi^*(x)) \in \mathcal{K}_\zeta$ for some $\zeta \in \mathcal{C}$, that

$$\begin{aligned} e_n(x) &= V(f(x,\pi^*(x))) + g(x,\pi^*(x)) + c_n(x) - V_n(x) \\ &= V(x) + c_n(x) - V_n(x), \end{aligned} \tag{2.31}$$

where $c_n(x) \to 0$ due to $V_n(x) \to V(x)$ for all $x \in \mathcal{K}_\zeta$. Consequently, by (2.25),

$$V_{n+1}(x) = V_n(x) + k[V(x) + c_n(x) - V_n(x)], \tag{2.32}$$

and thus

$$V_{n+1}(x) - V(x) - c_n(x) = [1-k][V_n(x) - V(x) - c_n(x)]. \tag{2.33}$$

Hence, $V_n(x) \to V(x)$ for all $x \in \mathcal{K}_{m+1}$ due to $k \in (0,2)$. □

Remark 2.3. *While Theorem 2.8 guarantees pointwise convergence of proportional feedback VI, a proof of convergence of the corresponding control policies, generated according to* (2.7)*, is left to future research. Convergence of minimizer/maximizer sequences implying convergence of control policies is a general issue in approximation methods for optimization and optimal control [56]. Note that for* $0 < k \leq 1$ *and* $V_0 = 0$*, it follows that* $V_n(x) \leq J(x,\pi_n) \leq V(x)$ *for all* x*, where* $\pi_n(x) \in \arg\max_{u \in U}\{g(x,u) + V_n(f(x,u))\}$ *is the approximation of the optimal control policy based on* V_n *at the current iteration* n*. Since* $V_n \to V$*, the total yield of* π_n *converges to the optimal total yield. The numerical examples in Section 2.6 and others, including comparisons with mixed-integer programming solutions in the linear system case [48], indicate that convergence to the optimal total yield also occurs for* $V_0 \neq 0$ *and* $1 < k < 2$*.*

2.3.2 Practical Considerations

Theorem 2.8 assumes that iterations (2.25) are applied to each $x \in G$. However, in practice, iterations (2.25) are applied to a discrete subset of G, which is denoted by $\tilde{G} = \{x^i \in G : i \in I\}$ with $I = \{1,2,...,i_{\max}\}$ (G is assumed to be a continuous set, which is the case for most practical problems). For $x \notin \tilde{G}$, $V_n(x)$ is approximated by a function approximator (e.g., linear interpolation). The approximation induces an error and, instead of $V_n \to V$, $V_n \to \tilde{V}$ as $e_n(x) \to 0$ for all $x \in \tilde{G}$, where $\tilde{V}$ is an approximation of the value function. The corresponding control policy is denoted by $\tilde{\pi}^*$, which is defined by (2.7) with V replaced by $\tilde{V}$. As $\tilde{G}$ becomes denser in G, it is expected that $\tilde{V} \to V$ and $\tilde{\pi}^*$

approaches an optimal control policy π^*.

The theoretical results in Theorem 2.8 suggest that the fastest convergence is achieved when $k = 1$. However, the convergence behavior may be different when iterations (2.25) are applied to the discretized problem, i.e., the discrete set $\tilde{G}$. To show this, suppose the assumptions of Theorem 2.8 hold and there exists a point $x' \in \tilde{G}$ with $V_n(x') \to \tilde{V}(x')$. Now assume $\tilde{G}$ is such that there exists another point $x \in \tilde{G}$, sufficiently close to x', for which $e_n(x)$ can be expressed as follows

$$e_n(x) = \tilde{V}(x') + [d(x) + \tilde{c}_n(x)][V_n(x) - \tilde{V}(x')] + g(x, \tilde{\pi}^*(x)) - V_n(x), \tag{2.34}$$

with $d(x) \in [0, 1)$ and $\tilde{c}_n(x) \to 0$ as illustrated in Figure 2.1. Based on (2.25) and (2.34),

$$e_{n+1}(x) = [\tilde{c}_{n+1}(x) - \tilde{c}_n(x)][V_{n+1}(x) - \tilde{V}(x')] + [1 - k(1 - d(x) - \tilde{c}_{n+1}(x))]e_n(x). \tag{2.35}$$

As $n \to \infty$, updates (2.35) can be approximated as $e_{n+1}(x) = [1 - k(1 - d(x))]e_n(x)$, suggesting $e_n(x) \to 0$ and thus $V_n(x) \to \tilde{V}(x)$, if

$$|1 - k(1 - d(x))| < 1. \tag{2.36}$$

Now suppose for each $x \in \tilde{G}$, there exists a neighbor $x' \in \tilde{G}$ such that $e_n(x)$ can be expressed by (2.34) with $\tilde{c}_n(x) \to 0$. Then $V_n(x) \to \tilde{V}(x)$ for all $x \in \tilde{G}$ if (2.36) holds for all $x \in \tilde{G}$. Note that $d(x)$ may be different for each x. Assuming that $d(x) \in [0, 1)$ for all $x \in \tilde{G}$, for some $x \in \tilde{G}$, $d(x)$ may be close to 1 and convergence occurs for $k \in (0, k_{\max})$ with $k_{\max} > 2$. However, there may be some $x \in \tilde{G}$ with $d(x)$ close or equal to 0, requiring $k \in (0, 2)$ for convergence. Thus, in line with Theorem 2.8, convergence follows for $k \in (0, 2)$ if $d(x) \in [0, 1)$ for all $x \in \tilde{G}$.

On the other hand, if $\tilde{G}$ and $\tilde{V}$ are such that there exists $x \in \tilde{G}$ for which $d(x) < 0$, i.e., there exists no neighbor x' such that (2.34) holds with $d(x) \in [0, 1)$, then $k \in (0, k_{\max})$ with $k_{\max} < 2$ is required to satisfy (2.36) and establish $V_n(x) \to \tilde{V}(x)$ for all $x \in \tilde{G}$. If there exists some $x \in \tilde{G}$ such that $d(x) = 1$, which is not possible in theory due to Assumptions 2.1 and 2.2, no convergence would occur for any k.

In contrast to the theoretical case in Theorem 2.8, the fastest convergence may not be achieved for $k = 1$, but for individual $k = 1/(1 - d(x))$ as implied by (2.36). Note that $k = 1$ is the optimal gain only if $d(x) = 0$ for all $x \in \tilde{G}$, which represents the theoretical case $\tilde{G} = G$. This motivates the introduction of individual adaptive gains, as it is considered in the following section (Section 2.3.3).

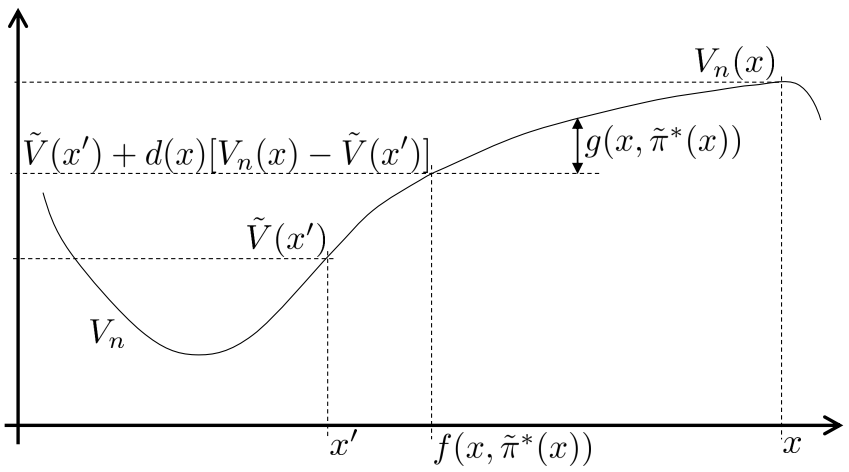

Figure 2.1: Illustration of (2.34), x and x' sufficiently close such that $\max_{u \in U}\{V_n(f(x,u)) + g(x,u)\} = \tilde{V}(x') + [d(x) + \tilde{c}_n(x)][V_n(x) - \tilde{V}(x')] + g(x, \tilde{\pi}^*(x))$, $d(x) \in [0,1)$ and $\tilde{c}_n(x) \to 0$.

Remark 2.4. *For the discretized problem, convergence of* (2.25) *to* $\tilde{V}$ *requires that* $\tilde{G} \cap \mathcal{K}_0$ *is nonempty. This is because at least one* $x' \in \tilde{G}$ *with* $V_n(x') \to \tilde{V}(x')$ *is required to "initiate" convergence of the other points in* $\tilde{G}$ *as described by* (2.34) – (2.36)*. In fact, for each* $x' \in \mathcal{K}_0$*,* $\tilde{V}(x') = V(x')$ *and* $V_n(x') \to V(x')$*, independent of* $\tilde{G}$*, if a solution to* (2.3) *exists and* $k \in (0,2)$*, as can be inferred from* (2.29) *and* (2.30)*.*

2.3.3 Adaptive Proportional Feedback VI

As explained in the previous section (Section 2.3.2), $k = 1$ may not be the optimal choice for fast convergence in a numerical setting. In fact, according to (2.36), the optimal gain may be a function of the state vector x, i.e., $k : G \to \mathbb{R}_+$. Therefore, proportional feedback VI is extended by introducing individual (i.e., state-dependent) gains. Furthermore, the gains $k(x)$ become adaptive in the sense that $k(x)$ is large when the error $e_n(x)$ is large and $k(x)$ is small when $e_n(x)$ is small. Thus, adaptive proportional feedback VI with learning rate $\delta \geq 0$ is as follows

$$\begin{aligned} &V_{n+1}(x) = V_n(x) + k_n(x)e_n(x), \text{ if } x \in G, \\ &V_{n+1}(x) = 0, \text{ if } x \notin G, \\ &k_{n+1}(x) = k_n(x) + \delta e_n(x), \end{aligned} \tag{2.37}$$

where the error $e_n(x)$ is given by (2.24). It is straightforward to show that, if the convergence conditions for k from Theorem 2.8 hold for each individual $k_n(x)$, i.e., if $k_n(x) \in$

$(0, 2)$ is introduced as a bound for all $x \in G$ and $n \in \mathbb{Z}_{\geq 0}$, then the sequence of functions defined in (2.37) converges pointwise to $V(x)$ for all $x \in G$. Note that the practical considerations in Section 2.3.2 also hold for adaptive proportional feedback VI.

2.4 ADP Approach

In this section, an ADP approach is presented to obtain an approximation of the value function and optimal control policy. The approach presented here employs proportional feedback VI, see (2.25). Note that adaptive proportional feedback VI, see (2.37), may be used as well.

As outlined in Section 2.3.2, proportional feedback VI is applied to a discretized subset $\tilde{G}$ of G, where $V_n(x)$ is computed according to

$$V_n(x) = \begin{cases} \text{function approximator at } x, \text{ if } x \in G, \\ 0, \text{ if } x \notin G. \end{cases} \tag{2.38}$$

After each iteration, the function approximator is updated using the data from the previous iteration, denoted by

$$V_{\text{train}} = \{V_{\text{train},x^i} : x^i \in \tilde{G}\}. \tag{2.39}$$

The algorithm is considered to be converged, i.e., $V_n \to \tilde{V}$, when $|e_n(x)| \leq \varepsilon$ for all $x \in \tilde{G}$, where $\varepsilon > 0$ is a prescribed threshold. This procedure is outlined in Algorithm 1.

Algorithm 2.1 ADP procedure to compute approximations of value function and optimal control policy

$\tilde{G} \leftarrow$ generate discrete subset of G
$n \leftarrow 0$
$V_{\text{train},x} \leftarrow$ set initial values for each $x \in \tilde{G}$
$V_0 \leftarrow$ update function approximator based on V_{train}
while $\max_{x \in \tilde{G}} |e_n(x)| > \varepsilon$ **do**
    for each $x \in \tilde{G}$ **do**
        $V_{\text{train},x} = V_n(x) + ke_n(x)$
    end for
    $V_{n+1} \leftarrow$ update function approximator based on V_{train}
    $n \leftarrow n + 1$
end while
$\tilde{V} \leftarrow V_n$

In analogy to (2.7), the approximation of the optimal control at $x \in G$ is given by

$$\tilde{\pi}^*(x) \in \arg\max_{u \in U} \left\{ g(x,u) + \tilde{V}(f(x,u)) \right\}. \tag{2.40}$$

Most ADP approaches use NNs as function approximators [57–60]. For the numerical case studies in this chapter (Section 2.6.2), Gaussian processes [61, 62] are used, as initial numerical studies suggest better results than with NNs.

2.5 Base-Trajectory VI

The focus of this section is on time maximization problems, i.e., DCOC problems with the objective of maximizing the first exit-time ($g \equiv 1$),

$$J(x_0, \pi) = \tau(x_0, \pi) \to \max_{\pi \in \Pi}. \tag{2.41}$$

According to (2.7), the optimal control policy may be characterized by the value function, in this case,

$$\pi^*(x) \in \arg\max_{u \in U} V(f(x,u)). \tag{2.42}$$

The value function may be obtained with conventional or (adaptive) proportional feedback VI (Section 2.3). In this section, a different algorithm is proposed and it is shown that, in a numerical setting, it is more accurate than conventional or (adaptive) proportional feedback VI, i.e., it provides better control policies that satisfy constraints longer. Unlike conventional or (adaptive) proportional feedback VI, the proposed algorithm converges to the value function by gradually connecting pieces of an optimal control policy. Using a specified base control policy, the corresponding base trajectory is traversed until deviating from the base control provides an improvement. This approach is motivated by properties of optimal control for spacecraft where the optimal solution consists of extensive coasting phases with zero-control (no thrust) as the base policy [44, 45]. The proposed algorithm is therefore referred to as base-trajectory VI.

Throughout this section, the following two assumptions are made.

Assumption 2.3. The set U contains the origin.

Assumption 2.4. A solution to problem (2.41) exists and is denoted by π^*.

Assumption 2.3 helps in the formulation of base-trajectory VI in Section 2.5.1. Note that every nonempty set in $\mathbb{R}^p$ can be transformed by a change of coordinates (origin shift) such that Assumption 2.3 is satisfied.

Remark 2.5. *The existence of a solution by Assumption 2.4 implies that there exists $\bar{T} > 0$ such that $\tau(x, \pi) \leq \bar{T}$ for all $x \in G$ and $\pi \in \Pi$.*

2.5.1 Theoretical Results

The base policy is defined as the zero-control policy π_0. The zero-control policy and the corresponding zero-control trajectory $S_0(x_0)$ that emanates from $x_0 \in G$ are given by

$$\pi_0 \in \Pi \text{ such that } \pi_0(x) = 0 \text{ for all } x \in G, \tag{2.43}$$

$$S_0(x_0) = \left\{x_0, x_1 = f(x_0, 0), x_2 = f(x_1, 0), ..., x_{\tau(x_0,\pi_0)-1} = f(x_{\tau(x_0,\pi_0)-2}, 0)\right\}. \tag{2.44}$$

Moreover, the time instant that corresponds to a state x' on the zero-control trajectory $S_0(x_0)$ is defined by

$$t_{x'}(x_0) = \inf \left\{t \in \mathbb{Z}_+ : x_{t-1} = x' \in S_0(x_0)\right\}. \tag{2.45}$$

Base-trajectory VI is based on finding the state on the zero-control trajectory at which it is optimal to switch to a different trajectory by using the control input $u \in U$. While, without loss of generality, the zero-control policy is assumed as the base policy, an alternative policy, which ideally is known to be optimal on a subset of sets, can be chosen as the base policy. With this idea, an expression for the value function V can be formulated as stated in the following theorem, where the pair $(x^*, u) \in S_0(x) \times U$ defines the state on the zero-control trajectory at which the switch to a different trajectory occurs using the control input u.

Theorem 2.9. *Suppose Assumptions 2.3 and 2.4 hold. Then the value function V is the unique solution to*

$$\tilde{V}(x) = \max_{(x^*,u)\in S_0(x)\times U} \left\{\tilde{V}(f(x^*, u)) + t_{x^*}(x)\right\}, \tag{2.46}$$

for all $x \in G$, and $\tilde{V}(x) = 0$ if $x \notin G$.

Proof. For $x \notin G$, $\tau(x, \pi) = 0$ and thus $V(x) = \tilde{V}(x) = 0$. Now let $x = x_0 \in G$ be a given initial state and apply (2.46) until the resulting state trajectory,

$$\left\{x, ..., x^{*1}, f(x^{*1}, u^{*1}), ..., x^{*2}, f(x^{*2}, u^{*2}), ..., x_{\tilde{\tau}(x)}\right\}, \tag{2.47}$$

exits G, where the corresponding exit-time is denoted by $\tilde{\tau}(x)$, which is bounded by As-

sumption 2.4 and Remark 2.5. Moreover, (x^{*k}, u^{*k}) in (2.47) is the argument that maximizes (2.46) when (2.46) is applied for the k-th time to evaluate $V(f(x^{*k-1}, u^{*k-1}))$, where $x \triangleq f(x^{*0}, u^{*0})$. Hence, (2.46) may be written as

$$\begin{aligned}\tilde{V}(x) &= t_{x^{*1}} + \tilde{V}(f(x^{*1}, u^{*1})) \\ &= t_{x^{*1}}(x) + t_{x^{*2}}(f(x^{*1}, u^{*1})) + ... + \tilde{V}(x_{\tilde{\tau}(x)}) \\ &= \sum_{t=0}^{\tilde{\tau}(x)-1} 1 \\ &= \tilde{\tau}(x),\end{aligned} \tag{2.48}$$

since $\tilde{V}(x_{\tilde{\tau}(x)}) = 0$ due to $x_{\tilde{\tau}(x)} \notin G$. It follows from (2.4), (2.46), (2.48), and the principle of optimality [63] that $\tilde{V}(x) = \max_{\pi \in \Pi} \tau(x, \pi) = V(x)$. □

In analogy to (2.46) from Theorem 2.9, the base-trajectory VI algorithm is defined as follows

$$\begin{aligned}V_n(x) &= \max_{(x^*, u) \in S_0(x) \times U} \{V_{n-1}(f(x^*, u)) + t_{x^*}(x)\}, \text{ if } x \in G, \\ V_n(x) &= 0, \text{ if } x \notin G.\end{aligned} \tag{2.49}$$

The sequence of functions in (2.49) is initialized by counting the steps on the zero-control trajectory until the constraints are violated, i.e., $V_0(x) = \tau(x, \pi_0)$ for all $x \in G$. This is summarized in Algorithm 2.2. The convergence of (2.49) is analyzed in Theorem 2.10.

Algorithm 2.2 Base-trajectory VI: compute $V_0(x)$, $x \in G$

```
x_0 ← x
t ← 0
while x_t ∈ G do
    x_{t+1} ← f(x_t, 0)
    t ← t + 1
end while
return V_0(x) ← t
```

Theorem 2.10. *Suppose Assumptions 2.3 and 2.4 hold. Then the sequence* $\{V_n(x)\}$ *defined by* (2.49) *converges pointwise to* $V(x)$ *for all* $x \in G$.

Proof. First, it is shown by induction that $\{V_n(x)\}$ is monotonically non-decreasing. For $n = 0$, it follows from Algorithm 2.2 that $V_0(x) = \tau(x, \pi_0)$ for all $x \in G$, where π_0 is

defined in (2.43). Consequently, for all $x \in G$,

$$\begin{aligned} V_1(x) &= \max_{(x^*,u)\in S_0(x)\times U} \{V_0(f(x^*,u)) + t_{x^*}(x)\} \\ &\geq \tau(x,\pi_0) = V_0(x). \end{aligned} \tag{2.50}$$

Now assume that for some n, $V_n(x) \geq V_{n-1}(x)$ for all $x \in G$. Using this assumption and (2.49), it follows that

$$\begin{aligned} V_{n+1}(x) - V_n(x) = &\max_{(x^*,u)\in S_0(x)\times U} \{V_n(f(x^*,u)) + t_{x^*}(x)\} \\ &- \max_{(x^*,u)\in S_0(x)\times U} \{V_{n-1}(f(x^*,u)) + t_{x^*}(x)\} \geq 0, \end{aligned} \tag{2.51}$$

which shows that $\{V_n(x)\}$ is monotonically non-decreasing for all $x \in G$. Next, proceeding by induction, it is shown that $\{V_n(x)\}$ is bounded by $V(x)$ for all $x \in G$. By Theorem 2.9, V is the unique solution to (2.46). Hence,

$$\begin{aligned} V(x) &= \max_{(x^*,u)\in S_0(x)\times U} \{V(f(x^*,u)) + t_{x^*}(x)\} \\ &\geq \tau(x,\pi_0) = V_0(x). \end{aligned} \tag{2.52}$$

Now assume that for some n, $V(x) \geq V_n(x)$ for all $x \in G$. Using this assumption as well as (2.46) and (2.49), it follows that for all $x \in G$,

$$\begin{aligned} V(x) - V_{n+1}(x) = &\max_{(x^*,u)\in S_0(x)\times U} \{V(f(x^*,u)) + t_{x^*}(x)\} \\ &- \max_{(x^*,u)\in S_0(x)\times U} \{V_n(f(x^*,u)) + t_{x^*}(x)\} \geq 0. \end{aligned} \tag{2.53}$$

Since $\{V_n(x)\}$ is a monotonic and bounded sequence, it converges pointwise to a function $\tilde{V}(x)$ defined by (2.46), where $V = \tilde{V}$ according to Theorem 2.9. □

Remark 2.6. *For base-trajectory VI, the set $S_0(x) \times U$ over which the expression in* (2.49) *is maximized can be reduced to $S_0(x) \times U \setminus \{0\}$. This is because if $(x^*, 0)$, where $x^* \in S_0(x)$, maximizes* (2.49)*, then $V_n(x) = \tau(x, \pi_0)$. This implies that $(x_{\tau(x,\pi_0)-1}, u)$, where $x_{\tau(x,\pi_0)-1} \in S_0(x)$, is also a maximizer in* (2.49) *for any $u \in U$.*

2.5.2 Numerical Implementation

In the following, the numerical implementation of base-trajectory VI and conventional VI [$k = 1$ in (2.25) or $k_0 \equiv 1$ and $\delta = 0$ in (2.37))] are compared. Note that similar results hold when comparing base-trajectory VI with (adaptive) proportional feedback VI ($k \neq 1$).

Algorithm 2.3 Numerical implementation of base-trajectory VI: compute $V_{\text{dis}}(x)$, $x \in G_{\text{dis}}$

$V_0(x) \leftarrow$ output of Algorithm 2.2
$n \leftarrow 1, \;\; \Delta V_1(x) \leftarrow 2\varepsilon$
while $\Delta V_n(x) \geq \varepsilon$ **do**
    $x_0 \leftarrow x, \;\; t \leftarrow 0$
    $\hat{V}_{\max} \leftarrow -1$
    while $x_t \in G$ **do**
        $\hat{V} \leftarrow \max\limits_{u \in U_{\text{dis}} \setminus \{0\}} \{F_{n-1}(f(x_t, u))\} + t$, see (2.56)
        if $\hat{V} \geq \hat{V}_{\max}$ **then**
            $\hat{V}_{\max} \leftarrow \hat{V}$
        end if
        $x_{t+1} \leftarrow f(x_t, 0)$
        $t \leftarrow t + 1$
    end while
    $V_n(x) \leftarrow \hat{V}_{\max}$
    $n \leftarrow n + 1$
    $\Delta V_n(x) \leftarrow |V_{n-1}(x) - V_{n-2}(x)|$
end while
return $V_{\text{dis}}(x) \leftarrow V_{n-1}(x)$

For the numerical implementation of base-trajectory VI as well as of conventional VI, a typical approach used in applications is considered, which is based on a discretization of the problem. In analogy to Section 2.3.2, the discretized subsets of G and U are denoted by

$$G_{\text{dis}} = \left\{x^i \in G, i \in I_x\right\}, \tag{2.54}$$

$$U_{\text{dis}} = \left\{u^i \in U, i \in I_u\right\}. \tag{2.55}$$

Similar to Assumption 2.3, it is assumed that the set U_{dis} contains the origin. Interpolation is used to evaluate V_n at $x \notin G_{\text{dis}}$,

$$\begin{aligned} F_n(x) &= \text{Interpolant}\left[V_n\right](x), \;\; \text{if } x \in G, \\ F_n(x) &= 0, \;\; \text{if } x \notin G, \end{aligned} \tag{2.56}$$

where linear interpolation is used for the numerical case studies in this chapter. Consequently, the approximate version of base-trajectory VI (2.49) is given by

$$\begin{aligned} V_n(x) &= \max_{(x^*,u) \in S_0(x) \times U_{\text{dis}} \setminus \{0\}} \{F_{n-1}(f(x^*, u)) + t_{x^*}(x)\}, \;\; \text{if } x \in G, \\ V_n(x) &= 0, \;\; \text{if } x \notin G, \end{aligned} \tag{2.57}$$

where the exclusion of the origin for the control input is due to Remark 2.6. For each $x \in G_{\text{dis}}$, the sequence of functions in (2.57) approaches V_{dis} pointwise, which is an approximation of the value function V. Algorithm 2.3 shows an implementation of (2.57), where $\varepsilon > 0$ is a convergence threshold.

Using (2.57) or Algorithm 2.3, respectively, an approximation of the optimal control policy π^* is obtained for all $x \in G$,

$$\pi_{\text{dis}}(x) \in \underset{u \in U_{\text{dis}}}{\arg\max}\, F_{\text{dis}}(f(x,u)), \tag{2.58}$$

where

$$\begin{aligned} &F_{\text{dis}}(x) = \text{Interpolant}\left[V_{\text{dis}}\right](x), \text{ if } x \in G, \\ &F_{\text{dis}}(x) = 0, \text{ if } x \notin G. \end{aligned} \tag{2.59}$$

Similar to (2.57), the approximate version of conventional VI is

$$\begin{aligned} &V_n(x) = \max_{u \in U_{\text{dis}}} F_{n-1}(f(x,u)) + 1, \text{ if } x \in G, \\ &V_n(x) = 0, \text{ if } x \notin G, \end{aligned} \tag{2.60}$$

which yields the approximate value function $V_{\text{dis,VI}}(x)$ for all $x \in G_{\text{dis}}$ and an approximation of the optimal control policy for all $x \in G$,

$$\pi_{\text{dis,VI}}(x) \in \underset{u \in U_{\text{dis}}}{\arg\max}\, F_{\text{dis,VI}}(f(x,u)), \tag{2.61}$$

where $F_{\text{dis,VI}}(x)$ is defined in analogy to $F_{\text{dis}}(x)$ in (2.59).

In general, $V \neq V_{\text{dis}} \neq V_{\text{dis,VI}}$ and $\pi^* \neq \pi_{\text{dis}} \neq \pi_{\text{dis,VI}}$, which is mainly due to the interpolation error imposed by (2.56). The interpolation error affects conventional VI and base-trajectory VI differently. The fraction of $V_n(x)$ that is affected by interpolation is greater for conventional VI, where only $1/V_n(x)$ is not subject to an interpolation error as can be seen in (2.60). In contrast, for base-trajectory VI, the fraction of $V_n(x)$ that is not directly affected by interpolation is $t_{x^*}(x)/V_n(x)$, where x^* is part of the argument that maximizes the expression in (2.57) and $t_{x^*}(x) \geq 1$ by (2.45). Consequently, the interpolation error is smaller for base-trajectory VI. Hence, it is expected that $\sum_{x \in G_{\text{dis}}} |V(x) - V_{\text{dis}}(x)| \leq \sum_{x \in G_{\text{dis}}} |V(x) - V_{\text{dis,VI}}(x)|$ and π_{dis} is closer to π^* than $\pi_{\text{dis,VI}}$.

Remark 2.7. *Instead of using interpolation in* (2.56)*, F_n and the approximate value function in* (2.59) *can be described by any suitable function approximator (for example, by NNs). Hence, the ADP approach in Section 2.4 can be extended and, instead of using*

proportional feedback VI, the proposed algorithm (i.e., base-trajectory VI) may be used to train the respective function approximator in line 7 of Algorithm 2.1. The ADP implementation of base-trajectory VI will be investigated in future work, where similar advantages over conventional or (adaptive) proportional feedback VI are expected due to its superior accuracy (see above).

Remark 2.8. *For base-trajectory VI, numerical studies (Section 2.6) suggest that, for each $x \in G_{\text{dis}}$, there exists $i_{\text{th}}(x) \geq 0$ such that the optimal pair (x^*, u) that maximizes the expression in* (2.57) *is the same for all subsequent iterations $n > i_{\text{th}}(x)$. Hence, when $n > i_{\text{th}}(x)$, instead of traversing the zero-control trajectory from $t = 0$ to $t = \tau(x, \pi_0)$ in Algorithm 2.3, lines 4 to 14 in Algorithm 2.3 can be reduced to*

$$V_n(x) = F_{n-1}\left(f(x^{**}, u^*)\right) + t_{x^{**}}(x), \tag{2.62}$$

where

$$(x^{**}, u^*) \in \underset{S_0(x) \times U_{\text{dis}} \setminus \{0\}}{\arg\max} \{F_{i_{\text{th}}(x)}\left(f(x^*, u)\right) + t_{x^*}(x)\}. \tag{2.63}$$

This reduces the computation time of base-trajectory VI. However, for most applications, it is not possible to determine $i_{\text{th}}(x)$ exactly for each $x \in G_{\text{dis}}$. In the numerical case studies in Section 2.6, a global $\tilde{i}_{\text{th}}$ is used sufficiently large such that $\tilde{i}_{\text{th}} > i_{\text{th}}(x)$ for most $x \in G_{\text{dis}}$, which yields an approximation of V_{dis}.

Remark 2.9. *Both conventional or (adaptive) proportional feedback VI and base-trajectory VI are parallelizable and multiple processing units (cores) may be used to obtain the approximate value function for all $x \in G_{\text{dis}}$. This is achieved by partitioning G_{dis} into ν sets of similar size,*

$$G_{\text{dis}} = G_{\text{dis},1} \cup G_{\text{dis},2} \cup ... \cup G_{\text{dis},\nu}, \tag{2.64}$$

where ν is the number of available processing units and the i-th core, $i = 1, 2, ..., \nu$, computes the value function for all $x \in G_{\text{dis},i}$ using either conventional or (adaptive) proportional feedback VI or base-trajectory VI.

2.6 Numerical Case Studies

2.6.1 LEO Satellite Station Keeping 1

Using (adaptive) proportional feedback VI (Section 2.3), the objective in this case study is to obtain a numerical approximation of a control policy that maximizes the operational

time of a LEO satellite. Hence, the yield function is $g \equiv 1$, i.e., $g(x,u) = 1$ for all $(x,u) \in G \times U$. The case study considers a satellite in an equatorial near-circular LEO subject to atmospheric drag and J2 perturbation. Thus, the satellite's motion can be modeled in polar coordinates (r, θ), where r is the radial distance from Earth's center to the satellite and θ is the polar angle of the position vector. The respective unit vectors are denoted by $\hat{r}$ and $\hat{\theta}$. The satellite's velocity in the $\hat{r}$-direction is v_r and v_θ is the rate of change of the polar angle. With m as the satellite's mass, the state vector is given by $x = [r, v_r, \theta, v_\theta, m]^\top$. The control input $u \in U = \{0, F_\mathrm{t}\}$ is an on-off thrust force F_t acting in the $\hat{\theta}$-direction. The discrete-time model for the LEO satellite is obtained from the continuous-time model [64, 65] using Euler's forward method with $\Delta t = 1$ sec, yielding

$$x_{t+1} = x_t + \Delta t[v_{r,t}, a_{r,t}, v_{\theta,t}, a_{\theta,t}, -c|u_t|]^\top, \tag{2.65}$$

with

$$a_{\theta,t} = -2v_{r,t}v_{\theta,t}/r_t - F_{\mathrm{D},\theta,t}/(r_t m_t) + u_t/(r_t m_t),$$
$$a_{r,t} = r_t v_{\theta,t}^2 - \mu/r_t^2 - F_{\mathrm{D},r,t}/m_t - 3\mu r_\mathrm{E}^2 J_2/(2r_t^4),$$

where $F_{\mathrm{D},\theta,t} = A_\theta \rho c_{\mathrm{d},\theta}(r_t v_{\theta,t})^2/2$ and $F_{\mathrm{D},r,t} = A_r \rho c_{\mathrm{d},r} v_{r,t}|v_{r,t}|/2$. The parameters r_E and μ are Earth's radius and gravitational parameter, respectively, and $J_2 = 1082.64 \times 10^{-6}$.

Note that the J2 perturbation term does not directly affect the polar angle due to the assumption of an equatorial orbit. The parameters of the drag perturbation are the satellite's reference areas in radial and polar directions A_r and A_θ, respectively, as well as the respective drag coefficients $c_{\mathrm{d,r}}$ and $c_{\mathrm{d},\theta}$. The atmospheric density is denoted by ρ. The constant c in (2.65) is the effective velocity of the thruster's exhaust jet, which may be expressed using the specific impulse I_sp of the thruster: $c = I_\mathrm{sp} 9.81 \mathrm{m/s}^2$.

The satellite that is considered is a three-unit (3-U) CubeSat with a dry mass of 4 kg. The satellite is assumed to have the following aerodynamic parameters: $c_{\mathrm{d,r}} = 2.1$, $A_\mathrm{r} = 0.0374\ \mathrm{m}^2$, $c_{\mathrm{d},\theta} = 2.3$, and $A_\theta = 0.0154\ \mathrm{m}^2$. The parameters of the thruster are adapted from a monopropellant hydrazine propulsion system for CubeSats described in [66, 67]. The thrust force is $F_\mathrm{t} = 0.96$ N and the specific impulse is $I_\mathrm{sp} = 221.4$ sec, yielding $c = 2,171.4\ \mathrm{m/sec}$. The propulsion system is of the size of a one-unit (1-U) CubeSat and consequently comprises a third of the example satellite. It is assumed that the propulsion system initially carries 350 g of fuel, thus $m_0 = 4.35$ kg.

The objective in this example is to maximize the time that the satellite stays within ± 10 % of the nominal orbital altitude of $h_0 = 300$ km. Therefore, given the fuel constraints

described above, the set one wants the state vector to remain inside is given by

$$G = \{x : 270\,\mathrm{km} \leq r - r_{\mathrm{E}} \leq 330\,\mathrm{km},\ m \geq 4\,\mathrm{kg}\}. \tag{2.66}$$

For numerical implementation of (adaptive) proportional feedback VI (see Section 2.3.2), the following discrete state space $\tilde{G} \subset G$ is considered

$$\begin{aligned}\tilde{G} = \{x : r - r_{\mathrm{E}} \in \{&270,\ 270 + 60/89,\ 270 + 120/89, ...,\ 330\}\,\mathrm{km},\\ v_r \in \{&-7,\ -7 + 14/29,\ -7 + 28/29, ...,\ 7\}\,\mathrm{m/sec},\\ v_\theta \in \{&1.145,\ 1.145 + 0.023/19,\ 1.145 + 0.046/19, ..., 1.168\}\,10^{-3}\,\mathrm{rad/sec},\\ m \in \{&4,\ 4 + 0.35/9,\ 4 + 0.7/9, ...,\ 4.35\}\,\mathrm{kg}\}.\end{aligned}$$

According to the satellite model in (2.65), the polar angle θ has no effect on the other states. Therefore, θ has no effect on G and, consequently, the value function V only depends on r, v_r, v_θ, and m.

The density ρ for a circular equatorial 300 km orbit is between 2.23×10^{-11} kg/m^3 and 3.72×10^{-11} kg/m^3 [68]. Here, a constant density of $\rho = 3.5 \times 10^{-11}$ kg/m^3 is assumed.

Note that for all states in G, the orbit will eventually decay ($r < r_{\mathrm{E}} + 270\,\mathrm{km}$) in finite time using no control ($u = 0$) due to atmospheric drag. Using control ($u \geq 0$) may extend the time until $r < r_{\mathrm{E}} + 270\,\mathrm{km}$. However, the system will eventually violate $m \geq 4\,\mathrm{kg}$ in finite time due to the finite amount of fuel. Based on these observations and Theorem 2.5, the DCOC problem is well-posed and a solution exists. In the following, such solutions are constructed numerically.

2.6.1.1 Proportional Feedback VI

First, proportional feedback VI, defined in (2.25), is analyzed. Throughout this case study, linear interpolation is used to evaluate $V_n(x)$ when $x \notin \tilde{G}$ and the initial values are set to $V_0(x) = 2$ for all $x \in \mathcal{H}$ and $V_n(x) = 1$ for all $x \in \mathcal{K}_0$ and $n \in \mathbb{Z}_{\geq 0}$, where $\mathcal{H}$ and $\mathcal{K}_0$ are given by (2.26) and (2.27), respectively. The convergence criterion for the algorithm is

$$\max_{x \in \tilde{G}} |e_n(x)| \leq \varepsilon, \tag{2.67}$$

where $\varepsilon = 10^{-3}$.

The top of Figure 2.2 shows the number of iterations until convergence for different values of the proportional gain k. The required number of iterations to convergence appears to decrease with increasing k. In contrast to Theorem 2.8 which requires $k \in (0, 2)$,

convergence also occurs for $k \geq 2$. This is due to the discretization of the state space for numerical purposes as explained in Section 2.3.2. Conventional VI ($k = 1$) converges in 20524 iterations. From the investigated gains $k \in \{0.5, 0.75, 1, ...\}$, the slowest convergence results for $k = 0.5$ (37659 iterations). The fastest convergence results for $k = 2.75$ with 8830 iterations, which is 2.3 times faster than with conventional VI. The algorithm fails to converge for $k \geq 3$.

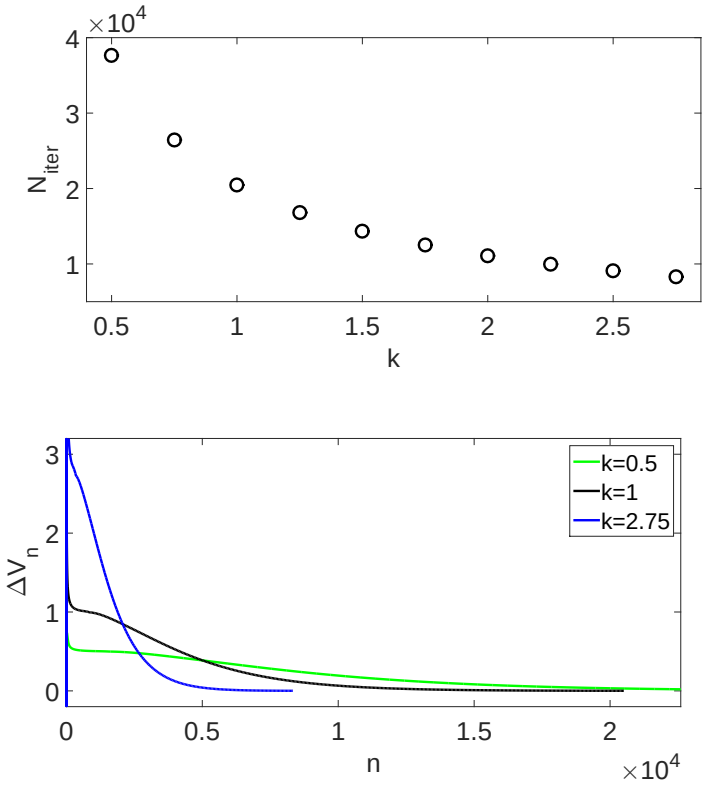

Figure 2.2: LEO satellite station keeping problem (case study 1), proportional feedback VI (2.25). Top: number of iterations, N_{iter}, until convergence vs. proportional gain k. Bottom: difference between V_n and V_{n-1} according to (2.68) vs. n.

The following criterion is used to assess the difference between the functions V_n and V_{n-1} at each iteration n of the algorithm,

$$\Delta V_n = \max_{x \in \tilde{G}} |V_n(x) - V_{n-1}(x)| \operatorname{sgn}(V_n(x^*) - V_{n-1}(x^*)), \tag{2.68}$$

where $x^* \in \arg\max_{x \in \tilde{G}} |V_n(x) - V_{n-1}(x)|$. The bottom of Figure 2.2 shows the difference between V_n and V_{n-1} at each iteration n for $k = 0.5$, $k = 1$, and $k = 2.75$ using criterion (2.68). For $k = 2.75$, ΔV_n initially oscillates around zero and becomes non-negative after

$n = 8$ iterations. Moreover, for the first 100 iterations, the differences between the current and previous iteration are larger for $k = 2.75$ compared to $k = 1$ and $k = 0.5$. Hence, larger updates to V_n can be made and the algorithm converges faster for $k = 2.75$.

2.6.1.2 Adaptive Proportional Feedback VI

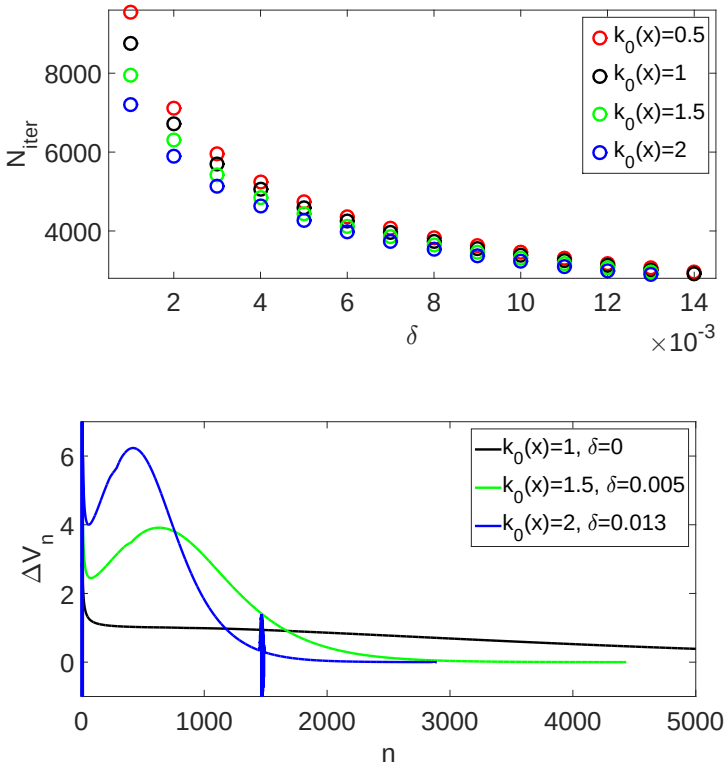

Figure 2.3: LEO satellite station keeping problem (case study 1), adaptive proportional feedback VI (2.37). Top: number of iterations until convergence, N_{iter}, vs. learning rate δ. Bottom: difference between V_n and V_{n-1} according to (2.68) vs. n.

Now adaptive proportional feedback VI, see (2.37), is used to numerically solve the DCOC problem. As before, the convergence criterion is given by (2.67) with $\varepsilon = 10^{-3}$. The number of iterations to convergence is shown in the top of Figure 2.3 for different learning rates $\delta \in \{0.001, 0.002, ...\}$ and different initial gains $k_0(x) = k_0$ for all $x \in \tilde{G}$. For a given initial gain $k_0(x)$, the number of iterations to convergence decreases with increasing learning rate. Furthermore, for a given δ, convergence occurs faster with increasing $k_0(x)$. However, the influence of $k_0(x)$ decreases with higher learning rates. The iterations fail to converge for initial gains of 0.5 and 1 if $\delta \geq 0.015$ and for initial gains of 1.5 and 2 if $\delta \geq 0.014$. The fastest convergence is achieved with $k_0(x) = 2$ and $\delta = 0.013$ where

convergence occurs after 2882 iterations, which is more than seven times as fast as with conventional VI. Note that for all different parameter settings, the same approximation of the value function is obtained within the prescribed accuracy.

The difference between V_n and V_{n-1} is plotted against the iteration number n for two example cases, $k_0(x) = 2$ with $\delta = 0.013$ and $k_0(x) = 1.5$ with $\delta = 0.005$, in the bottom of Figure 2.3. The plot also includes the first 5000 iterations of conventional VI ($k_0(x) = 1$ with $\delta = 0$) for comparison. Due to the relatively small and static gain, conventional VI only generates small updates to $V_n(x)$ at each iteration. Therefore, given the initial $V_0(x) = 2$, convergence is slow where $V(x)$ is large. In contrast, the adaptive algorithm increases the gain where $V(x)$ is large, which results in larger updates and thus faster convergence. Using $k_0(x) = 2$ with $\delta = 0.013$, the sequence of functions V_n begins to oscillate after about 1450 iterations. However, the oscillations decay within the next 50 iterations and the sequence eventually converges. The oscillations indicate that the algorithm is close to becoming unstable for the given parameter values.

2.6.1.3 Sample Trajectories

An optimal control policy satisfies (2.7). Hence, since $g \equiv 1$, an approximation of an optimal control policy is given by $\tilde{\pi}^*(x) \in \arg\max_{u \in U} \tilde{V}(f(x,u))$, where $\tilde{V}$ is an approximation of the value function obtained by the numerical implementation of (adaptive) proportional feedback VI (see Sections 2.6.1.1 and 2.6.1.2). During the closed-loop simulations, cubic interpolation is used to evaluate $\tilde{V}(x)$ if $x \notin \tilde{G}$. The control policy is applied to two test cases with different initial conditions.

For the first test case, the satellite is initially in a circular orbit of the nominal altitude $h_0 = 300$ km. The satellite's altitude for the first two weeks is shown in Figure 2.4, where the prescribed boundaries are indicated by blue lines. Note that the satellite would already violate the altitude constraints after only 5.5 days without using any control ($u \equiv 0$). The DCOC-based policy, however, steers the satellite into an elliptical orbit with an average altitude of about 299 km. As seen in the bottom of Figure 2.4, this orbit is maintained by using only sporadic control pulses. The fuel consumption in the first week is about 4.4 $\%$ of the available fuel, where the majority of the fuel is used in the first minutes for establishing the elliptical orbit. The fuel consumption in the second week is lower with approximately 2.8 $\%$ of the available fuel. The prescribed constraints are violated after 339 days.

The second test case assumes an initial circular orbit of 275 km altitude which is close to the lower bound for r. The altitude for the first two weeks is shown in the top of Figure 2.5, indicating that the DCOC-based control policy is able to maintain the satellite within the specified constraints. The control inputs are shown in the bottom of Figure 2.5. Eventually,

the fuel is depleted and constraint violation occurs after 268 days. Without using any control, the system violates the altitude constraints after only 14 minutes.

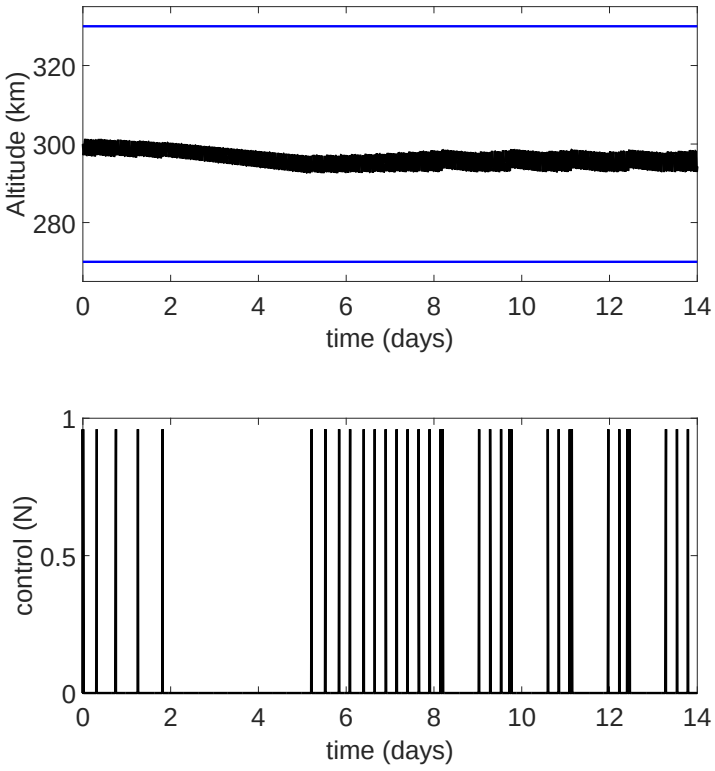

Figure 2.4: LEO satellite station keeping problem (case study 1) – initial altitude of 300 km: altitude $r - r_{\mathrm{E}}$ (top) and control input u (bottom) vs. time.

2.6.2 VDP Oscillator 1 and LEO Satellite Station Keeping 2

This section presents two time maximization problems ($J = \tau$, i.e., $g \equiv 1$) as numerical case studies. A VDP oscillator is considered in the first case study (Section 2.6.2.1) and a LEO satellite in the second case study (Section 2.6.2.2). In both case studies, an approximation of the optimal control policy is computed with the ADP procedure in Algorithm 2.1. In contrast to most ADP approaches that are based on NNs [59,60], a Gaussian process (Kriging interpolation) [61, 62] is employed to approximate the value function and V_n, see (2.38), as initial numerical studies suggest better results than with NNs.

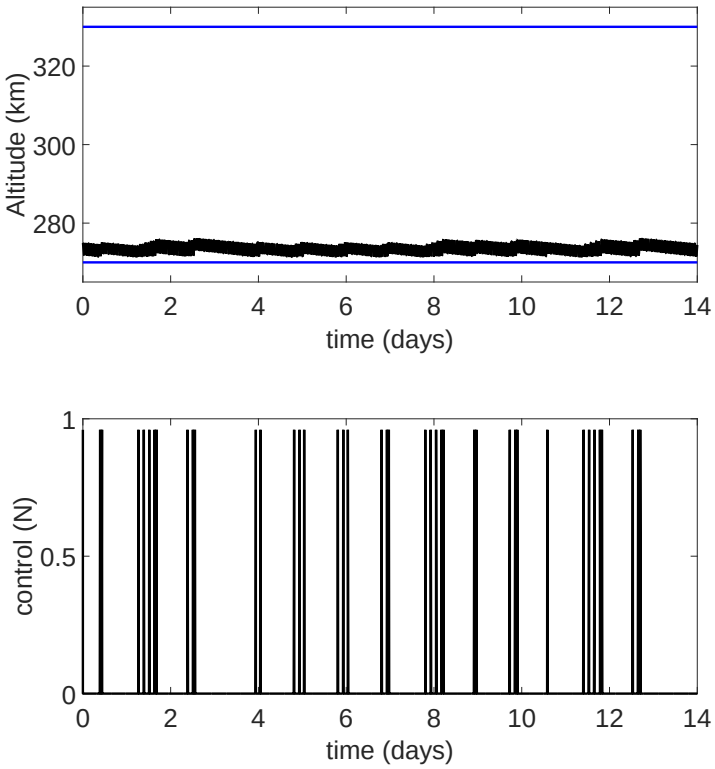

Figure 2.5: LEO satellite station keeping problem (case study 1) – initial altitude of 275 km: altitude $r - r_{\mathrm{E}}$ (top) and control input u (bottom) vs. time.

The DACE toolbox for MATLAB [69] is used to represent the Kriging model, where a linear regression model and a linear covariance function are chosen for the Kriging model (options *regpol1* and *corrlin*, respectively, in the DACE toolbox). The ADP approach is compared to a common approach in conventional DP, where V is approximated on a set of discrete points (grid), $\tilde{G} \subset G$, with equidistant spacing. Iterations (2.25) are applied to the grid using linear interpolation to approximate V_n between the grid points and $V_n(x) = 0$ if $x \notin G$. This approach is referred to as the DP approach in the following and the results of the ADP and DP approach are compared. For both approaches, the initial values are set to $V_{\mathrm{train},x} = 1$ (step 3 of Algorithm 2.1) and $V_0(x) = 1$, respectively, for each point x of the discretized set $\tilde{G} \subset G$. A convergence threshold of $\varepsilon = 0.1$ is used for the VDP problem and $\varepsilon = 0.5$ (due to greater first exit-times) for the LEO station keeping problem. Note that there is no significant difference in the results when using $\varepsilon < 0.1$ for the VDP problem and $\varepsilon < 0.5$ for the LEO station keeping problem.

In order to generate the discretized set $\tilde{G}$, a grid of q_i points on $\mathbb{R}$ with equidistant

spacing from $a_i \leq b_i$ to b_i is denoted by

$$\mathrm{gd}(a_i, b_i, q_i) = \left\{ a_i, a_i + \frac{b_i - a_i}{q_i - 1}, a_i + 2\frac{b_i - a_i}{q_i - 1}, ..., b_i \right\}. \tag{2.69}$$

Furthermore, an n-dimensional grid with $q_1 \times q_2 \times ... \times q_n$ points is defined by

$$\begin{aligned} \mathrm{GD}(a, b, q) = \{ & x \in \mathbb{R}^n : x_1 \in \mathrm{gd}(a_1, b_1, q_1), \\ & x_2 \in \mathrm{gd}(a_2, b_2, q_2),\ ..., x_n \in \mathrm{gd}(a_n, b_n, q_n) \}, \end{aligned} \tag{2.70}$$

where $x = [x_1, x_2, ..., x_n]^\top$, $a = [a_1, ..., a_n]^\top$, $b = [b_1, ..., b_n]^\top$, and $q = [q_1, ..., q_n]^\top$. In addition, a set of n_{lhs} points in $\mathbb{R}^n$ obtained by Latin hypercube sampling (lhs) is denoted by $\mathrm{lhs}(n_{\mathrm{lhs}})$. The discretization of G for the ADP approach is constructed by combining (2.70) and $\mathrm{lhs}(n_{\mathrm{lhs}})$, i.e., $\tilde{G} = \mathrm{GD}(a, b, q) \cup \mathrm{lhs}(n_{\mathrm{lhs}})$, where the MATLAB function *lhsdesign* is used to generate $\mathrm{lhs}(n_{\mathrm{lhs}})$. All computations reported in this section (Section 2.6.2) are performed in MATLAB 2015a on a laptop with an i5-6300 processor and 8 GB RAM.

2.6.2.1 VDP Oscillator 1

The discrete-time model of a forced van der Pol oscillator is obtained from the continuous-time model based on Euler's forward method,

$$x_{t+1} = x_t + \Delta t [r_{2,t}, s_t, 1]^\top, \tag{2.71}$$

where $s_t = u_t \sin(\omega r_{3,t}) + 2(1 - r_{1,t}^2) r_{2,t} - r_{1,t}$, $x_t = [r_{1,t}, r_{2,t}, r_{3,t}]^\top$ denotes the state vector at a time instant t, and the control input is u_t. The sampling time is set to $\Delta t = 0.01$ sec and $\omega = 10\ \mathrm{sec}^{-1}$. The objective is to maximize the first exit-time from the set

$$G = \{ x : r_1 \in [1, 3], r_2 \in [1, 3] \}, \tag{2.72}$$

subject to control constraints defined by $U = \{-10, -9, ..., 0, ..., 9, 10\}$. In order to construct $\mathrm{GD}(a, b, q)$, see (2.70), $a = [1, 1, 0]^\top$ and $b = [3, 3, 1\,\mathrm{sec}]^\top$. For the ADP approach where $\tilde{G} = \mathrm{GD}(a, b, q) \cup \mathrm{lhs}(n_{\mathrm{lhs}})$, $q = [3, 3, 3]^\top$ is selected for the grid part. For the DP approach, $\tilde{G} = \mathrm{GD}(a, b, q)$ is parameterized by $q = [q_0, q_0, q_0]^\top$.

Figure 2.6 (top) plots the number of iterations, N_{iter}, until convergence against the proportional gain factor k for the DP ($q_0 = 25$) and ADP approach ($n_{\mathrm{lhs}} = 25$ for lhs). The corresponding computation time t_{comp} is shown in Figure 2.6 (bottom). Note that the configurations shown in Figure 2.6 converge to the same control policy for the ADP approach (within the tolerance prescribed by ε). Likewise, the resulting control policies

in Figure 2.6 are the same for the DP approach. However, the DP control policy may be different from the ADP control policy.

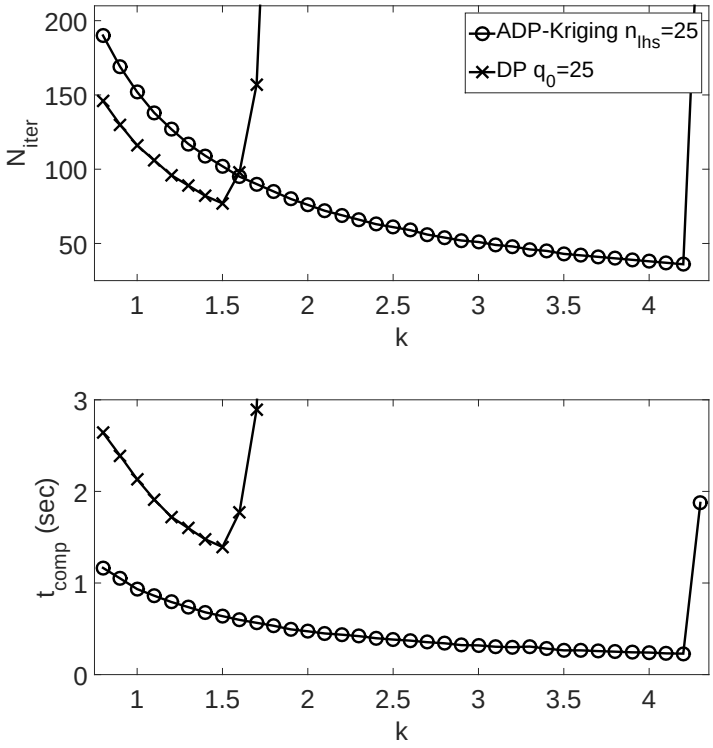

Figure 2.6: VDP oscillator problem (case study 1). Top: number of iterations until convergence vs. gain k. Bottom: computation time (until convergence) vs. gain k.

q_0	10	15	20	25	30	35
t_{comp} (sec)	0.2	0.45	1.2	2.13	3.53	5.8
$\tau(x_0, \tilde{\pi}^*)$	52	58	59	59	59	59

Table 2.1: VDP oscillator problem (case study 1), DP approach. Time to compute DCOC policy ($k = 1$) and first exit-time $\tau(x_0, \tilde{\pi}^*)$ for $x_0 = [1.5, 3, 0]^\top$ for different grids $\tilde{G} = \text{GD}(a, b, [q_0, q_0, q_0]^\top)$.

As can be seen in Figure 2.6, for the DP approach, N_{iter} and t_{comp} decrease with increasing gains until $k = 1.5$ and the approach fails to converge for $k \geq 1.95$. Similarly, the convergence rate improves with increasing k for the ADP approach and convergence is maintained until $k = 4.3$. The observed convergence behavior is different from the theoretical results in Theorem 2.8 and depends on $\tilde{G}$ and $\tilde{V}$ (here, either linear interpolation

or Kriging interpolation) as discussed in Section 2.3.2. The fastest convergence for the DP approach is achieved with $k = 1.5$ (77 iterations), which is 1.5 times faster than with conventional VI ($k = 1$). For the ADP approach, 36 iterations are required when $k = 4.2$, which is 4.2 times faster than with conventional VI.

n_{lhs}	15	25	35	45	55	65
t_{comp} (sec)	0.32	0.52	0.69	0.88	1.23	1.45
$\tau(x_0, \tilde{\pi}^*)$	58	58	58	58	58	59

Table 2.2: VDP oscillator problem (case study 1), ADP-Kriging approach. Time to compute DCOC policy ($k = 1.8$) and first exit-time $\tau(x_0, \tilde{\pi}^*)$ for $x_0 = [1.5, 3, 0]^\top$ for different training sets $\tilde{G} = \text{GD}(a, b, [3, 3, 3]^\top) \cup \text{lhs}(n_{\text{lhs}})$.

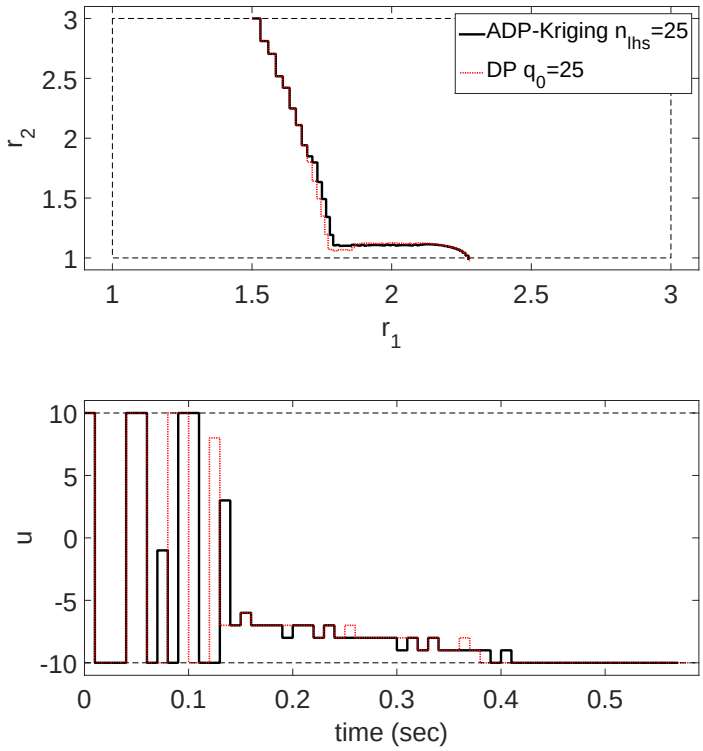

Figure 2.7: VDP oscillator problem (case study 1), $x_0 = [1.5, 3, 0]^\top$. Top: r_1 vs. r_2. Bottom: control u vs. time.

Table 2.1 lists t_{comp} and the corresponding first exit-time for $x_0 = [1.5, 3, 0]^\top$ for the DP approach ($k = 1$) for different grid sizes. The computation time increases exponentially with the grid size, whereas the solution approaches the optimal solution as $\tilde{G}$ becomes

denser in G (i.e., $q_0 \to \infty$). Similarly, Table 2.2 shows t_{comp} and $\tau(x_0, \tilde{\pi}^*)$ for the ADP approach using different $\tilde{G}$. The first exit-times for x_0 are similar to the DP approach while less time is required for computing the ADP-based control policies. Figure 2.7 shows the respective trajectories emanating from x_0 for the DP ($q_0 = 25$) and ADP control policy ($n_{\text{lhs}} = 25$) in the r_1-r_2 plane (top) and the control inputs over time (bottom). The state and control constraints are indicated by dashed lines.

2.6.2.2 LEO Satellite Station Keeping 2

The satellite model (including parameters) is identical to the model used in Section 2.6.1, except for the sampling time which is set to $\Delta t = 2$ sec here. It is assumed that the propulsion system initially carries 40 g of fuel ($m_0 = 4.04$ kg) and the objective is to maximize the time that the satellite stays within ± 20 km of a nominal orbital altitude of 300 km. Hence,

$$G = \{x : r - r_{\text{E}} \in [280, 320]\,\text{km},\ m \geq 4\,\text{kg}\}. \tag{2.73}$$

As explained in Section 2.6.1, the polar angle θ has no effect on the other states and the DCOC problem is independent of θ. Thus, based on (2.70),

$$a = [r_{\text{E}} + 280\,\text{km}, -10\,\text{m/sec}, 0, 0.001147\,\text{rad/sec}, 4\,\text{kg}]^\top,$$

$$b = [r_{\text{E}} + 320\,\text{km}, 10\,\text{m/sec}, 0, 0.001167\,\text{rad/sec}, 4.04\,\text{kg}]^\top,$$

are used to construct a grid. In analogy to the previous example (Section 2.6.2.1), $q = [q_0, q_0, 0, q_0, q_0]^\top$ for the DP approach and $q = [3, 3, 0, 3, 3]^\top$ combined with $\text{lhs}(n_{\text{lhs}})$ for the ADP approach.

Figure 2.8 shows the number of iterations (top) and computation time (bottom) until convergence for different k using the DP ($q_0 = 50$) and ADP approach ($n_{\text{lhs}} = 275$). In line with the previous case studies (Sections 2.6.1.1 and 2.6.2.1), the convergence rate improves with increasing k until k reaches a certain limit. The DP and ADP approach fail to converge for $k \geq 1.5$ and $k \geq 2.1$, respectively. While the computed value functions may differ between the DP and ADP approach, both approaches converge to the same respective approximate value function (within the accuracy prescribed by ε) for the configurations shown in Figure 2.8, independent of k.

The fastest convergence for the DP approach is achieved with $k = 1.2$ (1.2 times faster than conventional VI) and with $k = 2$ for the ADP approach (about twice as fast as conventional VI). For an initial 300 km circular orbit, i.e., $r_0 = r_{\text{E}} + 300\,\text{km}$ and $x_0 = [r_0, 0, 0, \sqrt{\mu/r_0^3}, 4.04\,\text{kg}]^\top$, constraint violation occurs for the first time after about

31 days with the ADP policy compared to 23.7 days when using the DP approach.

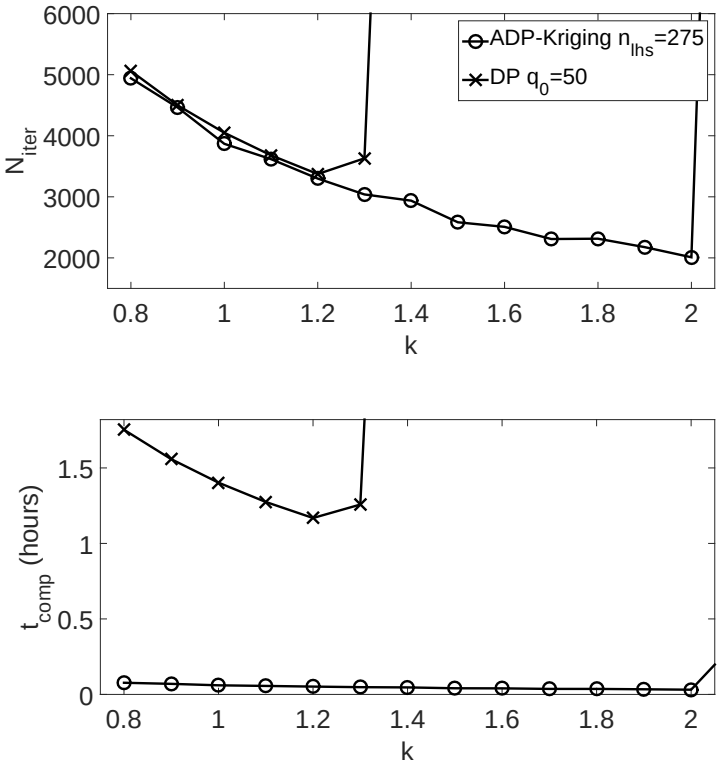

Figure 2.8: LEO satellite station keeping problem (case study 2). Top: number of iterations until convergence vs. gain k. Bottom: computation time (until convergence) vs. gain k.

q_0	30	40	50	60	70
t_{comp} (hours)	0.13	0.52	1.4	3.42	7.85
$t_{\tau(x_0,\tilde{\pi}^*)}$ (days)	2.6	1.7	23.7	26.7	31.1

Table 2.3: LEO satellite station keeping problem (case study 2), DP approach. Time to compute DCOC policy ($k = 1$) and time $t_{\tau(x_0,\tilde{\pi}^*)}$ at constraint violation for initial 300 km circular orbit and different $\tilde{G} = \text{GD}(a, b, [q_0, q_0, 0, q_0, q_0]^\top)$.

A relatively dense grid ($q_0 = 70$) is required for the DP approach to achieve a similar first exit-time performance as the ADP approach. Moreover, the DP approach takes significantly longer to compute the control policy (see bottom of Figure 2.8). This can also be verified in Tables 2.3 and 2.4 that list the respective computation time and first exit-time for x_0 for different grids $\tilde{G}$, where the required computation time of the DP approach increases exponentially with increasing grid size (curse of dimensionality). Hence,

the ADP approach appears to be more suitable for higher-dimensional problems than the DP approach.

n_{lhs}	250	275	300	325	350
t_{comp} (hours)	0.02	0.04	0.04	0.05	0.06
$t_{\tau(x_0,\tilde{\pi}^*)}$ (days)	30.9	31.1	31	31.6	31.4

Table 2.4: LEO satellite station keeping problem (case study 2), ADP-Kriging approach. Time to compute DCOC policy ($k = 1.8$) and time $t_{\tau(x_0,\tilde{\pi}^*)}$ at constraint violation for an initial 300 km circular orbit for $\tilde{G} = \text{GD}(a, b, [3, 3, 0, 3, 3]^\top) \cup \text{lhs}(n_{\text{lhs}})$.

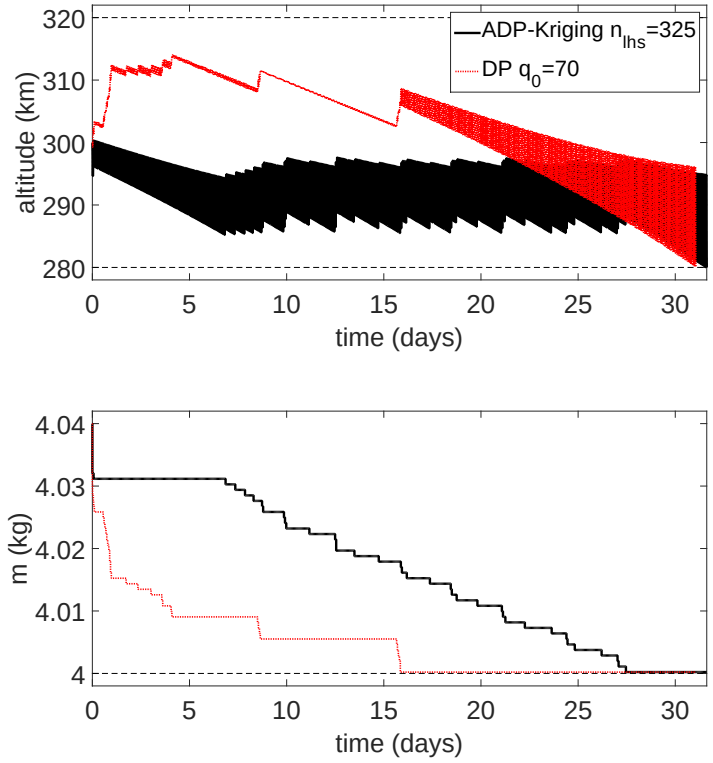

Figure 2.9: LEO satellite station keeping problem (case study 2), initial 300 km circular orbit. Top: altitude vs. time. Bottom: spacecraft mass m vs. time.

The respective trajectories of the satellite's altitude and mass are plotted over time for the DP ($q_0 = 70$) and ADP approach ($n_{\text{lhs}} = 325$) in Figure 2.9, where the constraints are indicated by dashed lines. Compared to the ADP approach, the DP approach converges to a different solution, which may indicate that the optimal control policy is not unique in this problem.

2.6.3 VDP Oscillator 2 and N-S GEO Satellite Station Keeping

Using base-trajectory VI (Section 2.5), two numerical DCOC case studies of maximizing the time until prescribed constraints are violated (i.e., $g \equiv 1$) are discussed in this section. In the first case study (Section 2.6.3.1) a forced VDP oscillator is considered and in the second case study (Section 2.6.3.2) N-S station keeping of a GEO satellite is addressed. In both cases, the discrete-time model is obtained from the respective continuous-time model using Euler's forward method, where the sampling time is denoted by Δt. The convergence threshold for the numerical implementation of base-trajectory VI, see Algorithm 2.3, is set to $\varepsilon = 0.01$.

The approximation of the value function and optimal control policy, obtained with base-trajectory VI, are denoted by V_{dis} and π_{dis}, respectively. Likewise, $V_{\text{dis,VI}}$ and $\pi_{\text{dis,VI}}$ denote the approximate value function and optimal control policy, respectively when using conventional VI. In order to assess the numerical solutions, the following criterion is defined:

$$\Delta V_{\text{dis}}(x) = F_{\text{dis}}(x)/\tau(x, \pi_{\text{dis}}), \tag{2.74}$$

where F_{dis} is given by (2.59). The criterion $\Delta V_{\text{dis,VI}}(x)$ for conventional VI is similarly defined by replacing F_{dis} and π_{dis} with $F_{\text{dis,VI}}$ and $\pi_{\text{dis,VI}}$. According to (2.4), the value function equals the first exit-time when using an optimal control policy. Therefore, a numerical solution is considered to be close to an optimal solution if $\Delta V_{\text{dis}}(x)$ or $\Delta V_{\text{dis,VI}}(x)$, respectively, is close to 1.

Conventional VI and base-trajectory VI are both implemented as a C program in this section (Section 2.6.3). All computations are performed on a computing node with two six-core 2.67 GHz Intel Xeon X5650 processors, i.e., $\nu = 12$, and 48 GB usable RAM. The respective control policies are computed offline and used online for feedback control during the simulations.

2.6.3.1 VDP Oscillator 2

The time-varying discrete-time model of a forced van der Pol oscillator considered in this case study is

$$\begin{aligned} r_{1,t+1} &= r_{1,t} + \Delta t r_{2,t}, \\ r_{2,t+1} &= r_{2,t} + \Delta t\big(\alpha_t \sin(\omega_t t \Delta t) + 2(1 - r_{1,t}^2) r_{2,t} - r_{1,t}\big), \end{aligned} \tag{2.75}$$

where $x_t = [r_{1,t}, r_{2,t}, t]^\top$ denotes the state vector and $u_t = [\alpha_t, \omega_t]^\top$ is the control input at a time instant t. Note that the time instant t is a part of the state vector because the

value function and the control policy depend on time since the system is time-dependent. A sampling time of $\Delta t = 0.001$ sec is used in this example. The state constraints are given by

$$G = \{x : r_1 \in [1, 2], r_2 \in [1, 2]\}. \tag{2.76}$$

Two different cases are treated: first, one control variable; second, two control variables. For the first case, the control variable ω_t is fixed to a constant value $\omega_t \equiv 5$ Hz and $u_t = \alpha_t \in U = [-10, 10]$. Using (2.69), G and U are discretized as follows

$$G_{\text{dis}} = \{x : r_1 \in \text{gd}(1, 2, 4),\ r_2 \in \text{gd}(1, 2, 4),\ t \in \text{gd}(0, 1000, 10)\},$$

$$U_{\text{dis}} = \text{gd}(-10, 10, 21).$$

This discretization of G is referred to as the nominal grid. In order to generate trajectories closer to the optimal solution, another grid on G with 105 equidistant points for each of the states is defined. This discretization is referred to as the dense grid, which also uses $U_{\text{dis}} = \text{gd}(-10, 10, 21)$. Based on Remark 2.8, $\tilde{i}_{\text{th}} = 150$ for the nominal grid and $\tilde{i}_{\text{th}} = 600$ for the dense grid. Note that a larger value of $\tilde{i}_{\text{th}}$ is expected to yield a more accurate approximation of the optimal control policy, but increases computation times.

Figure 2.10 (top) shows example trajectories for an initial state $x_0 = [1, 2, 0]^\top$ using four different control policies: π_{dis} obtained by base-trajectory VI for the nominal grid (solid black line); $\pi_{\text{dis,VI}}$ obtained by conventional VI for the nominal grid (dashed-dotted green line); π_{dis} obtained by base-trajectory VI for the dense grid (dashed red line); $\pi_{\text{dis,VI}}$ obtained by conventional VI for the dense grid (dotted blue line). The bottom plot in Figure 2.10 displays the approximated value function F_{dis} or $F_{\text{dis,VI}}$, respectively, over time. It can be seen that the solution of base-trajectory VI performs slightly better than the solution of conventional VI for the nominal grid. In particular, the approximation of the value function obtained by base-trajectory VI is closer to the actual value function, which decreases by one at each time instant according to (2.6).

The slightly better performance of base-trajectory VI for the nominal grid is also verified in Table 2.5, which shows the performance criterion ΔV defined in (2.74) and the first exit-time for the initial state vector $[1, 2, 0]^\top$, where the solution of base-trajectory VI generates a ΔV closer to one and satisfies the prescribed constraints two time steps longer than the solution by conventional VI, i.e., $\tau(x, \pi_{\text{dis}}) = 627$ and $\tau(x, \pi_{\text{dis,VI}}) = 625$. The time to compute the control policy with base-trajectory VI (run Algorithm 2.3 for all $x \in G_{\text{dis}}$) is 0.5 sec compared to 0.2 sec for conventional VI as seen in Table 2.5.

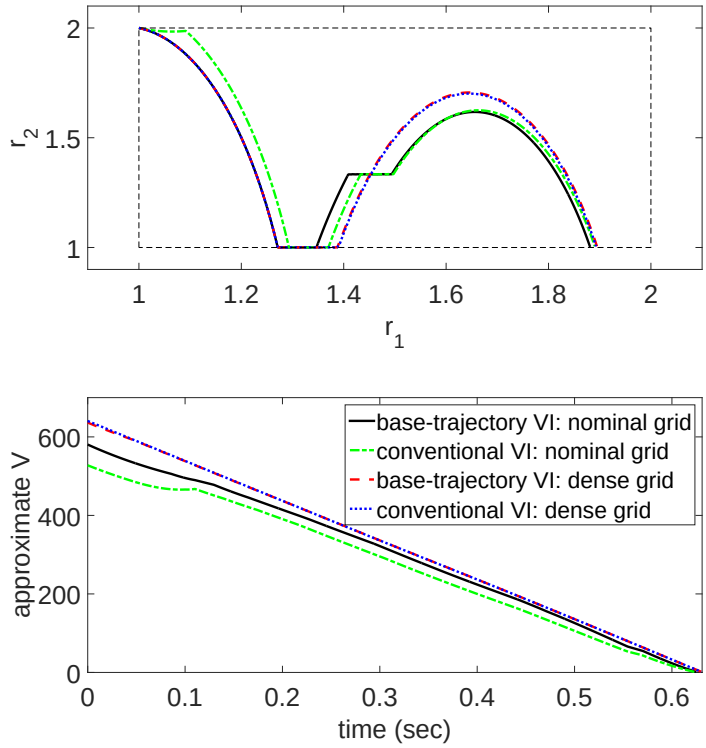

Figure 2.10: VDP oscillator problem (case study 2) – one control variable, $x_0 = [1, 2, 0]^\top$. Top: state trajectory in r_1-r_2 plane. Bottom: approximation of the value function vs. time.

	$\Delta V_{\text{dis}}(x)$ or $\Delta V_{\text{dis,VI}}(x)$	$\tau(x, \pi_{\text{dis}})$ or $\tau(x, \pi_{\text{dis,VI}})$	comp. time
base-trajectory VI, nominal grid	0.926	627	0.5 sec
conventional VI, nominal grid	0.844	625	0.2 sec
base-trajectory VI, dense grid	1.007	632	10,834 sec
conventional VI, dense grid	1.014	632	143 sec

Table 2.5: VDP oscillator problem (case study 2) with one control variable. $\Delta V(x)$ according to (2.74) and time steps until constraint violation $\tau(x, \pi)$ for $x = [1, 2, 0]^\top$ as well as required time (wall time) to compute the respective control policies on G.

For the dense grid, the solutions of base-trajectory VI and conventional VI are nearly identical. According to Table 2.5, both solutions violate the constraints for the first time after 632 steps and the respective performance criterion is close to one, indicating that the solutions are close to an optimal solution. Base-trajectory VI requires more time to

compute the control policy for the dense grid (see Table 2.5), which can be reduced by lowering $\tilde{i}_{\text{th}}$.

Now consider the case of two control variables, where, in addition to α, ω serves as a control variable. In this case, the control constraints are given by

$$U = \{u : \alpha \in \{-10, 0, 10\}, \omega \in \{0, 0.5, 1, ..., 5\} \text{ Hz}\},$$

and $U_{\text{dis}} = U$ is used. The nominal discretization of G for this case is chosen as follows:

$$G_{\text{dis}} = \{x : r_1 \in \text{gd}(1, 2, 15),\ r_2 \in \text{gd}(1, 2, 15),\ t \in \text{gd}(0, 1000, 30)\},$$

and $\tilde{i}_{\text{th}} = 100$ is set for base-trajectory VI (see Remark 2.8). Furthermore, another discretization referred to as the dense grid is constructed with $\text{gd}(0, 1000, 251)$ for t and 35 non-equidistant points for each r_1 and r_2. The focus of the non-equidistant points for r_1 and r_2 is on the area where $r_1 \geq 1.25$ and $r_2 \leq 1.15$ since most optimal trajectories spend a considerable amount of time there. When the dense grid is used, $\tilde{i}_{\text{th}}$ is set to 750.

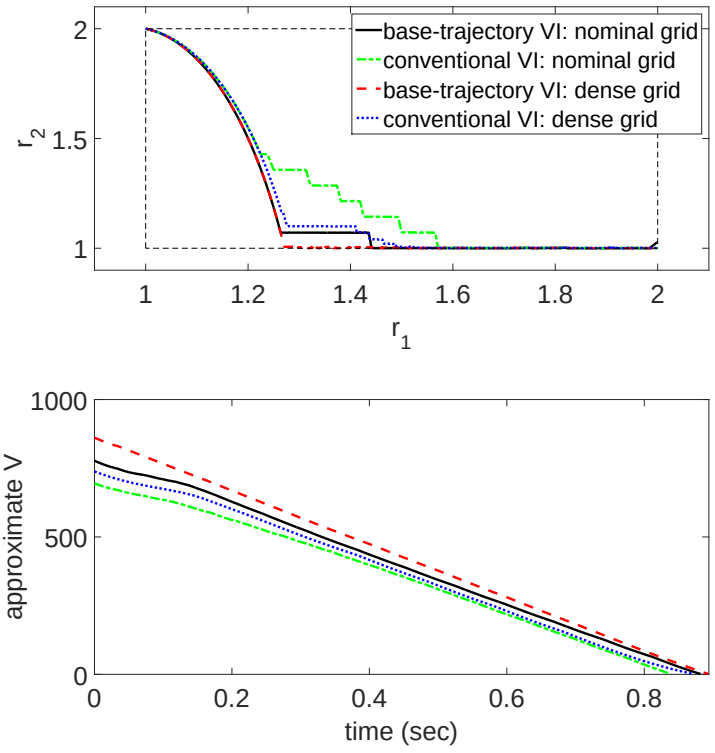

Figure 2.11: VDP oscillator problem (case study 2) – two control variables, $x_0 = [1, 2, 0]^\top$. Top: state trajectory in r_1-r_2 plane. Bottom: approximation of the value function vs. time.

Figure 2.11 shows the state trajectories as well as the approximated value function vs. time for $x_0 = [1, 2, 0]^\top$ and different control policies. Table 2.6 lists the corresponding performance criteria ΔV, the first exit-times from G, and the times required to compute the control policies. The nominal grid solution of base-trajectory VI shows better performance compared to both the nominal and dense grid solutions by conventional VI. This is because the DCOC problem and the shape of the value function become more complex with two control variables, increasing the interpolation error, which is greater for conventional VI as described in Section 2.5.2. The nominal grid solution of base-trajectory VI satisfies the constraints for 885 time steps, whereas only 841 (nominal grid) and 875 (dense grid) time steps are achieved with conventional VI. With 38 sec, base-trajectory VI requires more computation time for computing the control policy than conventional VI on the nominal grid (6 sec) and less time than conventional VI on the dense grid (67 sec).

	$\Delta V_{\text{dis}}(x)$ or $\Delta V_{\text{dis,VI}}(x)$	$\tau(x, \pi_{\text{dis}})$ or $\tau(x, \pi_{\text{dis,VI}})$	comp. time
base-trajectory VI, nominal grid	0.879	885	38 sec
conventional VI, nominal grid	0.828	841	6 sec
base-trajectory VI, dense grid	0.961	896	9,842 sec
conventional VI, dense grid	0.844	875	67 sec

Table 2.6: VDP oscillator problem (case study 2) with two control variables. $\Delta V(x)$ according to (2.74) and time steps until constraint violation $\tau(x, \pi)$ for $x = [1, 2, 0]^\top$ as well as required time (wall time) to compute the respective control policies on G_{dis}.

Base-trajectory VI applied to the dense grid generates the solution closest to an optimal solution as indicated by the approximation of the value function plotted over time (bottom plot in Figure 2.11) and ΔV_{dis} in Table 2.6. For the considered initial condition, constraint violation occurs for the first time after 896 time steps. However, the computation time for obtaining the control policy is considerably longer than for the other policies. The computation time of base-trajectory VI can be reduced to some extent by lowering $\tilde{i}_{\text{th}}$, which may decrease the performance of the control policy. In general, there is a trade-off between computation time and performance/accuracy of the control policy. As shown in this case study and in the subsequent one (Section 2.6.3.2), base-trajectory VI applied to a sparse grid (nominal grid) provides the best balance between computation time and control performance.

2.6.3.2 N-S Station Keeping of a GEO Satellite

In this case study, a DCOC problem of N-S (out of orbital plane) GEO satellite station keeping is treated, where the objective is to satisfy position and fuel constraints for as long as possible. When the satellite is sufficiently close to the nominal orbit its motion is accurately described by the linear Clohessy-Wiltshire (CW) equations [65], where the out-of-plane (orbital plane) motion is decoupled from the in-plane motion. Thus, most GEO station keeping approaches use separate control strategies for the N-S and E-W directions [45, 70, 71]. Based on the CW equations, the following model for the N-S dynamics of the GEO satellite is used,

$$\begin{aligned} r_{t+1} &= r_t + \Delta t v_t, \\ v_{t+1} &= v_t + \Delta t \left(-n_0^2 r_t + u_t + a_{\mathrm{p}}(t\Delta t)\right), \\ \mathrm{fuel}_{t+1} &= \mathrm{fuel}_t - \Delta t |u_t| / u_{\mathrm{norm}}, \end{aligned} \tag{2.77}$$

where r and v are the out-of-plane position and velocity of the satellite relative to the nominal orbit, $\mathrm{fuel} \in \mathbb{Z}_{\geq 0}$ is a normalized variable indicating the amount of fuel available for maneuvering, and $n_0 = 7.3 \times 10^{-5}\,\mathrm{rad/sec}$ is the angular rate of the nominal geostationary orbit. The state vector at a time instant $t \in \mathbb{Z}_{\geq 0}$ is $x_t = [r_t, v_t, \mathrm{fuel}_t, t]^\top$. A 4000 kg satellite, equipped with a 0.2 N on/off-thruster for each of the north and south directions, is considered, i.e., $u_t \in U = \{-0.00005, 0, 0.00005\}\ \mathrm{m/sec^2}$. The state constraints for this problem are given by

$$G = \{x : r \in [-7.4, 7.4]\,\mathrm{km},\ v \in [-0.65, 0.65]\,\mathrm{m/sec},\ \mathrm{fuel} \geq 0\}, \tag{2.78}$$

where the position constraint is equivalent to a latitude window of $\pm 0.01\,\mathrm{deg}$.

The term $a_{\mathrm{p}}(t\Delta t)$ in (2.77) describes time-varying orbital perturbations. Here, perturbations due to solar radiation pressure, J2 (perturbation due to Earth's non-spherical shape), and the gravity of the Moon and Sun are taken into account. Given the state constraints, the satellite's trajectory, which is unknown in advance, is sufficiently close to the trajectory of the nominal orbit such that $a_{\mathrm{p}}(t\Delta t)$ can be computed in advance for all t based on the known nominal orbit. More information on how to obtain $a_{\mathrm{p}}(t\Delta t)$ can be found in [45]. For this problem, $t = 0$ corresponds to September 3, 2015, at 12 am (CT) with the position of the nominal orbit being on the x-axis of the Earth-centered inertial (ECI) coordinate frame [64].

A sampling time of $\Delta t = 10$ sec is used. However, for computing the control policies, larger time increments of 1000 sec are employed by using the state transition formula

for linear discrete-time systems as in [45], i.e., $t \in \{0, 100, 200, ...\}$. This provides a compromise between performance/accuracy of the controller and computation time. During the closed-loop simulations ($\Delta t = 10$ sec), the control value u_t is computed every 1000 sec while u_t is applied continuously throughout the subsequent 1000 sec. Moreover, $u_{\text{norm}} = \Delta t 0.005\,\text{m/sec}^2$ in (2.77), i.e., $\text{fuel}_t - \text{fuel}_{t+100} = 1$ each time a nonzero control is applied.

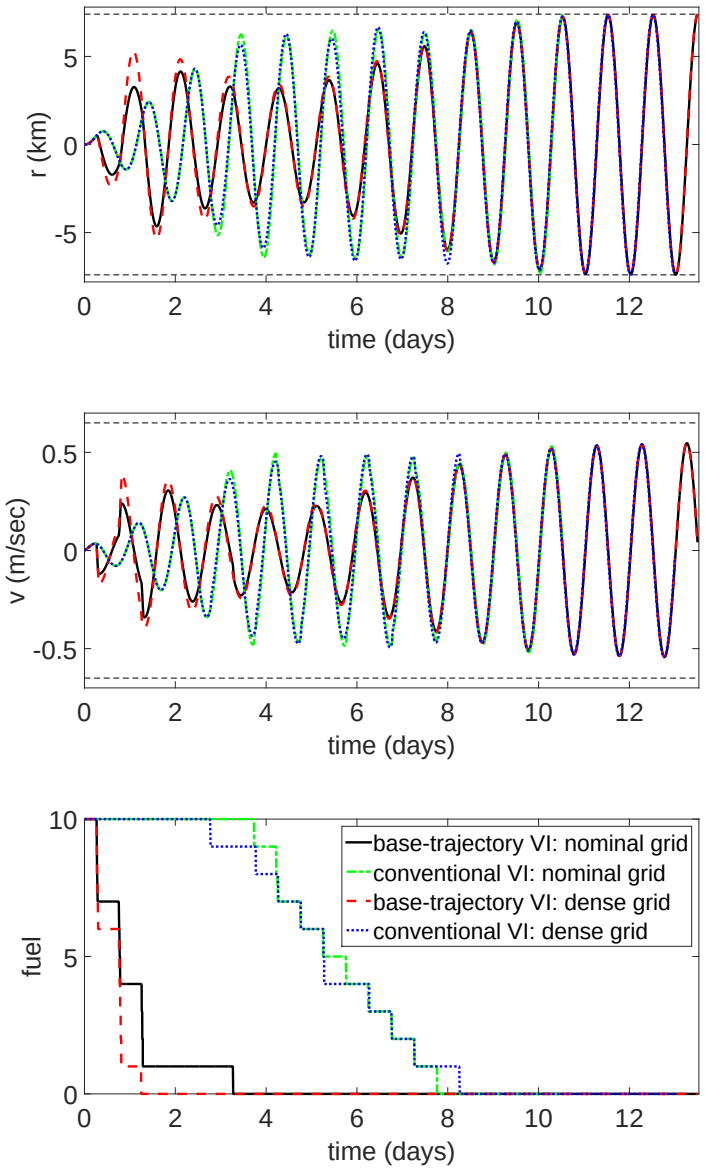

Figure 2.12: N-S GEO satellite station keeping problem, $x_0 = [0, 0, 10, 0]^\top$. Position r (top) and velocity v (middle) relative to nominal orbit vs. time as well as fuel (bottom) vs. time.

For G_{dis}, $r \in \text{gd}(-7.4\,\text{km}, 7.4\,\text{km}, 75)$, $v \in \text{gd}(-0.65\,\text{m/sec}, 0.65\,\text{m/sec}, 75)$, $\text{fuel} \in \{0, 1, ..., 10\}$, and $t \in \text{gd}(0, 120000, 500)$ are chosen as the nominal grid. Similarly, a dense grid with 251 equidistant points for each r and v, 2500 equidistant points for t, and $\text{fuel} \in \{0, 1, ..., 10\}$ is constructed. Moreover, $U_{\text{dis}} = U$. For base-trajectory VI, $\tilde{i}_{\text{th}} = 30$ for both the nominal and the dense grid. Note that base-trajectory VI converges in 11 iterations due to the exact discretization of the variable fuel. This is not true for conventional VI (1920 and 1837 iterations until convergence for the nominal and dense grid, respectively) because, in contrast to base-trajectory VI, $V_n(x) = V_{n+k}(x)$ does not necessarily hold for each $x \in G_{\text{dis}}$ with $\text{fuel} = n$ and $k \in \mathbb{Z}_+$.

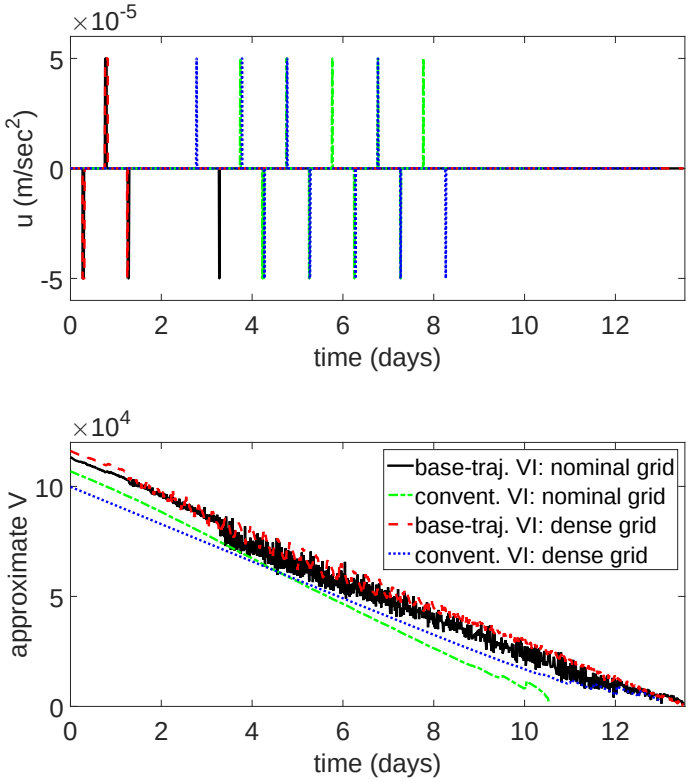

Figure 2.13: N-S GEO satellite station keeping problem, $x_0 = [0, 0, 10, 0]^\top$. Control input (top) and approximation of the value function (bottom) vs. time.

Figures 2.12 and 2.13 show the results for an initial $x_0 = [0, 0, 10, 0]^\top$, i.e., 10 nonzero control actions can be applied without violating the fuel constraint. Table 2.7 lists the corresponding ΔV and τ as well as computation times. As in the previous example, base-trajectory VI performs better than conventional VI due to smaller interpolation errors. While the computation times of base-trajectory VI and conventional VI are comparable, constraints are violated after only 10.53 days (nominal grid) and 13 days (dense grid) when

using conventional VI.

Both solutions of base-trajectory VI violate the constraints around the same time: 13.5 days for the nominal grid and 13.49 days for the dense grid, indicating that the optimal exit-time is around 13.5 days here. The slightly better result (longer time until constraint violation) for the nominal/sparser grid may be attributed to the fact that convergence to the value function is not uniform, see Theorem 2.10, and depends on G_{dis} for the discretized problem. For both base-trajectory VI and conventional VI, it is expected that the respective control policy approaches an optimal control policy pointwise as G_{dis} becomes denser in G. As can be seen in Figure 2.12 (bottom) and Figure 2.13 (top), the control policies obtained by the two algorithms are different. Base-trajectory VI yields policies that use all of the available fuel in the beginning, whereas the policies obtained by conventional VI wait about 3 days before starting to use the available fuel. This suggests that the optimal control policy for this problem may not unique.

	$\Delta V_{\mathrm{dis}}(x)$ or $\Delta V_{\mathrm{dis,VI}}(x)$	$\tau(x, \pi_{\mathrm{dis}})$ or $\tau(x, \pi_{\mathrm{dis,VI}})$	comp. time
base-trajectory VI, nominal grid	0.97	116650	1871 sec
conventional VI, nominal grid	1.173	90967	959 sec
base-trajectory VI, dense grid	0.996	116596	67530 sec
conventional VI, dense grid	0.887	112356	64739 sec

Table 2.7: N-S GEO satellite station keeping problem. $\Delta V(x)$ according to (2.74) and time steps until constraint violation $\tau(x, \pi)$ for $x = [0, 0, 10, 0]^\top$ as well as required time (wall time) to compute the respective control policies on G_{dis}.

2.6.4 Spacecraft Attitude Control

Base-trajectory VI (Section 2.5) is applied to attitude control of an axisymmetric spacecraft/rocket during a translational thrusting maneuver with a fixed thrust vector misalignment. The spacecraft's mass and inertia properties in this case study are time-varying due to the mass flow required for the orbital maneuver. Given fuel constraints for the attitude control system, the objective is to counteract the parasitic moment resulting from the thrust vector misalignment and to maximize the time during which the orientation of the spacecraft symmetry axis stays within a prescribed cone. Numerical results are presented in this section, including a robustness analysis of the DCOC-based controller with respect to uncertainties in the thrust vector misalignment.

2.6.4.1 Model Formulation

The general rotational dynamics of a rigid body with time-varying mass and inertia properties are derived in Appendix A. Equation (A.12) in Appendix A states the governing equation for a rigid body $\mathcal{B}$ with center of mass c using a body-fixed point z. The specific spacecraft model considered in this case study is outlined in Fig. 2.14. A cylindrical spacecraft with four sections is considered. The first section comprises the main engine with evenly distributed mass m_{e}. The engine is followed by the tanks of the oxidizer (ox) and the fuel (f). Both the oxidizer and the fuel have initial masses $m_{\mathrm{ox},0}$ and $m_{\mathrm{f},0}$, respectively, which decrease at constant mass flow rates $\dot{m}_{\mathrm{ox}}$ and $\dot{m}_{\mathrm{f}}$, respectively. The slosh dynamics in the tanks are neglected. The fourth section of the rocket is the payload with evenly distributed mass m_{p}.

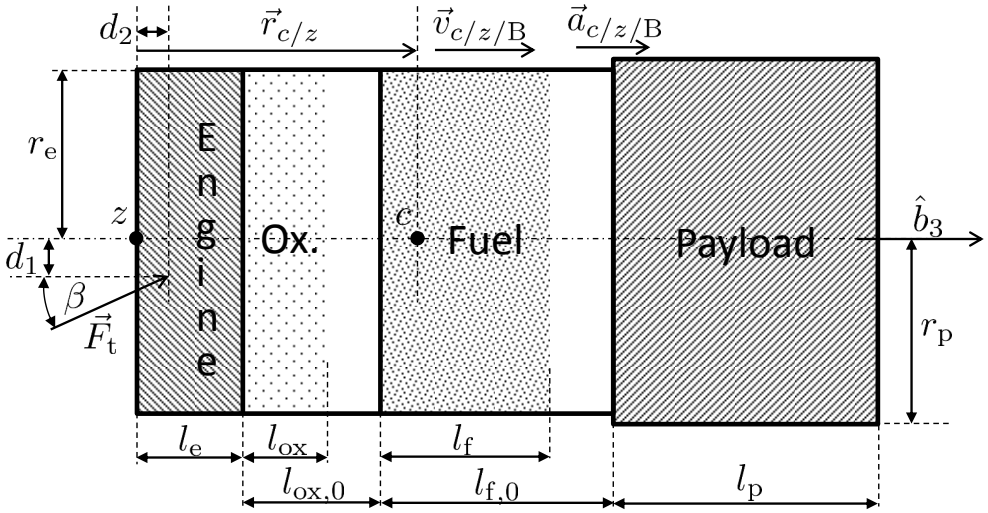

Figure 2.14: Spacecraft attitude control problem: model of an axisymmetric spacecraft.

The governing equations are resolved in the body-fixed principal frame $\mathrm{F_B}$ with mutually perpendicular frame vectors $(\hat{b}_1, \hat{b}_2, \hat{b}_3)$, where $\hat{b}_3$ is the symmetry axis of the spacecraft. As shown in Figure 2.14, there is a thrust force $\vec{F}_{\mathrm{t}}$ acting on the spacecraft that is misdirected by an angle β relative to the symmetry axis. Moreover, the thrust force is displaced relative to the point z by a radial distance d_1 and longitudinal distance d_2. As derived in Appendix A, the equations describing the rotational dynamics of the spacecraft are

$$\begin{aligned}\dot{\omega}_1 =& \frac{1}{J_{\mathrm{T}}(t) - r_3^2(t) m_{\mathcal{B}}(t)} \Big[\Big(2 r_3(t) v_3(t) m_{\mathcal{B}}(t) + r_3^2(t) \dot{m}_{\mathcal{B}} - \dot{J}_{\mathrm{T}}(t) \Big) \omega_1 \\ & + u_1 + M_1 + \big(J_{\mathrm{T}}(t) - J_{\mathrm{R}}(t) - r_3^2(t) m_{\mathcal{B}}(t) \big) \omega_2 \omega_3 - r_3(t) f_2 \Big],\end{aligned} \tag{2.79}$$

$$\begin{aligned}\dot{\omega}_2 =& \frac{1}{J_{\mathrm{T}}(t) - r_3^2(t) m_{\mathcal{B}}(t)} \Big[\Big(2 r_3(t) v_3(t) m_{\mathcal{B}}(t) + r_3^2(t) \dot{m}_{\mathcal{B}} - \dot{J}_{\mathrm{T}}(t) \Big) \omega_2 \\ & + u_2 + M_2 + \big(J_{\mathrm{R}}(t) - J_{\mathrm{T}}(t) + r_3^2(t) m_{\mathcal{B}}(t) \big) \omega_1 \omega_3 + r_3(t) f_1 \Big],\end{aligned} \tag{2.80}$$

where ω_1 and ω_2 are the spacecraft's angular velocity vector projections on its body-fixed axes $\hat{b}_1$ and $\hat{b}_2$, respectively. The angular velocity projection $\omega_3 = \omega_3(t)$ on the symmetry axis $\hat{b}_3$ can be obtained explicitly as a function of time, see (A.29). The time-dependent parameters $r_3(t)$ and $v_3(t)$ are the position and velocity of the center of mass c relative to point z, see (A.19) and (A.20). Equation (A.14) describes the time-varying mass of the spacecraft $m_{\mathcal{B}}(t)$. The time-varying components of the moment of inertia $J_{\mathrm{R}}(t)$, $J_{\mathrm{T}}(t)$, and its derivative $\dot{J}_{\mathrm{T}}(t)$ are given by (A.22), (A.23), and (A.25), respectively. The external force acting along the $\hat{b}_1$-axis is f_1 and likewise f_2 is acting along the $\hat{b}_2$-axis. The external moments around the two body-fixed axes are M_1 and M_2 ($M_3 = 0$ is assumed). There are two control inputs $u_1 \in \{-\alpha, 0, \alpha\}$ and $u_2 \in \{-\alpha, 0, \alpha\}$ which are the control moments around the $\hat{b}_1$- and $\hat{b}_2$-axis, respectively.

In order to describe the attitude kinematics, a parametrization introduced by Tsiotras and Longuski [72] is used. The orientation of the 3-axis of the inertial reference frame $\mathrm{F_A}$, denoted by $\hat{a}_3$, expressed in the body-fixed frame $\mathrm{F_B}$, is described by the two variables θ_1 and θ_2,

$$\theta_1 = \frac{b}{1+c}, \quad \theta_2 = -\frac{a}{1+c}, \tag{2.81}$$

where a, b, and c are the components of $\hat{a}_3$ expressed in the body-fixed frame, i.e., $\hat{a}_3 = a\hat{b_1} + b\hat{b}_2 + c\hat{b}_3$. Note that θ_1 is the real part and θ_2 is the imaginary part of a complex variable θ which results from the stereographic projection $\sigma : \mathcal{S}^2 \setminus \{0, 0, -1\} \to \mathbb{C}$, where $\mathcal{S}^2$ denotes the surface of the unit sphere in $\mathbb{R}^3$. Using θ_1 and θ_2 for attitude representation, the kinematic equations are given by [72]

$$\dot{\theta}_1 = \omega_3 \theta_2 + \omega_2 \theta_1 \theta_2 + \frac{\omega_1}{2} \left(1 + \theta_1^2 - \theta_2^2 \right), \tag{2.82}$$

$$\dot{\theta}_2 = -\omega_3 \theta_1 + \omega_1 \theta_1 \theta_2 + \frac{\omega_2}{2} \left(1 + \theta_2^2 - \theta_1^2 \right). \tag{2.83}$$

The available propellant mass for the spacecraft's attitude control system is denoted by m. The differential equation describing m is as follows

$$\dot{m} = -c_m \left(|u_1| + |u_2| \right), \tag{2.84}$$

where $c_m > 0$ is a constant. In summary, the time-dependent system is described by five states: $x = [\omega_1, \omega_2, \theta_1, \theta_2, m]^\top$. There are two control inputs u_1 and u_2. The governing equations are given by (2.79), (2.80), (2.82), (2.83), and (2.84). The continuous-time model

is converted into a discrete-time formulation using Euler's forward method,

$$x_{k+1} = x_k + \left[\dot{\omega}_1(t_k), \dot{\omega}_2(t_k), \dot{\theta}_1(t_k), \dot{\theta}_2(t_k), \dot{m}(t_k)\right]^\top \Delta t, \tag{2.85}$$

where a sampling time of $\Delta t = 0.3$ sec is used in this case study.

2.6.4.2 Model Parameters

The engine characteristics and parameters of the spacecraft are similar to the A-4 engine [73], which uses Hydrazine as fuel and Dinitrogen Tetroxide as the oxidizer. The engine generates a thrust of $F_\mathrm{t} = 33360$ N with $I_\mathrm{sp} = 320$ sec and oxidizer and fuel mass flow rates of $\dot{m}_\mathrm{ox} = 5.8$ kg/sec and $\dot{m}_\mathrm{f} = 4.83$ kg/sec, respectively. With an engine mass of $m_\mathrm{e} = 117$ kg, initial fuel and oxidizer masses of $m_{\mathrm{f},0} = 2273$ kg and $m_{\mathrm{ox},0} = 2727$ kg, and a payload mass of $m_\mathrm{p} = 8000$ kg, the total wet mass of the spacecraft is $m_{\mathcal{B},0} = 13117$ kg. The burn time for the orbital maneuver is $T = 200$ sec. Note that this maneuver generates an increase in velocity of $\Delta v = 0.56$ km/sec. For this example, a nominal thrust vector misalignment of $\beta = 0.1$ deg and $d_1 = 2$ mm are assumed. Moreover, $d_2 = 1$ m is chosen (see Figure 2.14). Thus, the components of the external moment relative to point z are

$$M_1 = d_1 F_\mathrm{t} \cos(\beta) = 66.7 \text{ Nm}, \quad M_2 = d_2 F_\mathrm{t} \sin(\beta) = 58.2 \text{ Nm}, \quad M_3 = 0. \tag{2.86}$$

The components of the external force acting on the spacecraft are

$$f_1 = -F_\mathrm{t} \sin(\beta) = -58.2 \text{ N}, \quad f_2 = 0 \text{ Nm}, \quad f_3 = F_\mathrm{t} \cos(\beta) = 33359.9 \text{ N}. \tag{2.87}$$

The attitude control system for the axes $\hat{b}_1$ and $\hat{b}_2$ comprises eight R-4D thrusters [74]. Each thruster can generate a force of 490 N with an I_sp of 312 sec. With an effective lever of 2.86 m, each pair of R-4D thrusters generates a moment of 1401 Nm about the respective axis ($\hat{b}_1$ or $\hat{b}_2$) relative to point z. Thus, the control inputs u_1 and u_2 take values from the set $\{-1401, 0, 1401\}$ Nm. The constant that describes the propellant consumption of the attitude control system is $c_m = 2/I_\mathrm{sp}/g/2.86\,\mathrm{m} = 2.285 \times 10^{-4}$ sec/m^2. The remaining parameters of the spacecraft are summarized in the following table.

$l_\mathrm{p} = 3$ m	$l_{\mathrm{ox},0} = 0.96$ m	$l_\mathrm{e} = 1.75$ m	$l_{\mathrm{f},0} = 1.15$ m
$r_\mathrm{E} = 0.8$ m	$r_\mathrm{p} = 1.5$ m	$\rho_\mathrm{ox} = 1456$ kg/m^3	$\rho_\mathrm{f} = 1013$ kg/m^3

Table 2.8: Spacecraft attitude control problem: spacecraft parameters.

2.6.4.3 DCOC Problem and State Space Discretization

The state constraints are described by

$$G = \left\{ x \in \mathbb{R}^5 : \ \sqrt{\theta_1^2 + \theta_2^2} \leq \theta_{\text{limit}}, \ m \geq 0 \right\}. \tag{2.88}$$

The constraint on the orientation of the spacecraft symmetry axis in (2.88) defines a cone, where a half angle of 0.5 deg is chosen here. This corresponds to $\theta_{\text{limit}} = 0.004363$ according to (2.81). The minimum propellant mass for the attitude control system is $m = 0$.

The following nominal discretization is used for the numerical implementation of base-trajectory VI (Algorithm 2.3),

$$\begin{aligned} G_{\text{dis}} = \Big\{ & x \in G : \omega_1, \omega_2 \in \text{gr}(-0.86\,\text{deg/sec}, 0.86\,\text{deg/sec}, 29), \\ & \theta_1, \theta_2 \in \text{gd}(-0.004363, 0.004363, 17), \ m \in \text{gd}(0, 15.27\,\text{kg}, 160) \Big\}, \end{aligned} \tag{2.89}$$

where $\text{gd}(...)$ is given by (2.69). In addition, the time horizon is discretized by $t \in T_{\text{dis}}$, where

$$T_{\text{dis}} = \text{gd}(0, 200\,\text{sec}, 40). \tag{2.90}$$

In addition to the nominal discretization in (2.89) and (2.90), the influence of different discretization choices on the solution is analyzed in the following.

2.6.4.4 Results

Base-trajectory VI (Algorithm 2.3) with linear interpolation in (2.56) is employed to approximate V at the points of the discretized state and time spaces. Based on Remark 2.8, $i_{\text{th}}(x) = 8$ is used for all x with $m \geq 0.77$ kg. The sequence of functions is initialized as $V_0(x) = 1$ for all $x \in \mathcal{K}_0 \cap G_{\text{dis}}$, where $\mathcal{K}_0$ is defined in (2.27), and $V_0(x) = 2$ otherwise. The method is implemented as a C program, where all simulations in this section (Section 2.6.4) are run on a desktop computer with an Intel Core i7-3770 processor and 15.8 GB usable memory. The quality of a numerical solution is assessed with the ΔV_{dis} criterion, defined in (2.74), where $\Delta V_{\text{dis}} = 1$ for an optimal solution.

A computation time of 439 sec is required to solve the DCOC problem numerically. Figure 2.15 shows the simulation results for an initial condition of $\omega_{0,1} = \omega_{0,2} = \theta_{0,1} = \theta_{0,2} = 0$, $m_0 = 15.27$ kg, and $t_0 = 0$. The spacecraft stays inside the prescribed set G during the entire maneuver time of 200 sec. This can be seen in Figure 2.15 (top), which shows the trajectory of the attitude parameters in the complex plane, including the boundary θ_{limit} (red circle), as well as the propellant mass. The remaining propellant mass at the end

is 0.672 kg. The approximation of the value function is linearly decreasing in time as seen in Figure 2.15 (bottom, right) and $\Delta V_{\text{dis}} = 0.9954$, suggesting that the numerical solution is close to an optimal solution (based on necessary conditions). The control input is plotted in Figure 2.15 (bottom left).

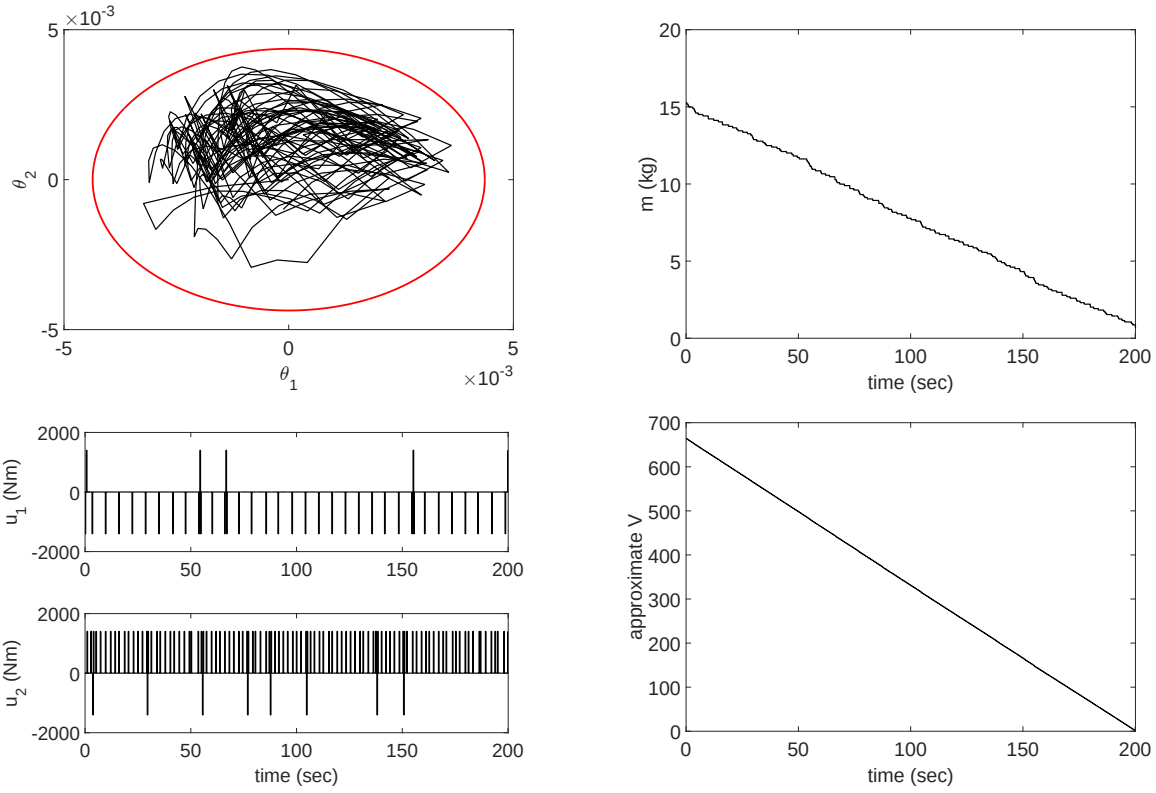

Figure 2.15: Spacecraft attitude control problem for nominal disturbance [see (2.86) and (2.87)], nominal grid [see (2.89)], and initial condition $\omega_{0,1} = \omega_{0,2} = \theta_{0,1} = \theta_{0,2} = 0$, $m_0 = 15.27$ kg. Top: attitude parameters in complex plane (left) and propellant mass m vs. time (right). Bottom: control moments u_1 and u_2 vs. time (left) and approximation of value function vs. time (right).

The influence of the state and time space discretization is investigated by defining two additional discretizations. In addition to the nominal discretization in (2.89) with $n_\omega = 29$ grid points for ω_1 and ω_2, $n_\theta = 17$ grid points for θ_1 and θ_2, and $n_t = 40$ grid points for the time, a dense discretization with $n_\omega = 32$, $n_\theta = 20$, and $n_t = 45$ as well as a sparse discretization with $n_\omega = 25$, $n_\theta = 14$, and $n_t = 36$ are defined. Table 2.9 compares the remaining mass at $t = 200$ sec, the criterion ΔV_{dis}, and the computation time for the three discretizations: sparse, nominal, and dense. The computation time increases exponentially with the density of the discretization (curse of dimensionality). The sparse grid requires 183 sec in contrast to the dense grid with 909 sec. However, the accuracy of the solution

improves with the grid density. The sparse grid solution is further away from the optimum, whereas the solution of the dense grid is closer to being optimal (based on ΔV_{dis}). This is also reflected in the remaining mass after 200 sec, where the solution of the sparse grid has no propellant left after 200 sec (however not violating the constraints). Both the solutions for the nominal and the dense grid have 0.672 kg propellant left at the end.

Discretization	$m(t = 200\,\text{sec})$	ΔV_{dis}	Computational time
sparse: $n_\omega = 25$, $n_\theta = 14$, $n_t = 36$	0.0 kg	0.986	183 sec
nominal: $n_\omega = 29$, $n_\theta = 17$, $n_t = 40$	0.672 kg	0.9954	439 sec
dense: $n_\omega = 32$, $n_\theta = 20$, $n_t = 45$	0.672 kg	0.9988	909 sec

Table 2.9: Spacecraft attitude control problem: influence of state and time space discretization on the simulation results for nominal disturbance [see (2.86) and (2.87)] and $\omega_{0,1} = \omega_{0,2} = \theta_{0,1} = \theta_{0,2} = 0$, $m_0 = 15.27$ kg.

2.6.4.5 Robustness Analysis

The results in the previous section (Section 2.6.4.4) are based on exact knowledge of the disturbances f_1, f_2, M_1, and M_2. Now the robustness of the solution is analyzed with respect to uncertainties in the disturbances. The control policy based on the nominal approximation of V, obtained on the nominal grid [see (2.89)] and assuming the disturbance values in (2.86) and (2.87), is used. However, the actual values of the disturbances are increased/decreased from the nominal case by 10 %, 25 %, and 50 %.

Table 2.10 shows the differences in the solution for the initial condition $\omega_{0,1} = \omega_{0,2} = \theta_{0,1} = \theta_{0,2} = 0$, $m_0 = 15.27$. It can be seen that there are no significant differences in the solutions for a 10 % uncertainty in the disturbances. The solutions for the 10 % uncertainty case yield the same $\Delta V_{\text{dis}} = 0.9954$ as the nominal solution. For a 25 % difference in actual disturbance values, however, the solutions start to deviate from the nominal solution, yielding $\Delta V_{\text{dis}} = 1.031$ (for 25 % increase) and $\Delta V_{\text{dis}} = 1.0742$ (for 25 % decrease). Moreover, the constraints are violated before the end of the 200 sec maneuver. An uncertainty of 50 % in the disturbances results in further performance degradation (based on ΔV_{dis} and first exit-time). An interesting case is when the actual disturbances are zero ($f_1 = f_2 = M_1 = M_2 = 0$). For this case and the particular initial condition, it would be intuitive not to apply any control action. However, the controller is based on the nominal disturbance values and tries to steer the system to a supposedly more efficient region. This is certainly not optimal in this case and the constraints are violated after 145.8 sec with $\Delta V_{\text{dis}} = 1.3647$.

Disturbances	$t_{\tau(x,\pi_{\text{dis}})}$	$m(t_{\tau(x,\pi_{\text{dis}})})$ or $m(t = 200\,\text{sec})$	ΔV_{dis}
nominal	> 200 sec	0.672 kg	0.9954
10 % increase from nominal	> 200 sec	0.769 kg	0.9954
10 % decrease from nominal	> 200 sec	0.769 kg	0.9954
25 % increase from nominal	193.2 sec	0.0 kg	1.031
25 % decrease from nominal	185.4 sec	0.0 kg	1.0742
50 % increase from nominal	163.5 sec	0.0 kg	1.2186
50 % decrease from nominal	153 sec	0.0 kg	1.3025
zero disturbances	145.8 sec	0.0 kg	1.3647

Table 2.10: Spacecraft attitude control problem: robustness analysis of the solution with respect to uncertainties in the disturbances for an initial condition of $\omega_{0,1} = \omega_{0,2} = \theta_{0,1} = \theta_{0,2} = 0$, $m_0 = 15.27$ kg, where the controller is based on the nominal approximation of V (i.e., assuming nominal disturbances).

2.7 Summary

The focus of this chapter was on deterministic DCOC problems that were analyzed and solved using DP techniques. An optimal control policy was characterized by the value function and several new methods, including proportional feedback VI (as well as its extension to adaptive gains), an ADP approach, and base-trajectory VI, were developed to obtain the value function. In addition, properties of the value function and conditions for the existence of a solution were derived.

Numerical case studies of different DCOC problems (VDP oscillator, LEO and GEO satellite station keeping, and spacecraft attitude control) were treated. The numerical implementations of the developed methods were shown to efficiently generate approximations of optimal control policies that successfully counteract drift in order to delay constraint violations. In particular, proportional feedback VI was able to obtain a solution up to 4.2 times faster than conventional VI. Its extension, adaptive proportional feedback VI, converged to a solution more than 7 times as fast as conventional VI. Furthermore, it was demonstrated that base-trajectory VI is more accurate than (i.e., outperforms) conventional VI in a numerical setting due to smaller interpolation errors. The developed ADP approach was able to efficiently obtain good-quality suboptimal solutions. Compared to conventional DP techniques, the ADP approach mitigates the curse of dimensionality and appears to be more suitable for higher-dimensional DCOC problems.

CHAPTER 3

Deterministic DCOC – Open-Loop Solutions and MPC

3.1 Problem Formulation

The deterministic DCOC problem considered in this chapter is formulated in analogy to the previous chapter (Section 2.1). However, the sets describing the state and control constraints are assumed to be polyhedral and time-dependent here, i.e.,

$$U_t = \{u \in \mathbb{R}^p : C_{\mathrm{c},t} u \leq b_{\mathrm{c},t}\}, \tag{3.1}$$

$$G_t = \{x \in \mathbb{R}^n : C_{\mathrm{s},t} x \leq b_{\mathrm{s},t}\}, \tag{3.2}$$

where $t \in \mathbb{Z}_{\geq 0}$ denotes the time instant and $C_{\mathrm{c},t}$, $C_{\mathrm{s},t}$, $b_{\mathrm{c},t}$, and $b_{\mathrm{s},t}$ are matrices and vectors, respectively, of proper size.

Moreover, the focus is on a special class of problems with the objective of maximizing the first exit-time (time maximization problems). In addition to considering closed-loop control policies (similar to Section 2.1), the open-loop formulation of the problem is considered as well. In this regard, an open-loop control sequence is denoted by $\{u_t\} = \{u_0, u_1, u_2, ...\}$ and the set of admissible control sequences is given by

$$U_{\mathrm{seq}} = \{\{u_t\} : u_t \in U_t \text{ for all } t \in \mathbb{Z}_{\geq 0}\}. \tag{3.3}$$

Given an initial $x_0 \in G_0$ and the control sequence $\{u_t\} \in U_{\mathrm{seq}}$, the corresponding first exit-time for the open-loop case is defined as

$$\tau(x_0, \{u_t\}) = \inf\{t \in \mathbb{Z}_+ : x_t \notin G_t\}, \tag{3.4}$$

where x_t is the response of (2.1) to the initial condition x_0 and control sequence $\{u_t\}$. The

class of open-loop DCOC problems considered in this chapter is given by

$$\begin{aligned} &\max_{\{u_t\}\in U_{\text{seq}}} \tau(x_0, \{u_t\}) \\ &\text{subject to } x_{t+1} = f_t(x_t, u_t). \end{aligned} \tag{3.5}$$

Likewise, when closed-loop control policies are considered, the problem reads

$$\begin{aligned} &\max_{\pi\in\Pi} \tau(x_0, \pi) \\ &\text{subject to } x_{t+1} = f_t(x_t, \pi(x_t, t)), \end{aligned} \tag{3.6}$$

where $\pi : \mathbb{R}^n \times \mathbb{Z}_{\geq 0} \to U_t \subset \mathbb{R}^p$, $t \in \mathbb{Z}_{\geq 0}$, is a control policy, the set of admissible control policies is Π, and $\tau(x_0, \pi)$ is as in (2.2). Note that the problem formulations in (3.5) and (3.6) explicitly consider time-varying systems (as f_t is time-dependent). Equivalently, a time-varying system may be modeled by including the time instant t in the state vector x as done in the previous chapter (Sections 2.6.2.1, 2.6.3.1, and 2.6.3.2).

Throughout this chapter, it is assumed that a solution to the DCOC problem [either (3.5) or (3.6)] exists. Conditions for the existence of a solution can be found in Section 2.2. In this case, since the objective function in (3.5) or (3.6), respectively, is integer-valued, a solution to the respective DCOC problem exists under Assumption 2.2 since the maximum value of a bounded integer-valued function exists.

Assumption 3.1. There exists a solution to both problems (3.5) and (3.6) for all $x_0 \in G_0$.

Remark 3.1. *The optimal first exit-time of problems* (3.5) *and* (3.6) *is the same, i.e.,* $\tau(x_0, \{u_t^*\}) = \tau(x_0, \pi^*)$*, where* $\{u_t^*\}$ *is a solution to* (3.5) *and* π^* *is a solution to* (3.6)*, since both problems are subject to the same nonlinear system model and* $\{u_t^*\}$ *can be constructed as* $\{u_t^*\} = \{\pi^*(x_0, 0), \pi^*(f_0(x_0, \pi^*(x_0, 0)), 1), \pi^*(f_1(...), 2), ...\}$.

3.2 Open-Loop Solutions

3.2.1 Linear Systems

In this section, the open-loop linear DCOC problem is considered, which is used in Section 3.3 to formulate an MPC strategy that approximates the solution to the closed-loop nonlinear problem (3.6). The linear problem may be obtained from the nonlinear problem by linearizing the nonlinear model about a certain reference trajectory and adding a time-varying disturbance term d_t. Then, in analogy to (3.5), the open-loop linear DCOC

problem is as follows

$$\begin{aligned} &\max_{\{u_t\}\in U_{\text{seq}}} \tau(x_0, \{u_t\}) \\ &\text{subject to } x_{t+1} = A_t x_t + B_t u_t + d_t, \end{aligned} \tag{3.7}$$

where $A_t \in \mathbb{R}^{n\times n}$, $B_t \in \mathbb{R}^{n\times p}$, and $d_t \in \mathbb{R}^n$ are time-dependent matrices and vectors, respectively.

3.2.1.1 MILP Formulation

Consider the following MILP,

$$\begin{aligned} &\min_{\{u_t\},\{\delta_t\}} \sum_{t=1}^{N} \delta_t \quad \text{s.t.} \\ &x_{t+1} = A_t x_t + B_t u_t + d_t \\ &\delta_{t-1} \leq \delta_t \\ &\delta_t \in \{0,1\} \subset \mathbb{Z} \\ &C_{\text{s},t} x_t \leq b_{\text{s},t} + \mathbf{1} M \delta_t \\ &u_t \in U_t, \end{aligned} \tag{3.8}$$

where $x_0 \in G_0$, U_t is defined in (3.1), $N \in \mathbb{Z}_+$ is the time horizon, $M \in \mathbb{R}_+$, and $\mathbf{1}$ denotes the n-dimensional vector of ones. The binary variable δ_t is an indicator variable for the condition $x_t \notin G_t$, where G_t is a polyhedral set according to (3.2). Thus, in case $x_t \notin G_t$, $\delta_t = 1$ and a solution to the MILP exists if M is sufficiently large as stated in Lemma 3.1. In the following, $\{x_t\}$ denotes a state trajectory corresponding to a control sequence $\{u_t\}$ and the dynamics in (3.7).

Lemma 3.1. *Assume $M \in \mathbb{R}_+$ is sufficiently large such that $C_{\text{s},t} x_t \leq b_{\text{s},t} + \mathbf{1}M$ for all $t \in \{0, 1, ..., N\}$ and all $\{x_t\}$ satisfying* (3.8) *for any control sequence $\{u_t\} \in U_{\text{seq}}$ and $x_0 \in G_0$. Then a solution to* (3.8) *exists.*

Proof. Let $x_0 \in G_0$ be a given initial condition. Since M is sufficiently large by assumption, $\delta_t \equiv 1$ is feasible for all $\{u_t\} \in U_{\text{seq}}$ and $\{x_t\}$ satisfying $x_{t+1} = A_t x_t + B_t u_t + d_t$. Since the number of possible δ_t sequences is finite and a feasible solution exists for at least one of them, the solution existence to (3.8) follows. □

In the following theorem, conditions are stated under which solutions to MILP (3.8) are equivalent to solutions of the open-loop linear DCOC problem (3.7).

Theorem 3.1. *Suppose Assumption 3.1 holds, $N \geq \tau(x_0, \{u_t\})$ for all $\{u_t\} \in U_{\text{seq}}$, $x_0 \in G_0$, and M is sufficiently large as in Lemma 3.1. Then solutions to MILP* (3.8) *and the open-loop linear DCOC problem* (3.7) *are equivalent, i.e., a solution to the MILP is also a solution to the open-loop linear DCOC problem and vice versa.*

Proof. Let $x_0 \in G_0$ be a given initial condition. A solution to the open-loop linear DCOC problem (3.7) exists by assumption. Suppose $\{u_t^*\}$ is a solution to (3.7) with corresponding state trajectory $\{x_t^*\}$. Then,

$$\tau(x_0, \{u_t^*\}) \geq \tau(x_0, \{u_t'\}), \tag{3.9}$$

for all $\{u_t'\} \in U_{\text{seq}}$ with corresponding state trajectory $\{x_t'\}$, where $x_0 = x_0^* = x_0'$. Now (3.2), (3.4), the constraints in MILP (3.8), and $N \geq \tau(x_0, \{u_t\})$ for all $\{u_t\} \in U_{\text{seq}}$ imply that $\delta_t^* = 1$ for $t \in \{\tau(x_0, \{u_t^*\}), ..., N\}$ and $\delta_t' = 1$ for $t \in \{\tau(x_0, \{u_t'\}), ..., N\}$, where $\{\delta_t^*\}$ and $\{\delta_t'\}$ are solutions to MILP (3.8) for $\{u_t\} = \{u_t^*\}$ and $\{u_t\} = \{u_t'\}$, respectively, fixed. Consequently, $\delta_t^* = 0$ for $t < \tau(x_0, \{u_t^*\})$ and $\delta_t' = 0$ for $t < \tau(x_0, \{u_t'\})$. This and (3.9) imply that

$$\sum_{t=1}^{N} \delta_t^* = N - \tau(x_0, \{u_t^*\}) + 1 \leq N - \tau(x_0, \{u_t'\}) + 1 = \sum_{t=1}^{N} \delta_t', \tag{3.10}$$

for all $(\{u_t'\}, \{\delta_t'\})$ that satisfy the constraints of the MILP. Therefore, together with $\{\delta_t^*\}$, $\{u_t^*\}$ is a solution to the MILP.

For the second part of the proof, it needs to be shown that a solution to the MILP is also a solution of (3.7), where a solution to the MILP exists by Lemma 3.1. Suppose that $(\{u_t^{\text{MILP}}\}, \{\delta_t^{\text{MILP}}\})$ solves the MILP, i.e.,

$$\sum_{t=1}^{N} \delta_t^{\text{MILP}} \leq \sum_{t=1}^{N} \delta_t', \tag{3.11}$$

for all $(\{u_t'\}, \{\delta_t'\})$ that satisfy the constraints in (3.8). For a given admissible $\{u_t'\}$, let $\{\bar{\delta}_t'\}$ be such that $\bar{\delta}_t' = 0$ iff $t < \tau(x_0, \{u_t'\})$, which is always feasible with respect to (3.8) due to M being sufficiently large by assumption. Hence,

$$\tau(x_0, \{u_t'\}) = 1 + \sum_{t=1}^{N} (1 - \bar{\delta}_t') = N + 1 - \sum_{t=1}^{N} \bar{\delta}_t'. \tag{3.12}$$

Then, by (3.11),

$$\begin{aligned}\tau(x_0, \{u_t^{\text{MILP}}\}) &= \min\{t : \delta_t^{\text{MILP}} = 1\} \\ &= N + 1 - \sum_{t=1}^{N} \delta_t^{\text{MILP}} \geq N + 1 - \sum_{t=1}^{N} \bar{\delta}_t' = \tau(x_0, \{u_t'\}),\end{aligned} \tag{3.13}$$

for all $\{u_t'\} \in U_{\text{seq}}$. Consequently, $\{u_t^{\text{MILP}}\}$ is a solution to the DCOC problem (3.7). □

In practice, the complexity of MILP (3.8) can be reduced if a lower bound $\tau_{\text{lb}} \in \mathbb{Z}_+$ for the optimal first exit-time of the open-loop linear DCOC problem (3.7) is known, i.e., $\tau(x_0, \{u_t^*\}) \geq \tau_{\text{lb}}$, where $\{u_t^*\}$ is a solution to (3.7). In this case, $\delta_t = 0$ may be set for $t = 1, ..., \tau_{\text{lb}} - 1$, yielding MILP (3.14). Compared to MILP (3.8), MILP (3.14) reduces the number of binary variables to optimize from N to $N - \tau_{\text{lb}} + 1$. Note that τ_{lb} can be chosen as corresponding to exit-time under any given admissible control law.

$$\begin{aligned}&\min_{\{u_t\}, \{\delta_{\tau_{\text{lb}}}, ..., \delta_N\}} \sum_{t=\tau_{\text{lb}}}^{N} \delta_t \quad \text{s.t.} \\ &x_{t+1} = A_t x_t + B_t u_t + d_t \\ &\delta_{t-1} \leq \delta_t \\ &\delta_t \in \{0, 1\} \subset \mathbb{Z},\ t = \tau_{\text{lb}}, ..., N \\ &C_{\text{s},t} x_t \leq b_{\text{s},t},\ t = 1, ..., \tau_{\text{lb}} - 1 \\ &C_{\text{s},t} x_t \leq b_{\text{s},t} + \mathbf{1} M \delta_t,\ t = \tau_{\text{lb}}, ..., N \\ &u_t \in U_t.\end{aligned} \tag{3.14}$$

3.2.1.2 LP Formulation

MILP is in the class of NP-complete problems and the worst-case computation time grows exponentially with the number of integer variables $\{\delta_{\tau_{\text{lb}}}, ..., \delta_N\}$ [75–77]. Consequently, efficient and robust computation of a solution to (3.7) cannot be guaranteed with MILP. Thus, MILP (3.14) is relaxed by replacing the binary variables δ_t with non-negative real variables ε_t, which leads to a standard LP for which efficient and robust solvers exist. The LP is stated in (3.15) in analogy to MILP (3.14), where $\varepsilon_t \in \mathbb{R}_{\geq 0}$ and $q_t \in \mathbb{R}_+$ are weights.

As in MILP (3.14), $\tau_{\text{lb}} \in \mathbb{Z}_+$ is a lower bound on the optimal first exit-time of the open-loop linear DCOC problem (3.7) and $\varepsilon_t = 0$ for $t = 1, ..., \tau_{\text{lb}} - 1$. The solution to LP (3.15) is only guaranteed to be optimal with respect to (3.7) when the time horizon is $N = \tau(x_0, \{u_t^*\}) - 1$, where $\{u_t^*\}$ is a solution to (3.7). In contrast to the MILP formulation,

$N \geq \tau(x_0, \{u_t^*\})$ does not guarantee an optimal solution with respect to (3.7). Furthermore, note that (3.15) does not require the upper bound M used in (3.14). On the other hand, if such an M is known, under the additional constraint $\varepsilon_t \leq M$ and for $q_t \equiv 1/M$, (3.15) corresponds to the LP relaxation of (3.14) by setting $\delta_t = \varepsilon_t / M$, $0 \leq \delta_t \leq 1$.

$$
\begin{aligned}
&\min_{\{u_t\},\{\varepsilon_{\tau_{\mathrm{lb}}},\ldots,\varepsilon_N\}} \sum_{t=\tau_{\mathrm{lb}}}^{N} q_t \varepsilon_t \quad \text{s.t.} \\
&x_{t+1} = A_t x_t + B_t u_t + d_t \\
&0 \leq \varepsilon_{t-1} \leq \varepsilon_t \\
&C_{\mathrm{s},t} x_t \leq b_{\mathrm{s},t}, \quad t = 1, \ldots, \tau_{\mathrm{lb}} - 1 \\
&C_{\mathrm{s},t} x_t \leq b_{\mathrm{s},t} + \mathbf{1}\varepsilon_t, \quad t = \tau_{\mathrm{lb}}, \ldots, N \\
&u_t \in U_t.
\end{aligned}
\tag{3.15}
$$

3.2.1.3 Iterative Procedure

The time horizon in Theorem 3.1 is assumed to satisfy $N \geq \tau(x_0, \{u_t\})$ for all admissible control sequences $\{u_t\}$. This condition can be restated as $N \geq \tau(x_0, \{u_t^*\}) - 1$, where $\{u_t^*\}$ is a solution to the open-loop linear DCOC problem (3.7). It is straightforward to show that solutions to the MILP can only be optimal with respect to (3.7) if N satisfies this condition. However, the optimal first exit-time is a priori unknown and, consequently, it is not possible to choose N such that $N \geq \tau(x_0, \{u_t^*\}) - 1$ is guaranteed to hold. Moreover, choosing N very large, i.e., $N \gg \tau(x_0, \{u_t^*\})$, is prohibitive because it increases the number of integer variables, which in turn increases (possibly exponentially) the computation time.

A similar problem arises when solving LP (3.15). While the solution to the LP is not guaranteed to be optimal with respect to (3.7) for $N \neq \tau(x_0, \{u_t^*\}) - 1$, the best solutions appear to be obtained when $N = \tau(x_0, \{u_t^*\}) - 1 + \gamma$ for some small $\gamma \in \mathbb{Z}_{\geq 0}$ or when τ_{lb} is close to $\tau(x_0, \{u_t^*\})$. Therefore, an algorithm is proposed that iteratively updates N while reducing the number of decision variables δ_t or ε_t, respectively, until a proper N is found. The algorithm for the LP (Algorithm 3.1) is stated first because its solution may be used to initialize the algorithm for the MILP (Algorithm 3.2).

The LP-based Algorithm 3.1 is as follows. In Step 1, the lower bound τ_{lb} is initialized based on the zero-control solution (assuming $0 \in U_t$ for all $t \in \mathbb{Z}_{\geq 0}$; otherwise any other admissible control sequence can be used to calculate a lower bound τ_{lb}). The time horizon N is initialized by adding a constant $\alpha^{\mathrm{LP}} \in \mathbb{Z}_+$ to τ_{lb} at Step 2. Then LP (3.15) is solved. If the solution does not exit G_t for the current time horizon, the solution is used as a new

lower bound (Step 6) and the time horizon N is increased by α^{LP} (Step 2). The procedure is repeated until the solution exits G_t. The number of variables for each LP in Algorithm 3.1 is $N(n+p)+\alpha^{\mathrm{LP}}+1$, where n and p are the dimensions of the state and control input vectors, respectively.

Algorithm 3.1 Obtain suboptimal solution to (3.7) based on LP

1: $\tau_{\mathrm{lb}} \leftarrow \tau(x_0, \{0\})$
2: $N \leftarrow \tau_{\mathrm{lb}} + \alpha^{\mathrm{LP}},\ \alpha^{\mathrm{LP}} \in \mathbb{Z}_+$
3: $\{u_t^{\mathrm{LP}}\}, \{\varepsilon_{\tau_{\mathrm{lb}}}^{\mathrm{LP}}, ..., \varepsilon_N^{\mathrm{LP}}\} \leftarrow$ solution of LP (3.15)
4: $\tau \leftarrow \max\{t \leq N : \varepsilon_t^{\mathrm{LP}} = 0\} + 1$
5: **if** $\varepsilon_N^{\mathrm{LP}} = 0$ **then**
6: $\quad \tau_{\mathrm{lb}} \leftarrow \tau$
7: $\quad$ go to Step 2
8: **end if**

There is a tradeoff between computation time and quality of the solution when choosing the parameter α^{LP}. In order to guarantee good-quality solutions, α^{LP} needs to be small because small updates on N ensure that the eventual N and τ_{lb} are close to the optimal first exit-time [which yields LP solutions close to a solution of problem (3.7)]. On the other hand, α^{LP} being small may require several iterations (i.e., solving LP (3.15) several times) in Algorithm 3.1 until the solution violates the prescribed constraints and Algorithm 3.1 terminates. Hence, a balance between the two extremes (α^{LP} being too small or too large) is desirable in order to efficiently obtain good-quality solutions with Algorithm 3.1.

Algorithm 3.2 Obtain solution to (3.7) based on MILP

1: $\tau_{\mathrm{lb}} \leftarrow$ output of Algorithm 3.1
2: $N \leftarrow \tau_{\mathrm{lb}} + \alpha^{\mathrm{MILP}}$
3: $\{u_t^{\mathrm{MILP}}\}, \{\delta_{\tau_{\mathrm{lb}}}^{\mathrm{MILP}}, ..., \delta_N^{\mathrm{MILP}}\} \leftarrow$ solution of MILP (3.14)
4: $\tau \leftarrow \max\{t \leq N : \delta_t^{\mathrm{MILP}} = 0\} + 1$
5: **if** $\delta_N^{\mathrm{MILP}} = 0$ **then**
6: $\quad \tau_{\mathrm{lb}} \leftarrow \tau$
7: $\quad$ go to Step 2
8: **end if**

Algorithm 3.2 outlines the iterative procedure based on MILP, which, according to Theorem 3.1, obtains an optimal solution with respect to the open-loop linear DCOC problem (3.7). The LP-based Algorithm 3.1 is used to initialize the lower bound τ_{lb} in Step 1. The

time horizon N is initialized in Step 2 by adding a constant integer α^{MILP} to τ_{lb} as in Algorithm 3.1. Then MILP (3.14) is solved and the time horizon and lower bound are updated until the solution exits the set G_t. Note that this procedure can be very effective for solving MILP because the number of binary variables at each iteration is $\alpha^{\text{MILP}} + 1$, where α^{MILP} is specified by the user.

3.2.2 Nonlinear Systems

After discussing the open-loop linear DCOC problem in the previous section (Section 3.2.1), this section focuses on the open-loop nonlinear DCOC problem (3.5). The program that solves (3.5) is similar to MILP (3.14), whereas the linear equality constraints in (3.14) are replaced by $x_{t+1} = f_t(x_t, u_t)$ to account for the nonlinear dynamics. This yields the following MINLP,

$$
\begin{aligned}
&\min_{\{u_t\},\{\delta_{\tau_{\text{lb}}},\ldots,\delta_N\}} \sum_{t=\tau_{\text{lb}}}^{N} \delta_t \quad \text{s.t.} \\
&x_{t+1} = f_t(x_t, u_t) \\
&\delta_{t-1} \leq \delta_t \\
&C_{\text{s},t} x_t \leq b_{\text{s},t}, \quad t = 1, \ldots, \tau_{\text{lb}} - 1 \\
&C_{\text{s},t} x_t \leq b_{\text{s},t} + \mathbf{1} M \delta_t, \quad t = \tau_{\text{lb}}, \ldots, N \\
&u_t \in U_t \\
&\delta_t \in \{0, 1\} \subset \mathbb{Z}, \quad t = \tau_{\text{lb}}, \ldots, N.
\end{aligned}
\tag{3.16}
$$

Theorem 3.2 provides conditions under which the solutions of the open-loop nonlinear DCOC problem (3.5) and MINLP (3.16) are equivalent. The proof of Theorem 3.2 is similar to the proof of Theorem 3.1 (linear case). For the sake of completeness, the proof can be found in Appendix B, where the lower bound, τ_{lb}, on the optimal first exit-time is explicitly considered. Similar to the assumptions in Lemma 3.1, the following assumptions about M and the time horizon N are made.

Assumption 3.2. The time horizon of MINLP (3.16) is sufficiently large such that $N \geq \tau(x_0, \{u_t\})$ for all $\{u_t\} \in U_{\text{seq}}$ and $x_0 \in G_0$. Moreover, M in (3.16) is sufficiently large such that, for any $\{u_t\} \in U_{\text{seq}}$ with corresponding state trajectory $\{x_t\}$ according to the dynamics in (3.5), $C_{\text{s},t} x_t \leq b_{\text{s},t} + \mathbf{1} M$ for all $t = \tau_{\text{lb}}, \ldots, N$.

Theorem 3.2. *Suppose Assumptions 3.1 and 3.2 hold. Then the solutions to the open-loop nonlinear DCOC problem* (3.5) *and MINLP* (3.16) *are equivalent.*

Proof. See Appendix B. □

As MILP, MINLP is in the class of NP-complete problems. Moreover, there are no MINLP solvers, especially in the online setting, that can obtain good-quality solutions to (3.16) for the problems considered in this dissertation (see Section 3.4). Therefore, similar to the linear problem in Section 3.2.1, MINLP (3.16) is relaxed by replacing the binary variables with non-negative real variables, yielding the following NLP

$$\begin{aligned}
&\min_{\{u_t\},\{\varepsilon_{\tau_{\mathrm{lb}}},\ldots,\varepsilon_N\}} \sum_{t=\tau_{\mathrm{lb}}}^{N} \varepsilon_t \quad \text{s.t.} \\
&x_{t+1} = f_t(x_t, u_t) \\
&0 \leq \varepsilon_{t-1} \leq \varepsilon_t \\
&C_{\mathrm{s},t} x_t \leq b_{\mathrm{s},t}, \;\; t = 1, \ldots, \tau_{\mathrm{lb}} - 1 \\
&C_{\mathrm{s},t} x_t \leq b_{\mathrm{s},t} + \mathbf{1}\varepsilon_t, \;\; t = \tau_{\mathrm{lb}}, \ldots, N \\
&u_t \in U_t.
\end{aligned} \tag{3.17}$$

NLP (3.17) is expected to obtain good-quality suboptimal solutions to the open-loop nonlinear DCOC problem (3.5) for proper choices of the time horizon N and upper bound τ_{lb}, i.e., N and τ_{lb} being close to the optimal first-exit time of problem (3.5). Similar to Section 3.2.1.3 (Algorithms 3.1 and 3.2), iterative procedures may be defined to update the time horizon and τ_{lb} of the respective nonlinear program, (3.16) or (3.17), until a proper solution is found. This is not pursued here since (repeatedly) solving nonlinear programs may take considerably longer than solving linear programs. Instead, based on linear model approximation of the nonlinear model, the solution of either MILP (3.14) or LP (3.15) (obtained by Algorithms 3.1 or 3.2) is used to set a proper τ_{lb} and time horizon for the nonlinear program, which may be further increased if the corresponding solution does not violate constraints.

3.3 MPC Scheme

This section describes how the MILP or LP formulation (Sections 3.2.1.1 and 3.2.1.2) is used to implement feedback control in order to compensate for unmodeled effects online and obtain a good-quality suboptimal solution to the closed-loop nonlinear DCOC problem (3.6). In this regard, the nonlinear model in (3.6) is approximated by a linear model of the form in (3.7), which may be obtained by linearizing the nonlinear model about a proper reference trajectory. Due to the availability of efficient and robust LP solvers, the MPC

scheme presented here is based on LP (3.15). Feedback is provided by recomputing the LP solution over a receding time horizon based on the current state vector of the system and applying the first element of the computed control sequence to the system at each time instant.

As outlined in the top of Figure 3.1, the linear-based MPC scheme may violate constraints ($x_t \notin G_t$) prematurely when applied to the nonlinear model (especially, when x_t is close to the boundary of G_t). This is due to unmodeled effects and may be prevented by sufficiently tightening the constraints for control computation, meaning the control is computed based on tightened state constraints in order to create a margin of safety. In analogy to (3.2), the tightened state constraints are defined by

$$\tilde{G}_t = \{x \in \mathbb{R}^n : \tilde{C}_{\mathrm{s},t} x \leq \tilde{b}_{\mathrm{s},t}\} \subset G_t, \tag{3.18}$$

and, using $q_t \equiv 1$, the modified LP for the MPC implementation is given by

$$\begin{aligned}
&\min_{\{u_t\},\{\varepsilon_{\tau_{\mathrm{lb}}},\ldots,\varepsilon_N\}} \sum_{t=\tau_{\mathrm{lb}}}^{N} \varepsilon_t \quad \text{s.t.} \\
&x_{t+1} = A_t x_t + B_t u_t + d_t \\
&0 \leq \varepsilon_{t-1} \leq \varepsilon_t \\
&\tilde{C}_{\mathrm{s},t} x_t \leq \tilde{b}_{\mathrm{s},t}, \quad t = 1, \ldots, \tau_{\mathrm{lb}} - 1 \\
&\tilde{C}_{\mathrm{s},t} x_t \leq \tilde{b}_{\mathrm{s},t} + \mathbf{1}\varepsilon_t, \quad t = \tau_{\mathrm{lb}}, \ldots, N \\
&u_t \in U_t,
\end{aligned} \tag{3.19}$$

where $x_0 \in \tilde{G}_0$. In addition, the MPC strategy is augmented by a controller that tries to recover $x_t \in \tilde{G}_t$ when the tightened constraints are violated (i.e., $x_t \notin \tilde{G}_t$). This controller is referred to as the recovery controller and is described by LP (3.20), which is similar to LP (3.19),

$$\begin{aligned}
&\min_{\{u_t\},\{\varepsilon_t\}} \sum_{t=1}^{N_{\mathrm{recover}}} \varepsilon_t \quad \text{s.t.} \\
&x_{t+1} = A_t x_t + B_t u_t + d_t \\
&0 \leq \varepsilon_t \\
&\tilde{C}_{\mathrm{s},t} x_t \leq \tilde{b}_{\mathrm{s},t} + \mathbf{1}\varepsilon_t \\
&u_t \in U_t,
\end{aligned} \tag{3.20}$$

where $x_0 \notin \tilde{G}_0$. In contrast to LP (3.19), initial constraint violation is assumed and the

inequality constraints $\varepsilon_{t-1} \leq \varepsilon_t$ are removed. Thus, the control sequence obtained by LP (3.20) tries to steer the state vector back into the set $\tilde{G}_t$ over the time horizon N_{recover}. Control computation based on $\tilde{G}_t$ [LP (3.19)], together with the recovery controller [LP (3.20)], may prevent premature violation of the original constraints ($x_t \notin G_t$) as illustrated in the bottom of Figure 3.1.

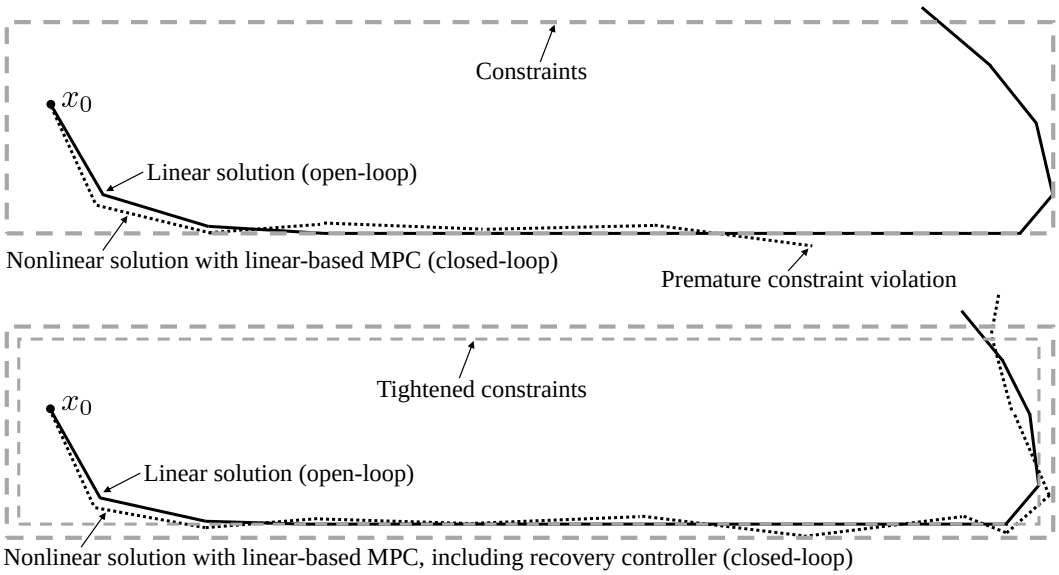

Figure 3.1: Illustration of effects of constraint tightening and recovery controller when linear-based MPC scheme is applied to nonlinear model. Top: state trajectories without constraint tightening. Bottom: state trajectories with constraint tightening and recovery controller.

The MPC strategy is outlined in Algorithm 3.3. At each time instant t_{sys}, the current state vector $x(t_{\text{sys}})$ is acquired and used as the initial x_0 for control computation (Step 4 of Algorithm 3.3). The time-dependent dynamics and constraints for the linear DCOC problem (3.7) are obtained in Step 3 based on the current time instant t_{sys} of the system for $t \in \{0, 1, ..., N_{\text{ub}}+\alpha^{\text{LP}}\}$, where $N_{\text{ub}}+\alpha^{\text{LP}}$ is the largest possible time horizon for LP (3.19) and $N_{\text{recover}} \ll N_{\text{ub}}$ is used for the time horizon of LP (3.20). The parameter N_{ub} is defined below. If the tightened state constraints are not satisfied by the current state, the recovery controller [LP (3.20)] is employed in Step 6 of Algorithm 3.3. Otherwise, in combination with a modified version of Algorithm 3.1, LP (3.19) is used for control computation. The first element of the computed control sequence is applied to the system in Step 20.

The modifications of the iterative procedure in Algorithm 3.1 for the MPC strategy include an upper bound, N_{ub}, on the time horizon N (Step 15 of Algorithm 3.3). This allows premature termination of the iteration if, for example, computation time limits need

to be satisfied. However, the optimal first exit-time for the current state vector may be greater than N_{ub}, which may lower the quality of the resulting solution.

Algorithm 3.3 LP-based MPC implementation

1: $t_{\mathrm{sys}} \leftarrow 0$
2: $\tau_{\mathrm{lb},0} \leftarrow$ set initial lower bound
3: $A_t, B_t, d_t, \tilde{G}_t, U_t \leftarrow$ obtain dynamics and constraints for all $t \in \{0, 1, ..., N_{\mathrm{ub}} + \alpha^{\mathrm{LP}}\}$
4: $x_0 \leftarrow$ current state $x(t_{\mathrm{sys}})$
5: **if** $x_0 \notin \tilde{G}_0$ **then**
6: $\quad \{u_t\} \leftarrow$ solution of LP (3.20)
7: **else**
8: $\quad \tau_{\mathrm{lb}} \leftarrow \tau_{\mathrm{lb},0}$
9: $\quad N \leftarrow \tau_{\mathrm{lb}} + \alpha^{\mathrm{LP}}$
10: $\quad$ **if** LP (3.19) is infeasible **then**
11: $\quad\quad \tau_{\mathrm{lb}} \leftarrow \min\{\tilde{\tau}(x_0, \{0\}), N_{\mathrm{ub}}\}$; go to Step 9
12: $\quad$ **end if**
13: $\quad \{u_t\}, \{\varepsilon_{\tau_{\mathrm{lb}}}, ..., \varepsilon_N\} \leftarrow$ solution of LP (3.19)
14: $\quad \tau \leftarrow \max\{t \leq N : \varepsilon_t = 0\} + 1$
15: $\quad$ **if** $\varepsilon_N = 0$ AND $N < N_{\mathrm{ub}}$ **then**
16: $\quad\quad \tau_{\mathrm{lb}} \leftarrow \tau$; go to Step 9
17: $\quad$ **end if**
18: $\quad \tau_{\mathrm{lb},0} \leftarrow \min\{\tau - 1, N_{\mathrm{ub}}\}$
19: **end if**
20: Apply u_0 as control input $u(t_{\mathrm{sys}})$ to the system
21: $t_{\mathrm{sys}} \leftarrow t_{\mathrm{sys}} + 1$; go to Step 3

In addition to N_{ub}, the variable $\tau_{\mathrm{lb},0}$ is introduced to initialize the lower bound, τ_{lb}, on the optimal first exit-time of the open-loop linear DCOC problem for LP (3.19) in Step 8, where $\tau_{\mathrm{lb},0}$ is updated over the receding time horizon in Step 18 based on the first exit-time of the previously computed solution. This significantly reduces the number of decision variables ε_t of LP (3.19) and decreases computation times. However, Steps 10 – 12 need to be added to check if LP (3.19) is feasible for the current τ_{lb} and N. This is important because, at time instant $t_{\mathrm{sys}}+1$, the predicted τ_{lb} may be greater than the actual optimal first exit-time for $x_0 \leftarrow x(t_{\mathrm{sys}} + 1)$ as a consequence of prediction errors caused by unmodeled effects. In this case, LP (3.19) becomes infeasible. Feasibility is recovered by recomputing τ_{lb} in Step 11 using the zero-control solution $\tilde{\tau}(x_0, \{0\})$ with respect to the tightened state constraints given by $\tilde{G}_t$, see (3.18), assuming $0 \in U_t$ (otherwise, any admissible control

sequence can be used).

3.4 Numerical Case Studies

3.4.1 VDP Oscillator and Spacecraft Attitude Control 1

Two numerical case studies of a VDP oscillator (Section 3.4.1.1) and of spacecraft attitude control (Section 3.4.1.2) are considered. In both case studies, the underlying model is linear, obtained through linearization and discretization of the respective nonlinear continuous-time model, and the open-loop linear DCOC problem (3.7) is investigated. In addition, the MPC scheme proposed in Section 3.3 is simulated in closed-loop with the linear model, even though the simulation model and the controller are based on the same model, i.e., there are no unmodeled effects. The performance of the MPC scheme when there are unmodeled effects is investigated in Section 3.4.2. Since there are no unmodeled effects in the two case studies, tightening the state constraints (see Figure 3.1) is not necessary and $\tilde{G}_t = G_t$ [see (3.18)] for all t.

In both problems, the objective is to maximize the time until specified constraints are violated for the first time. The simulations are stopped as soon as $x_t \notin G_t$ and the respective first exit-time is reported. Based on the developed approaches in Sections 3.2.1 and 3.3, the performance of the following controllers is analyzed:

- **MILP-direct / LP-direct (open-loop)**: direct solution of MILP (3.14) / LP (3.15) with $\tau_{\rm lb} = \tau(x_0, \{0\})$.
- **LP-iter / MILP-iter (open-loop)**: solution of LP or MILP using Algorithms 3.1 or 3.2, respectively.
- **MPC (closed-loop)**: LP-based MPC implementation (Algorithm 3.3) with $N_{\rm ub} = \infty$ (i.e., no bound on N is used) and $\tau_{\rm lb,0} = \tau(x_0, \{0\})$ in Step 2 of Algorithm 3.3.

The computation times reported in this section are for a laptop with an i5-6300 processor and 8 GB RAM running MATLAB 2015a. The Hybrid Toolbox [78] (lpsol and milpsol functions with default settings) is used for solving LPs and MILPs. For each MILP, $M = 100$ and, for LPs, $q_t \equiv 1$.

3.4.1.1 VDP Oscillator

Consider the continuous-time nonlinear model

$$\ddot{r}_{\rm VDP} = (1 - r_{\rm VDP}^2)\dot{r}_{\rm VDP} - r_{\rm VDP} + u. \tag{3.21}$$

Let $x = [r_1, r_2]^\top$ be the state vector, where $r_1 = r_{\text{VDP}}$ and $r_2 = \dot{r}_{\text{VDP}}$. Through linearization about $x_{\text{lin}} = [1.5, 2]^\top$ and using Euler's forward method with a sampling time $\Delta t = 0.015$ sec, the discrete-time linear model with added sinusoidal disturbance is obtained as follows

$$\begin{bmatrix} r_{1,t+1} \\ r_{2,t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0.015 \\ -0.105 & 0.9812 \end{bmatrix} \begin{bmatrix} r_{1,t} \\ r_{2,t} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.015 \end{bmatrix} u_t + \begin{bmatrix} 0 \\ 0.05 \sin(2\pi t \Delta t) \end{bmatrix}, \tag{3.22}$$

where the control input is $u_t \in [-12, 12]$. The state constraints for the DCOC problem are given by

$$G_t \equiv \{x : r_1 \in [1, 2], r_2 \in [1, 3]\}.$$

Controller	Parameter	τ	Computation time (msec)
MILP-direct	$N = 45$	44	7
	$N = 55$	44	10
	$N = 75$	44	14
MILP-iter $\alpha^{\text{LP}} = 25$	$\alpha^{\text{MILP}} = 5$	44	10
	$\alpha^{\text{MILP}} = 10$	44	11
	$\alpha^{\text{MILP}} = 15$	44	11
LP-direct	$N = 45$	44	3
	$N = 55$	44	4
	$N = 75$	41	7
LP-iter	$\alpha^{\text{LP}} = 5$	44	7
	$\alpha^{\text{LP}} = 15$	44	4
	$\alpha^{\text{LP}} = 25$	43	7

Table 3.1: VDP oscillator case study, open-loop control sequences with different parameters: first exit-time τ and computation time (worst-case over 100 simulation runs).

An initial $x_0 = [1, 3]^\top$ is assumed, for which the zero-control first exit-time $\tau(x_0, \{0\})$ is 15. Table 3.1 shows the first exit-time τ and the required computation time for the open-loop controllers. The solution of the MILP-based iterative procedure (Algorithm 3.2) is always optimal and the optimal first exit-time for this problem is 44. The LP-based open-loop control sequences obtain a solution faster than with MILP. However, it is not guaranteed that the optimal exit-time is achieved. The direct solution of LP (3.15) yields an optimal solution when N is close to the optimal exit-time. Both LP-direct and MILP-direct cannot find an optimal solution if $N < 43$. On the other hand, the iterative procedures

(MILP-iter and LP-iter) do not rely on guessing N sufficiently large and are therefore more robust in computing a solution to the DCOC problem.

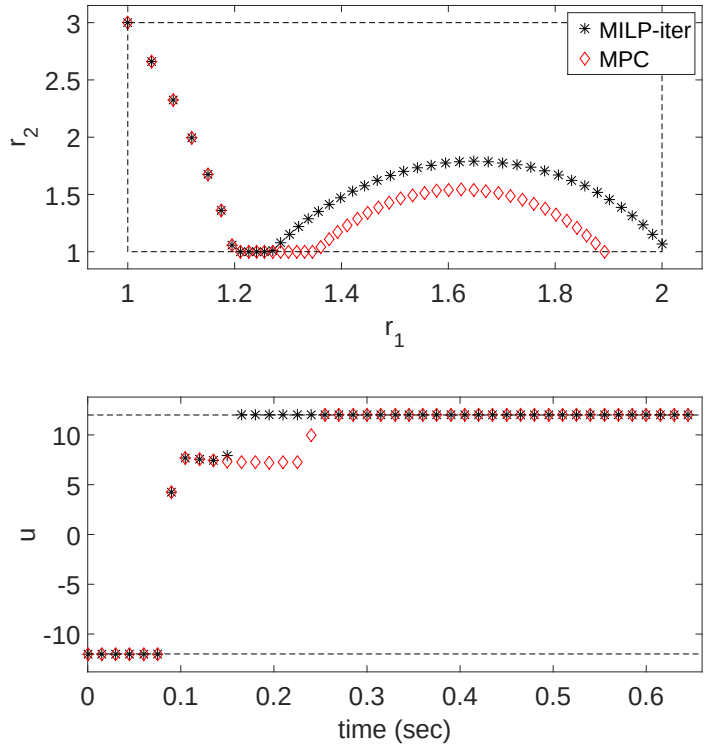

Figure 3.2: VDP oscillator case study. Top: state trajectories in r_1-r_2 plane. Bottom: control input u vs. time.

The LP-based MPC implementation (Algorithm 3.3 with $\alpha^{\mathrm{LP}} = 25$) obtains the optimal first exit-time of 44. The average and worst-case computation times over all time instants (over 100 simulation runs) are 2 msec and 7 msec, respectively, which suggests that it is feasible to recompute the control input at each time instant in real-time. Figure 3.2 shows the state and control trajectories for the open-loop controller MILP-iter and the MPC implementation. The state and control constraints are indicated by black dashed lines. The trajectories are different but each exits G_t after 44 steps. Thus, the optimal solution is not unique in this problem.

3.4.1.2 Spacecraft Attitude Control 1

The second problem is the attitude control of an underactuated spacecraft with the body-fixed frame being a principal frame and principal axes denoted by 1, 2, and 3. The spacecraft is equipped with two RWs aligned with the 1- and 3-axis, respectively, where

the moment of inertia of each wheel is $J_{\mathrm{w}} = 0.01\,\mathrm{kgm}^2$. The spacecraft principal moments of inertia are given by $J_1 = J_2 = 800\,\mathrm{kgm}^2$ and $J_3 = 300\,\mathrm{kgm}^2$. The spacecraft orientation is subject to drift caused by a constant external torque (e.g., from solar radiation pressure, where the orientation does not significantly change) with $M_1 = -1.2 \times 10^{-5}\,\mathrm{Nm}$, $M_2 = -10^{-5}\,\mathrm{Nm}$, and $M_3 = 0.9 \times 10^{-5}\,\mathrm{Nm}$. The state vector is $x = [\phi, \theta, \psi, \omega_1, \omega_2, \omega_3, \omega_{\mathrm{w1}}, \omega_{\mathrm{w3}}]^\top$, where ϕ, θ, and ψ are the 3-2-1 Euler angles describing the spacecraft orientation, ω_1, ω_2, and ω_3 are the spacecraft angular velocity vector projections onto the principal axes, and ω_{w1} and ω_{w3} are the respective RW spin rates. The control input vector is $u = [\alpha_{\mathrm{w1}}, \alpha_{\mathrm{w3}}]^\top$ comprising the angular accelerations of the two RWs, which are constrained by $\alpha_{\mathrm{w1}}, \alpha_{\mathrm{w3}} \in [-1, 1]\,\mathrm{rad/sec}^2$. Note that since the spacecraft is acted on by an external torque, its angular momentum is not conserved and the reduced order equations, obtained by eliminating the angular velocities, cannot be used.

Controller	Parameter	τ	Computation time (sec)
MILP-direct	$N = 175$	172	0.83
	$N = 200$	172	2.07
	$N = 225$	172	6.89
MILP-iter $\alpha^{\mathrm{LP}} = 50$	$\alpha^{\mathrm{MILP}} = 5$	172	1.35
	$\alpha^{\mathrm{MILP}} = 10$	172	1.39
	$\alpha^{\mathrm{MILP}} = 15$	172	1.44
LP-direct	$N = 175$	172	0.45
	$N = 200$	172	0.65
	$N = 225$	77	0.86
LP-iter	$\alpha^{\mathrm{LP}} = 25$	172	1.2
	$\alpha^{\mathrm{LP}} = 50$	172	1.02
	$\alpha^{\mathrm{LP}} = 75$	172	0.83

Table 3.2: Spacecraft attitude control case study, open-loop control sequences with different parameters: first exit-time τ and worst-case computation time over 100 simulation runs.

The objective for this problem is to maintain x within the set

$$G_t \equiv \{x : \phi, \theta \in [44.995, 45.005]\,\mathrm{deg},\ \psi \in [44.95, 45.05]\,\mathrm{deg},$$
$$\omega_{\mathrm{w1}} \in [10, 200]\,\mathrm{rad/sec},\ \omega_{\mathrm{w3}} \in [-200, -10]\,\mathrm{rad/sec}\},$$

for as long as possible. This set is defined by bounds on spacecraft attitude and RW spin

rates (RWs must operate below maximum speeds and avoid zero crossing). The constraints on the orientation are relatively tight, and correspond to precise pointing requirements required for some missions such as Kepler [79].

The discrete-time linear model is derived from the nonlinear continuous-time model [65] by linearizing about $x_{\text{lin}} = [0, 0, 0, \pi/4, \pi/4, \pi/4, 190, -100]^\top$ and using Euler's forward method with a sampling time $\Delta t = 2$ sec, yielding (for x and u in SI units) the following discrete-time equations

$$x_{t+1} = \begin{bmatrix} 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 1 & .003 & 0 & 0 & 0 \\ 0 & 0 & 0 & -.003 & 1 & -.005 & 0 & 0 \\ 0 & 0 & 0 & 0 & .013 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_t + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -.5 \times 10^{-5} & 0 \\ 0 & 0 \\ 0 & -.7 \times 10^{-5} \\ 2 & 0 \\ 0 & 2 \end{bmatrix} u_t + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -3 \times 10^{-8} \\ -.5 \times 10^{-8} \\ 6 \times 10^{-8} \\ 0 \\ 0 \end{bmatrix}.$$

The numerical conditioning of each LP and MILP is improved by normalizing the state vector according to $\hat{x} = O_{\text{tf}} x + o_{\text{tf}}$, where $O_{\text{tf}} \in \mathbb{R}^{8\times 8}$ and $o_{\text{tf}} \in \mathbb{R}^8$ are such that G_t is transformed into $\hat{G}_t \equiv \{\hat{x} : \hat{\phi}, \hat{\theta}, \hat{\psi}, \hat{\omega}_{\text{w1}}, \hat{\omega}_{\text{w3}} \in [0, 1]\}$, $\omega_1, \omega_2 = -10^{-4}\,\text{rad/sec}$, $\omega_3 = -10^{-2}\,\text{rad/sec}$ correspond to $\hat{\omega}_1, \hat{\omega}_2, \hat{\omega}_3 = 0$, and $\omega_1, \omega_2 = 10^{-4}\,\text{rad/sec}$, $\omega_3 = 10^{-2}\,\text{rad/sec}$ correspond to $\hat{\omega}_1, \hat{\omega}_2, \hat{\omega}_3 = 1$.

The following results are for an initial $x_0 = x_{\text{lin}}$ for which the zero-control exit-time is 54. Table 3.2 shows the first exit-time and computation time for different open-loop controllers. The results are similar to the VDP oscillator case study in Section 3.4.1.1. The optimal first exit-time is 172, see MILP-based open-loop controllers in Table 3.2. The LP-based open-loop controllers obtain an optimal solution if the time horizon N is close to the optimal first-exit time. In contrast to LP-direct, the worst-case computation time of MILP-direct increases exponentially with N, which, on the other hand, is prevented with the iterative procedure (MILP-iter). The MPC strategy (Algorithm 3.3 with $\alpha^{\text{LP}} = 50$) achieves the optimal first exit-time of 172. The worst-case computation time over 100 simulation runs is 1.14 sec. This is smaller than the sampling time ($\Delta t = 2$ sec), which suggest that real-time computation is possible. On average (over 100 simulation runs), 0.33 sec are required to compute the control input at each time instant.

Constraint violation occurs due to the uncontrollable Euler angle θ reaching its prescribed limit. Figure 3.3 (top) shows θ over time for the open-loop controller MILP-iter and the MPC implementation, where the constraints are indicated by black dashed lines.

As for the VDP oscillator problem (Figure 3.2), the optimal solution is not unique here. The control inputs are shown in Figure 3.3 (middle and bottom plot). It can be seen that both controllers are aggressively accelerating/decelerating the RWs, which may increase RW wear and the risk of RW failure. Hence, a term may be added to the objective function of the respective mathematical program to penalize excessive control inputs. This approach is pursued in Section 3.4.2.2 for a similar DCOC problem.

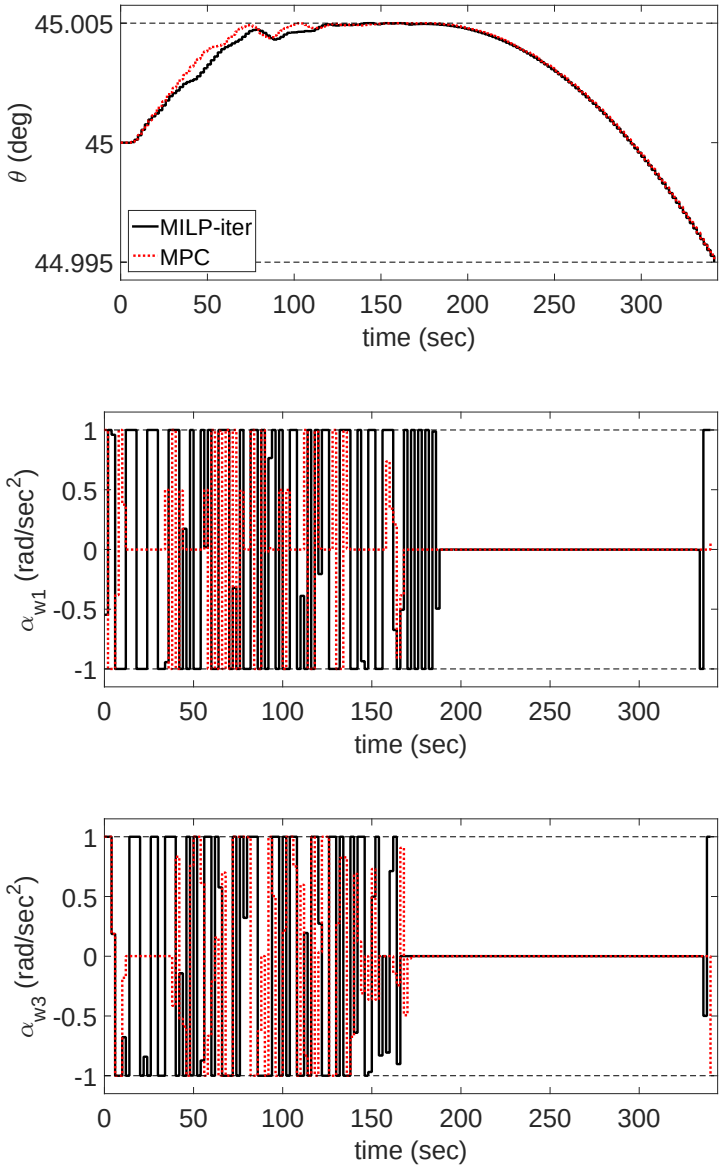

Figure 3.3: Spacecraft attitude control case study. Uncontrollable Euler angle θ vs. time (top) as well as control inputs α_{w1} (middle) and α_{w3} (bottom) vs. time.

3.4.2 GEO Satellite Station Keeping & Spacecraft Attitude Control 2

The LP-based MPC strategy from Section 3.3 is used to approximate a solution to the closed-loop nonlinear DCOC problem (3.6) in this section. Numerical results for two problems of GEO satellite station keeping (Section 3.4.2.1) and spacecraft attitude control (Section 3.4.2.2) are treated. In both problems, the open-loop solution of NLP (3.17) is compared against the LP-based MPC strategy simulated in closed-loop with the discrete-time nonlinear model. In addition, the MPC strategy is simulated in closed-loop with the corresponding continuous-time nonlinear model in order to provide more realistic results, where the control is held at a constant value during each sampling period (i.e., using a zero-order hold). Hence, the following three simulation scenarios are analyzed:

- **NLP-Discrete-Time**: open-loop solution of NLP (3.17) simulated on the discrete-time nonlinear model.
- **MPC-Discrete-Time**: MPC strategy (Algorithm 3.3) simulated in closed-loop with the discrete-time nonlinear model.
- **MPC-Continuous-Time**: MPC strategy (Algorithm 3.3) simulated in closed-loop with the continuous-time nonlinear model with a zero-order hold applied to the control input during each sampling period.

The GEO station keeping problem in Section 3.4.2.1 assumes a satellite subject to perturbations due to luni-solar gravity, SRP, and J2. Note that additional perturbations can readily be included, which is not done here since the results are compared to [80] where a satellite model with the aforementioned perturbations was considered. The satellite is equipped with either continuous or on/off thrusters, which consume a certain amount of fuel. Given an initial amount of fuel, the objective is to satisfy prescribed constraints on the satellite position and remaining fuel for as long as possible. Compared to other station keeping approaches [80–84], the DCOC approach yields the longest operation times for GEO satellites by directly maximizing the time until constraint violation.

For the spacecraft attitude control problem in Section 3.4.2.2, RWs are used to counteract drift caused by SRP disturbance torques. The cases of an underactuated spacecraft (one or two operable RWs) [79, 85–87] as well as of a fully actuated spacecraft with one RW being nearly saturated are considered. In both cases the control authority is limited. Consequently, prescribed orientation constraints will eventually be violated and the objective is to delay this event. As in Section 3.4.1.2, this case study is motivated by frequent situations, such as for the Kepler spacecraft [79,88,89], where pointing must be maintained to be able to image while RWs have failed.

As in Section 3.4.1, all computations are performed in MATLAB 2015a on a laptop with an i5-6300 processor and 8 GB RAM. LPs are solved with the Hybrid Toolbox [78] (lpsol function with default settings). While the proposed framework (see Section 3.1) allows the treatment of time-dependent state and control constraints, time-invariant constraints are assumed in both problems ($G_t \equiv G$ and $U_t \equiv U$). Similarly, $A_t \equiv A$ and $B_t \equiv B$ for the respective linear DCOC problem (3.7).

3.4.2.1 GEO Satellite Station Keeping

Nonlinear Model

Let frame $\mathcal{I}$ be the ECI frame and frame $\mathcal{H}$ be Hill's frame. The 1-axis of Hill's frame is pointing radially from the center of the Earth to the current position on the reference orbit, i.e., along $\vec{r}_{\mathrm{GEO/E}}$, and the 2-axis points in the orbital track direction of the GEO reference orbit. The 3-axis completes the right hand rule, pointing out of the equatorial plane in the GEO case. The spacecraft position vector relative to the GEO reference orbit, resolved in Hill's frame, is denoted by $\vec{r}_{\mathrm{SC/GEO}}|_{\mathcal{H}} = [r_1, r_2, r_3]^\top$. Similarly, the spacecraft velocity relative to the reference orbit with respect to Hill's frame, resolved in Hill's frame, is $\vec{v}_{\mathrm{SC/GEO}/\mathcal{H}}|_{\mathcal{H}} = [v_1, v_2, v_3]^\top$. Using Euler's forward method with $\Delta t = 500$ sec, the discrete-time nonlinear spacecraft model is obtained from the continuous-time nonlinear model, derived in Appendix C, yielding

$$\begin{bmatrix} r_{1,t+1} \\ r_{2,t+1} \\ r_{3,t+1} \\ v_{1,t+1} \\ v_{2,t+1} \\ v_{3,t+1} \end{bmatrix} = \begin{bmatrix} r_{1,t} \\ r_{2,t} \\ r_{3,t} \\ v_{1,t} \\ v_{2,t} \\ v_{3,t} \end{bmatrix} + \Delta t \begin{bmatrix} v_{1,t} \\ v_{2,t} \\ v_{3,t} \\ a_{1,t} \\ a_{2,t} \\ a_{3,t} \end{bmatrix}, \tag{3.23}$$

where, using $r_t = \sqrt{(r_{1,t} + r_0)^2 + r_{2,t}^2 + r_{3,t}^2}$,

$$\begin{aligned} a_{1,t} &= -\frac{\mu_{\mathrm{E}}(r_{1,t} + r_0)}{r_t^3} + 2n_0 v_{2,t} + n_0^2 r_{1,t} + \frac{\mu_{\mathrm{E}}}{r_0^2} + \frac{F_{1,t}}{m_{\mathrm{SC}}} + d_{\mathrm{p},1,t}, \\ a_{2,t} &= -\frac{\mu_{\mathrm{E}} r_{2,t}}{r_t^3} - 2n_0 v_{1,t} + n_0^2 r_{2,t} + \frac{F_{2,t}}{m_{\mathrm{SC}}} + d_{\mathrm{p},2,t}, \\ a_{3,t} &= -\frac{\mu_{\mathrm{E}} r_{3,t}}{r_t^3} + \frac{F_{3,t}}{m_{\mathrm{SC}}} + d_{\mathrm{p},3,t}. \end{aligned} \tag{3.24}$$

The control variables F_1, F_2, and F_3 are thrust forces projected on the axes of Hill's frame,

$$u = [F_1, F_2, F_3]^\top. \tag{3.25}$$

Likewise, $d_{\mathrm{p},1}$, $d_{\mathrm{p},2}$, and $d_{\mathrm{p},3}$ are time-dependent perturbations acting along the axes of Hill's frame according to (C.7)–(C.11), where disturbances due to luni-solar gravity, SRP, and J2 are taken into account here. In all simulations, the initial positions of Earth, Moon, and Sun are as of September 3, 2015, at 12 am (CT). The other parameters in (3.23) are the spacecraft mass m_{SC}, Earth's gravitational parameter μ_{E}, the GEO radius $r_0 = 42160$ km, and the GEO angular rate n_0, see (C.4). Since the fuel mass is assumed to be much smaller than m_{SC}, m_{SC} is considered constant.

In addition to the six states in (3.23), another state is introduced that takes account of the available fuel. To normalize fuel consumption, the accumulated Δv (total change in spacecraft velocity) due to accelerations generated by the thrust forces is introduced, i.e.,

$$\Delta v_{\mathrm{acc},t+1} = \Delta v_{\mathrm{acc},t} + \Delta v_t = \Delta v_{\mathrm{acc},t} + \Delta t \frac{\|u_t\|_1}{m_{\mathrm{SC}}}, \tag{3.26}$$

where $\|\cdot\|_1$ denotes the 1-norm. In summary, the discrete-time nonlinear spacecraft model is given by (3.23) and (3.26), where the control input vector is given by (3.25) and the state vector is

$$x = [r_1, r_2, r_3, v_1, v_2, v_3, \Delta v_{\mathrm{acc}}]^\top. \tag{3.27}$$

The MATLAB function $\mathrm{fmincon}$ (with the interior-point algorithm) is used to solve NLP (3.17). The time horizon N and the lower bound τ_{lb} in NLP (3.17) are chosen based on the open-loop solution to the linearized problem obtained by Algorithm 3.1. Moreover, the solution of Algorithm 3.1 serves as an initial guess to the nonlinear solver.

DCOC Problem

A station keeping window of ± 0.01 degrees in longitude and latitude is considered, which is an order of magnitude smaller compared to traditional station keeping approaches [82, 83, 90]. Note that future missions may require such small windows due to the growing number of GEO satellites. The chosen constraints on longitude and latitude approximately translate into position constraints of ± 7.4 km for r_1, r_2, and r_3. Hence, the set one wants the state vector to remain inside for as long as possible is given by

$$G = \{x \in \mathbb{R}^7 : \Delta v_{\mathrm{acc}} \in [0, \Delta v_{\mathrm{acc,max}}],\ r_i \in [-7.4, 7.4]\,\mathrm{km},\ i \in \{1, 2, 3\}\}, \tag{3.28}$$

where $\Delta v_{\mathrm{acc,max}}$ is a prescribed maximum value for the accumulated Δv, equivalent to the amount of fuel that is initially available to the control system. The tightened constraints $\tilde{G}_t \equiv \tilde{G}$ for the MPC implementation (see Figure 3.1) are obtained by reducing the position window by 0.1 %, yielding

$$\tilde{G} = \{x \in \mathbb{R}^7 : \Delta v_{\mathrm{acc}} \in [0, \Delta v_{\mathrm{acc,max}}],\, r_i \in [-7.3926, 7.3926]\,\mathrm{km},\, i \in \{1,2,3\}\}. \quad (3.29)$$

The satellite is equipped with six thrusters, where each thruster can generate a maximum thrust force of $F_{\mathrm{th}} = 0.1$ N, which is similar to the ion thruster discussed in [91]. Each thruster is assumed to point in one of the directions of Hill's frame (positive and negative directions). Thus, in the case of continuous-thrust control, the control constraints are given by

$$U_{\mathrm{cont}} = \{u \in \mathbb{R}^3 : F_i \in [-F_{\mathrm{th}}, F_{\mathrm{th}}],\, i \in \{1,2,3\}\}. \quad (3.30)$$

In the on/off-thrust case, each thruster can apply the discrete values $\{-F_{\mathrm{th}}, 0, F_{\mathrm{th}}\}$ and, consequently, the set of control constraints reads

$$U_{\mathrm{on/off}} = \{u \in \mathbb{Z}^3 : F_i \in \{-F_{\mathrm{th}}, 0, F_{\mathrm{th}}\},\, i \in \{1,2,3\}\}. \quad (3.31)$$

In the following, both continuous-thrust and on/off-thrust are investigated. A spacecraft mass of $m_{\mathrm{SC}} = 4000$ kg is assumed and the parameters for the SRP disturbance model in (C.10) are $C_{\mathrm{srp}} = 9.1 \times 10^{-6}\,\mathrm{N/m^2}$, $c_{\mathrm{refl}} = 0.6$, and $S_{\mathrm{SC}} = 200\,\mathrm{m^2}$.

Linear Model

The linear discrete-time model for the MPC implementation is obtained by linearizing the continuous-time nonlinear model in (C.6) about the GEO reference orbit, yielding the CW equations [65], and employing Euler's forward method to transform the continuous-time model into discrete-time. Furthermore, the nonlinear evolution of the accumulated Δv in (3.26) is approximated by introducing the auxiliary variables

$$\zeta = [\zeta_1, \zeta_2, \zeta_3]^\top, \quad (3.32)$$

and augmenting the linear discrete-time dynamics in (3.7) as follows

$$x_{t+1} = Ax_t + B \begin{bmatrix} u_t \\ \zeta_t \end{bmatrix} + d_t, \quad (3.33)$$

where the matrices A and B are stated in (3.36) and (3.37), respectively. Moreover, the

constraints

$$-\zeta_t \leq u_t \leq \zeta_t, \text{ for } t \in \{0, 1, ..., N-1\}, \tag{3.34}$$

are added to LP (3.15), (3.19), and (3.20), and the weighted sum of ζ_t values is added to the respective objective function. Thus, in the case of LP (3.15) or (3.19), the objective function is modified to

$$\sum_{t=\tau_{\mathrm{lb}}}^{N} \varepsilon_t + w \sum_{t=0}^{N-1} \mathbf{1}^\top \zeta_t, \tag{3.35a}$$

and in the case of LP (3.20), the objective function is modified to

$$\sum_{t=1}^{N_{\mathrm{recover}}} \varepsilon_t + w \sum_{t=0}^{N_{\mathrm{recover}}-1} \mathbf{1}^\top \zeta_t, \tag{3.35b}$$

where $w > 0$ is a weight that is set to $w = 0.005$. The matrices of the linear discrete-time model in (3.33) are as follows

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 3n_0^2\Delta t & 0 & 0 & 1 & 2n_0\Delta t & 0 & 0 \\ 0 & 0 & 0 & -2n_0\Delta t & 1 & 0 & 0 \\ 0 & 0 & -n_0^2\Delta t & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.36}$$

$$B = \begin{bmatrix} & & & 0_{3\times 6} & & \\ \frac{\Delta t}{m_{\mathrm{SC}}} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\Delta t}{m_{\mathrm{SC}}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\Delta t}{m_{\mathrm{SC}}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\Delta t}{m_{\mathrm{SC}}} & \frac{\Delta t}{m_{\mathrm{SC}}} & \frac{\Delta t}{m_{\mathrm{SC}}} \end{bmatrix}, \tag{3.37}$$

where n_0 is defined in (C.4) and the sampling time is $\Delta t = 500$ sec.

Following [80], the time-varying disturbance term d_t in (3.33) is computed in advance for the known GEO reference orbit. This is achieved by replacing $\vec{r}_{\mathrm{M/SC}}$, $\vec{r}_{\mathrm{S/SC}}$, and $\vec{r}_{\mathrm{SC/E}}$ in (C.7)–(C.11) with $\vec{r}_{\mathrm{M/GEO}}$, $\vec{r}_{\mathrm{S/GEO}}$, and $\vec{r}_{\mathrm{GEO/E}}$, respectively, where, instead of the spacecraft position (SC), the known trajectory of the GEO reference orbit is used. Thus, the disturbance accelerations for the GEO reference orbit are obtained at each time instant and d_t follows from multiplying these accelerations by the sampling time Δt (Euler's forward

method): $d_t = [0_{1\times 3}, \Delta t \bar{d}_{\mathrm{p},t}^\top, 0]^\top$, where $\bar{d}_{\mathrm{p},t}$ is the instantaneous disturbance vector for the GEO reference orbit, resolved in Hill's frame, according to (C.7).

The numerical conditioning of each LP is improved by normalizing the state vector by $x_{\mathrm{norm}} = O_{\mathrm{tf}} x + o_{\mathrm{tf}}$, where O_{tf} and o_{tf} are such that the state constraints defined in (3.29) are normalized as follows

$$\tilde{G}_{\mathrm{norm}} = \{x_{\mathrm{norm}} \in \mathbb{R}^7 : \Delta v_{\mathrm{acc,norm}} \in [0,1],\, r_{\mathrm{norm},i} \in [0,1],\, i \in \{1,2,3\}\},$$

and $v_i \in [-1, 1]\,\mathrm{m/sec}$ corresponds to $v_{\mathrm{norm},i} \in [0,1]$, $i \in \{1,2,3\}$. This improves the numerical conditioning of each LP and increases robustness and reduces computation times. The parameters in Algorithm 3.3 are set to $N_{\mathrm{ub}} = 600$, $\alpha^{\mathrm{LP}} = 30$, $N_{\mathrm{recover}} = 5$, and $\tau_{\mathrm{lb},0} = 300$ as an initial guess in Step 2 of Algorithm 3.3.

Remark 3.2. *If the optimal first exit-time of the linear problem is greater than the time horizon N of LP* (3.15) *or* (3.19)*, $\varepsilon_t \equiv 0$ for proper choices of w and the respective objective function, see* (3.35)*, becomes $w \sum_{t=0}^{N-1} \mathbf{1}^\top \zeta_t$. In this case, due to the additional constraints in* (3.34)*, the respective LP solution is equivalent to the minimum-fuel solution for the linear model that minimizes $\sum_{t=0}^{N-1} \|u_t\|_1$. On the other hand, if N is greater than the optimal first exit-time of the linear problem, there exists t^* such that $\varepsilon_t > 0$ for all $t \in [t^*, N]$, and the respective LP solution may be different from the minimum-fuel solution. In fact, since maximizing the first exit-time is not explicitly emphasized by the minimum-fuel solution, constraint violation may occur earlier than for the DCOC-based solution.*

Results for Continuous-Thrust

For the continuous-thrust case, the initial condition

$$x_0 = [0, 0, -5\,\mathrm{km}, 0, -0.4\,\mathrm{m/sec}, 0, 0]^\top, \tag{3.38}$$

is considered, and the maximum value for the accumulated Δv in (3.28) is chosen as $1\,\mathrm{m/sec}$. The results of the NLP-Discrete-Time and MPC-Discrete-Time simulations are plotted in Figure 3.4, where the dashed lines indicate the state and control constraints. Figure 3.4 also shows the results of the MPC-Continuous-Time simulation (the continuous-time nonlinear model is derived in Appendix C). For each case, constraint violation occurs as a consequence of reaching the prescribed fuel limit or, equivalently, the limit on Δv_{acc}. The trajectories for NLP-Discrete-Time and MPC-Discrete-Time simulations in Figure 3.4 are similar and constraint violation occurs after 415 time steps (2.4 days) for both approaches. This shows that the LP-based MPC scheme can be effective in the context of

DCOC of a nonlinear system.

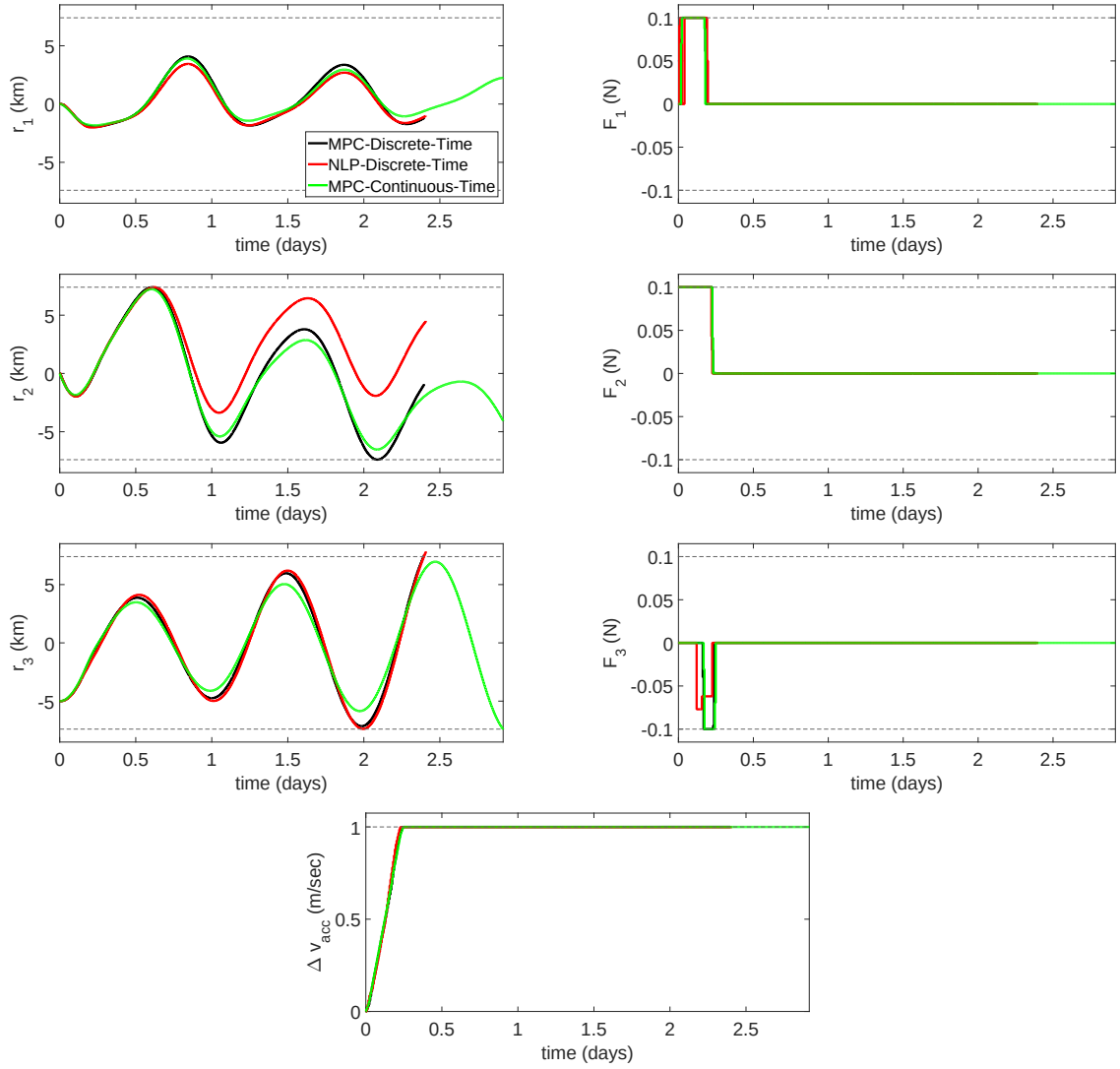

Figure 3.4: GEO satellite station keeping problem, continuous-thrust case: spacecraft position relative to GEO reference orbit, thrust forces in Hill's frame, and accumulated Δv vs. time.

Applying the MPC strategy to the continuous-time nonlinear model results in control trajectories similar to the MPC-Discrete-Time solution. However, the continuous-time dynamics extend constraint violation to 2.92 days, which is about 22 % greater than observed in simulations on the discrete-time model, which may be due to the relatively large sampling time ($\Delta t = 500$ sec) used for the discrete-time dynamics. The computation times are as follows. About 53 min are required to solve NLP (3.17) with the MATLAB function fmincon. In both MPC-Discrete-Time and MPC-Continuous-Time simulations, the MPC

strategy (Algorithm 3.3) requires on average 1.4 sec to compute the control input at each time instant with a worst-case computation time of 16 sec (which is smaller than $\Delta t = 500$ sec).

Results for On/Off-Thrust

In the on/off-thrust case, LPs (3.15), (3.19), and (3.20) become MILPs and NLP (3.17) becomes an MINLP due to the discrete control inputs, see (3.31). However, since solving mixed-integer programs is less robust than solving standard LPs, a different approach to handle on/off-thrust is proposed.

As in the continuous-thrust case, a control $u_t = [F_{1,t}, F_{2,t}, F_{3,t}]^\top \in U_{\text{cont}}$, see (3.30), is computed at each time instant $t \in \mathbb{Z}_{\geq 0}$ using the LP-based MPC implementation in Algorithm 3.3 with a sampling time of $\Delta t = 500$ sec. The continuous-thrust control u_t is transformed into discrete thrust values based on thrust impulse equivalence, i.e.,

$$F_{i,t}\Delta t = \text{sgn}(F_{i,t})F_{\text{th}}t_{\text{th},i}, \tag{3.39}$$

for each $i \in \{1, 2, 3\}$, where $t_{\text{th},i} \in [0, \Delta t]$ denotes the time of thrusting with $\text{sgn}(F_{i,t})F_{\text{th}}$ in the i-direction of Hill's frame and $F_{\text{th}} = 0.1$ N is the force generated by the respective on/off thruster. Since $t_{\text{th},i} \in [0, \Delta t]$, simulations need to be performed on the continuous-time model. During each $\Delta t = 500$ sec sampling interval, the following thrust forces are applied to the continuous-time nonlinear model,

$$F_{\text{on/off},i}(\tau_{\Delta t}) = \begin{cases} \text{sgn}(F_{i,t})0.1\,\text{N}, & \text{for } \tau_{\Delta t} \in [0, t_{\text{th},i}], \\ 0, & \text{otherwise}, \end{cases} \tag{3.40}$$

where $\tau_{\Delta t} \in [0, \Delta t]$, $t_{\text{th},i} = (|F_{i,t}|/0.1\,\text{N})\Delta t$ according to (3.39), $i \in \{1, 2, 3\}$ denotes the respective direction of Hill's frame, and $u_t = [F_{1,t}, F_{2,t}, F_{3,t}]^\top$ is provided by the MPC strategy (Algorithm 3.3).

For the initial states in (3.38) and $\Delta v_{\text{acc,max}} = 1\,\text{m/sec}$, the proposed on/off-thrust MPC strategy (in MPC-Continuous-Time simulation) generates the same first exit-time of 2.92 days as the continuous-thrust control strategy in Figure 3.4. This result is expected since both strategies are based on the MPC implementation in Algorithm 3.3 and the same thrust impulses are applied to the system.

Now let the initial state vector be $x_0 = [0, 0, 0, 0, 0, 0, 0]^\top$ and $\Delta v_{\text{acc,max}} = 10\,\text{m/sec}$, i.e., 10 times as much fuel as in the continuous-thrust case (Figure 3.4) is available. Figure 3.5 shows the MPC-Continuous-Time solution in the on/off-thrust case. Constraint violation occurs for the first time after 85 days. Due to the chosen upper bound, $N_{\text{ub}} = 600$,

on the LP time horizon in Algorithm 3.3, computation times are feasible and smaller than the sampling time $\Delta t = 500$ sec. On average, 11.2 sec are required to compute the control input at each time/sampling instant. The worst-case computation time of 63.3 sec occurs in the beginning when $\tau_{\mathrm{lb},0} = 300$ and several iterations are required in Algorithm 3.3 for N to exceed N_{ub}. The longer computation times of the MPC strategy in this case are due to the longer time horizon, i.e., greater first exit-time.

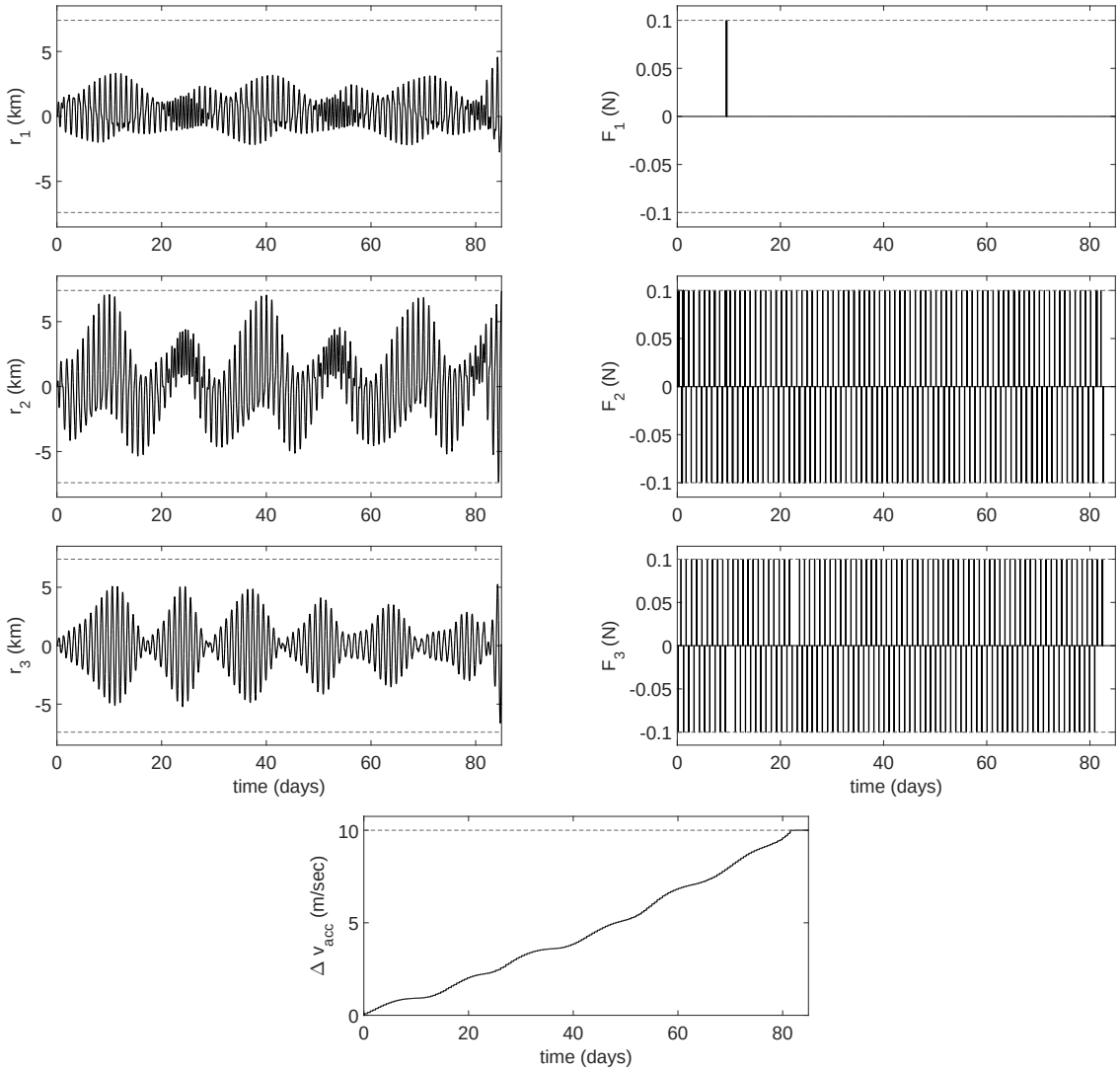

Figure 3.5: GEO satellite station keeping problem, MPC-Continuous-Time simulation in the on/off-thrust case: spacecraft position relative to GEO reference orbit, thrust forces in Hill's frame, and accumulated Δv vs. time.

Note that the proposed MPC strategy (in MPC-Continuous-Time simulation, using on/off-thrust) is able to satisfy the prescribed constraints for more than one year with an

initial amount of fuel equivalent to $\Delta v_{\mathrm{acc,max}} = 51.7\,\mathrm{m/sec}$. This is a nearly 18 % improvement in efficiency compared to a recently published GEO station keeping strategy that used quadratic-cost MPC [80], where, for a similar model using the same set of parameters, an accumulated Δv of $61\,\mathrm{m/sec}$ was required to satisfy the same position constraints [Eq. (3.28)] for one year. Furthermore, when increasing the size of the station keeping window to ± 0.05 degrees in longitude and latitude, the proposed MPC strategy achieves 420 days without constraint violation for $\Delta v_{\mathrm{acc,max}} = 51.7\,\mathrm{m/sec}$. This result suggests that, for the larger station keeping window, the proposed MPC strategy is able to perform one year of station keeping with a Δv_{acc} of approximately $(365/420)51.7\,\mathrm{m/sec} = 44.9\,\mathrm{m/sec}$.

3.4.2.2 Spacecraft Attitude Control 2

Nonlinear Model

The continuous-time nonlinear model for the spacecraft attitude control problem is derived in Appendix D. Euler's forward method is used to obtain the discrete-time model from the continuous-time model for a chosen sampling time of $\Delta t = 2$ sec. The state vector at a time instant $t \in \mathbb{Z}_{\geq 0}$ is given by

$$x_t = [\phi_t, \theta_t, \psi_t, \omega_{1,t}, \omega_{2,t}, \omega_{3,t}, \nu_{1,t}, ..., \nu_{p,t}]^\top, \tag{3.41}$$

where ϕ_t, θ_t, and ψ_t are the 3-2-1 Euler angles, $\bar{\omega}_{\mathcal{B}/\mathcal{I},t} = [\omega_{1,t}, \omega_{2,t}, \omega_{3,t}]^\top$ is the spacecraft angular velocity vector (angular velocity of the spacecraft body-fixed frame $\mathcal{B}$ relative to an inertial reference frame $\mathcal{I}$) expressed in the body-fixed frame, and $\bar{\nu}_t = [\nu_{1,t}, ..., \nu_{p,t}]^\top$ contains the spin rates of the p RWs. The RW accelerations serve as control variables and the control input vector for the discrete-time model is given by the instantaneous RW accelerations, i.e., $u_t = [\dot{\nu}_{1,t}, ..., \dot{\nu}_{p,t}]^\top$. Thus, the discrete-time nonlinear model is as follows

$$\begin{bmatrix} \phi_{t+1} \\ \theta_{t+1} \\ \psi_{t+1} \\ \bar{\omega}_{\mathcal{B}/\mathcal{I},t+1} \\ \bar{\nu}_{t+1} \end{bmatrix} = \begin{bmatrix} \phi_t \\ \theta_t \\ \psi_t \\ \bar{\omega}_{\mathcal{B}/\mathcal{I},t} \\ \bar{\nu}_t \end{bmatrix} + \Delta t \begin{bmatrix} \begin{bmatrix} 1 & \mathrm{s}(\phi_t)\mathrm{t}(\theta_t) & \mathrm{c}(\phi_t)\mathrm{t}(\theta_t) \\ 0 & \mathrm{c}(\phi_t) & -\mathrm{s}(\phi_t) \\ 0 & \mathrm{s}(\phi_t)/\mathrm{c}(\theta_t) & \mathrm{c}(\phi_t)/\mathrm{c}(\theta_t) \end{bmatrix} \bar{\omega}_{\mathcal{B}/\mathcal{I},t} \\ \bar{\alpha}_{\mathcal{B}/\mathcal{I},t} \\ u_t \end{bmatrix}, \tag{3.42}$$

where $\mathrm{c}(\cdot) = \cos(\cdot)$, $\mathrm{s}(\cdot) = \sin(\cdot)$, $\mathrm{t}(\cdot) = \tan(\cdot)$, and $\bar{\alpha}_{\mathcal{B}/\mathcal{I},t} = \dot{\bar{\omega}}_{\mathcal{B}/\mathcal{I},t}$ according to (D.5), i.e.,

$$\bar{\alpha}_{\mathcal{B}/\mathcal{I},t} = \bar{J}^{-1}(\bar{\tau}_{\mathrm{srp},t} - S[\bar{\omega}_{\mathcal{B}/\mathcal{I},t}](\bar{J}\bar{\omega}_{\mathcal{B}/\mathcal{I},t} + J_{\mathrm{w}} W \bar{\nu}_t) - J_{\mathrm{w}} W u_t), \tag{3.43}$$

where S is defined in (D.1). The SRP disturbance torque $\bar{\tau}_{\text{srp}}$ in (3.43) is a nonlinear function of the spacecraft orientation, see (D.7)–(D.9). The other parameters of the model are the moment of inertia matrix of the spacecraft bus, J, the moment of inertia of each RW (assuming identical RWs) about its spin axis, J_{w}, and the locked inertia $\bar{J}$ defined in (D.4), as well as the orientations of the RW spin axes given by W, see (D.3). The spacecraft parameters, adopted from [79], are listed in Table 3.3. Note that the vector representing the direction of the Sun is resolved in the inertial frame $\mathcal{I}$ in Table 3.3 and needs to be transformed into the spacecraft body-fixed frame $\mathcal{B}$ continuously using the current orientation of the spacecraft.

Parameter	Units	Value
J	kgm^2	$\text{diag}(430, 1210, 1300)$
J_{w}	kgm^2	0.043
L_x, L_y, L_z	m	2, 2.5, 5
l_x, l_y, l_z	m	0, 0.5, 0
$\hat{q}_{\text{S}}\|_{\mathcal{I}}$	-	$[0, 1/\sqrt{2}, 1/\sqrt{2}]^\top$
Φ_{S}	W/m^2	1367
c	m/sec	299,792,458
C_{diff}	-	0.2

Table 3.3: Model parameters for spacecraft attitude control problem.

In this case study, the NLP in (3.17) is solved with the MATLAB function fmincon using the sequential quadratic programming (SQP) algorithm with the open-loop solution of the linearized problem (obtained by Algorithm 3.1) as an initial guess.

DCOC Problem

The DCOC problem is formulated with the objective to satisfy prescribed constraints on spacecraft orientation and RW spin rates for as long as possible. The constraints define the set

$$\begin{aligned} G = \{x \in \mathbb{R}^{6+p} : \phi \in [\phi_{\min}, \phi_{\max}],\, \theta \in [\theta_{\min}, \theta_{\max}],\, \psi \in [\psi_{\min}, \psi_{\max}], \\ \nu_i \in [\nu_{i,\min}, \nu_{i,\max}],\, i \in \{1, 2, ..., p\}\}, \end{aligned} \tag{3.44}$$

where $\phi_{\min} < 0$, $\theta_{\min} < 0$, $\psi_{\min} < 0$, $\phi_{\max} > 0$, $\theta_{\max} > 0$, $\psi_{\max} > 0$, and $\nu_{i,\min} \leq \nu_{i,\max} \in \mathbb{R}$, $i \in \{1, 2, ..., p\}$. Similar to the GEO station keeping problem in Section 3.4.2.1, the state constraints are tightened for the LP-based MPC implementation as illustrated in Figure 3.1

by reducing the orientation constraints by 0.4 %. This yields the following reduced set

$$\begin{aligned}\tilde{G} = \{x \in \mathbb{R}^{6+p} : 1.004\phi \in [\phi_{\min}, \phi_{\max}],\ 1.004\theta \in [\theta_{\min}, \theta_{\max}],\\ 1.004\psi \in [\psi_{\min}, \psi_{\max}],\ \nu_i \in [\nu_{i,\min}, \nu_{i,\max}],\ i \in \{1, 2, ..., p\}\}.\end{aligned} \tag{3.45}$$

In the following case studies, the attitude and RW spin rate constraints in (3.44) and (3.45), respectively, are given by

$$\phi_{\min} = \theta_{\min} = -0.00175\,\text{rad},\ \psi_{\min} = -0.0175\,\text{rad}, \tag{3.46a}$$

$$\phi_{\max} = \theta_{\max} = 0.00175\,\text{rad},\ \psi_{\max} = 0.0175\,\text{rad}, \tag{3.46b}$$

$$\nu_{i,\min} = 10\,\text{rad/sec},\ \nu_{i,\max} = 250\,\text{rad/sec},\ i \in \{1, 2, ..., p\}. \tag{3.46c}$$

Note that the lower bound on the RW spin rates is chosen to avoid zero speed crossings and increase in RW wear and power consumption at low speeds. The initial states of the spacecraft are assumed to be

$$[\phi_0, \theta_0, \psi_0] = [-0.001, 0.00035, -0.0105]\,\text{rad}, \tag{3.47a}$$

$$[\omega_{1,0}, \omega_{2,0}, \omega_{3,0}] = [3.5, 3.5, 35] \times 10^{-5}\,\text{rad/sec}. \tag{3.47b}$$

The maximum angular acceleration of each RW is $4\,\text{rad/sec}^2$. Hence, $U = \{u \in \mathbb{R}^p : u_i \in [-4, 4]\,\text{rad/sec}^2,\ i \in \{1, 2, ..., p\}\}$.

Linear Model

The linear discrete-time model for the LP-based MPC implementation (Algorithm 3.3) is obtained by linearizing the continuous-time nonlinear model (derived in Appendix D) and employing Euler's forward method with a sampling time of $\Delta t = 2$ sec. The initial RW spin rates $\bar{\nu}_0$ and $\phi = \theta = \psi = 0$ are chosen as the reference for the linear model. The matrices and the disturbance term of the linear discrete-time model are therefore given by

$$A = \begin{bmatrix} I_{3\times3} & \Delta t I_{3\times3} & 0_{3\times p} \\ \Delta t \bar{J}^{-1} T & I_{3\times3} + \Delta t \bar{J}^{-1} J_{\text{w}} S[W\bar{\nu}_0] & 0_{3\times p} \\ 0_{p\times3} & 0_{p\times3} & I_{p\times p} \end{bmatrix}, \tag{3.48}$$

$$B = \begin{bmatrix} 0_{3\times p} \\ -\Delta t \bar{J}^{-1} J_{\text{w}} W \\ \Delta t I_{p\times p} \end{bmatrix},\quad d = \begin{bmatrix} 0_{3\times1} \\ \Delta t \bar{J}^{-1} \bar{\tau}_{\text{srp}}|_{\phi=\theta=\psi=0} \\ 0_{p\times1} \end{bmatrix}, \tag{3.49}$$

where $S[\cdot]$ is the skew-symmetric matrix defined in (D.1) and $\bar{\tau}_{\text{srp}}|_{\phi=\theta=\psi=0}$ is the SRP

torque when $\phi = \theta = \psi = 0$. Furthermore, $T \in \mathbb{R}^{3\times 3}$ in (3.48) results from numerically linearizing the SRP torque in (D.9) about $\phi = \theta = \psi = 0$, i.e.,

$$\bar{\tau}_{\text{srp}} \approx \bar{\tau}_{\text{srp}}|_{\phi=\theta=\psi=0} + T \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^\top.$$

In order to increase robustness and reduce computation times, the numerical conditioning of each LP is improved by normalizing the state vector x according to $x_{\text{norm}} = O_{\text{tf}} x + o_{\text{tf}}$, where O_{tf} and o_{tf} are such that the state constraints in (3.45) are normalized as follows

$$\begin{aligned}\tilde{G}_{\text{norm}} = \{x_{\text{norm}} \in \mathbb{R}^{6+p} : \phi_{\text{norm}} \in [0,1],\, \theta_{\text{norm}} \in [0,1],\, \psi_{\text{norm}} \in [0,1],\\ \nu_{\text{norm},i} \in [0,1],\, i \in \{1,2,...,p\}\},\end{aligned}$$

and $\omega_j \in [-10^{-3}, 10^{-3}]\,\text{rad/sec}$ corresponds to $\omega_{\text{norm},j} \in [0,1]$, $j \in \{1,2,3\}$.

Initial numerical results show that, compared to the NLP solution, the LP-based MPC strategy (in MPC-Discrete-Time and MPC-Continuous-Time simulations) yields similar first exit-times while, however, using substantially more control effort (see also middle and bottom plots in Figure 3.3), which may be undesirable. Hence, in order to avoid excessive control inputs, control inputs are penalized by considering the weighted sum of $\|u_t\|_1$ values as an additional objective to be minimized. As for example in [92], this is achieved by introducing the variables $\gamma_t \in \mathbb{R}^p$ for $t \in \{0,1,...,N-1\}$ and adding the weighted sum of γ_t values to the objective functions of LPs (3.19) and (3.20), yielding, respectively,

$$\sum_{t=\tau_{\text{lb}}}^{N} \varepsilon_t + w \sum_{t=0}^{N-1} \mathbf{1}^\top \gamma_t \text{ and } \sum_{t=1}^{N_{\text{recover}}} \varepsilon_t + w \sum_{t=0}^{N_{\text{recover}}-1} \mathbf{1}^\top \gamma_t, \tag{3.50}$$

where the weight is set to $w = 0.005$ here. Moreover, the constraint $-\gamma_t \le u_t \le \gamma_t$, $t \in \{0,1,...,N-1\}$, is added to LPs (3.19) and (3.20). This approach is similar to the linear approximation of the nonlinear dynamics of Δv_{acc} in the GEO satellite station keeping problem, see (3.26) and (3.32)–(3.34), and Remark 3.2 also holds in this case.

In the following, the parameters in Algorithm 3.3 are set to $N_{\text{ub}} = 200$, $N_{\text{add}} = 25$, $N_{\text{recover}} = 5$, and $\tau_{\text{lb},0} = 100$ (initial guess).

Results for One RW

First, the case of one operable RW ($p = 1$) with spin axis $\bar{g}_1 = [1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}]^\top$ (resolved in the spacecraft body-fixed frame) is considered. The set of state constraints and initial condition of the spacecraft for this case study are given by (3.46) and (3.47),

respectively, and the initial RW speed is assumed to be $\nu_0 = 100\,\mathrm{rad/sec}$. Figure 3.6 shows the trajectories of the Euler angles, RW speed, and control input for the NLP-Discrete-Time and MPC-Discrete-Time simulations as well as for the MPC-Continuous-Time simulation (where a zero-order hold is applied to the control during each 2 sec sampling interval). The constraints are indicated by gray dashed lines in Figure 3.6. Both the NLP-Discrete-Time and MPC-Discrete-Time solutions violate the constraints after 45 time steps, which is equivalent to 90 sec (1.5 min). The respective trajectories are similar but different, which indicates that the optimal solution to the DCOC problem in (3.5) or (3.6), respectively, may not be unique in this case.

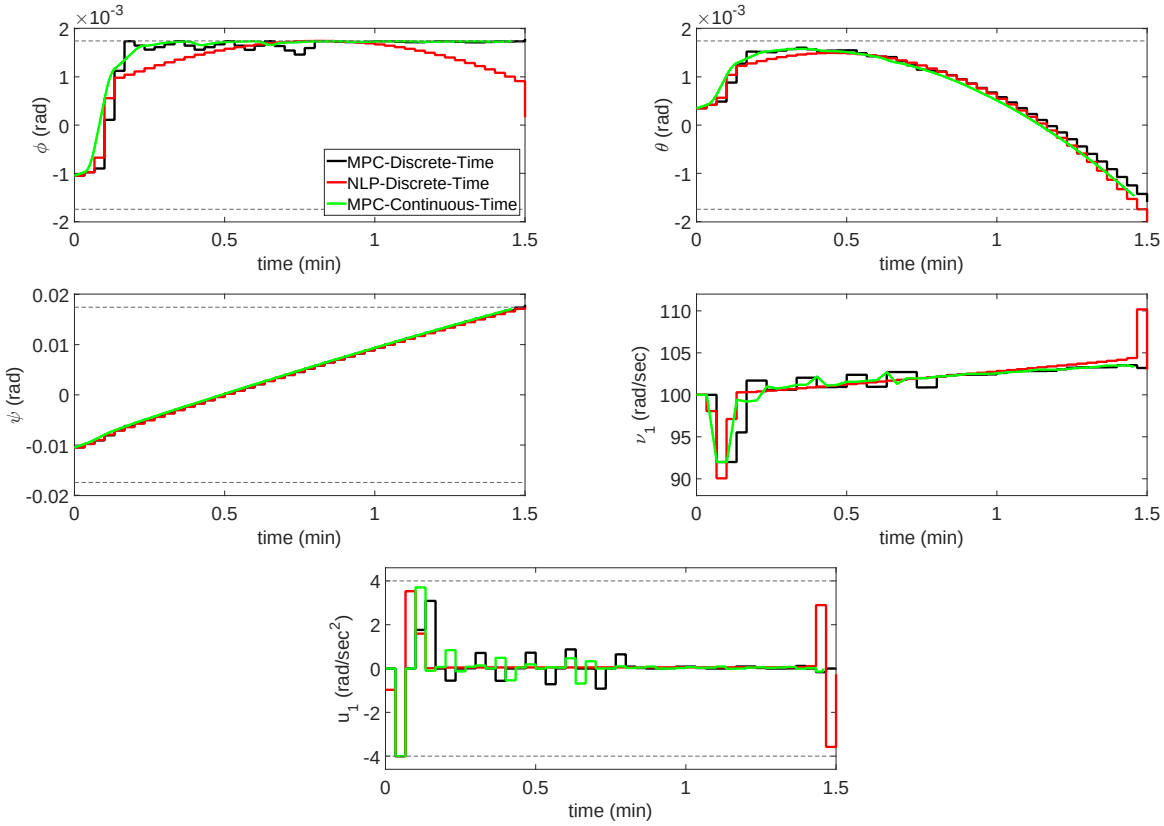

Figure 3.6: Spacecraft attitude control problem, one RW ($p = 1$): Euler angles, RW speed, and control input vs. time.

In MPC-Continuous-Time simulation, constraint violation occurs after 87.4 sec (1.46 min). The NLP solution is obtained in 49.9 sec with MATLAB's fmincon function, which is considerably faster compared to the GEO station keeping problem (Section 3.4.2.1) because of smaller first exit-times (and thus smaller time horizons). On average, the LP-based MPC implementation (in both MPC-Discrete-Time and MPC-Continuous-Time simulations) requires about 0.01 sec to compute the control input at each time instant and the

worst-case computation time is 0.08 sec.

Results for Two RWs

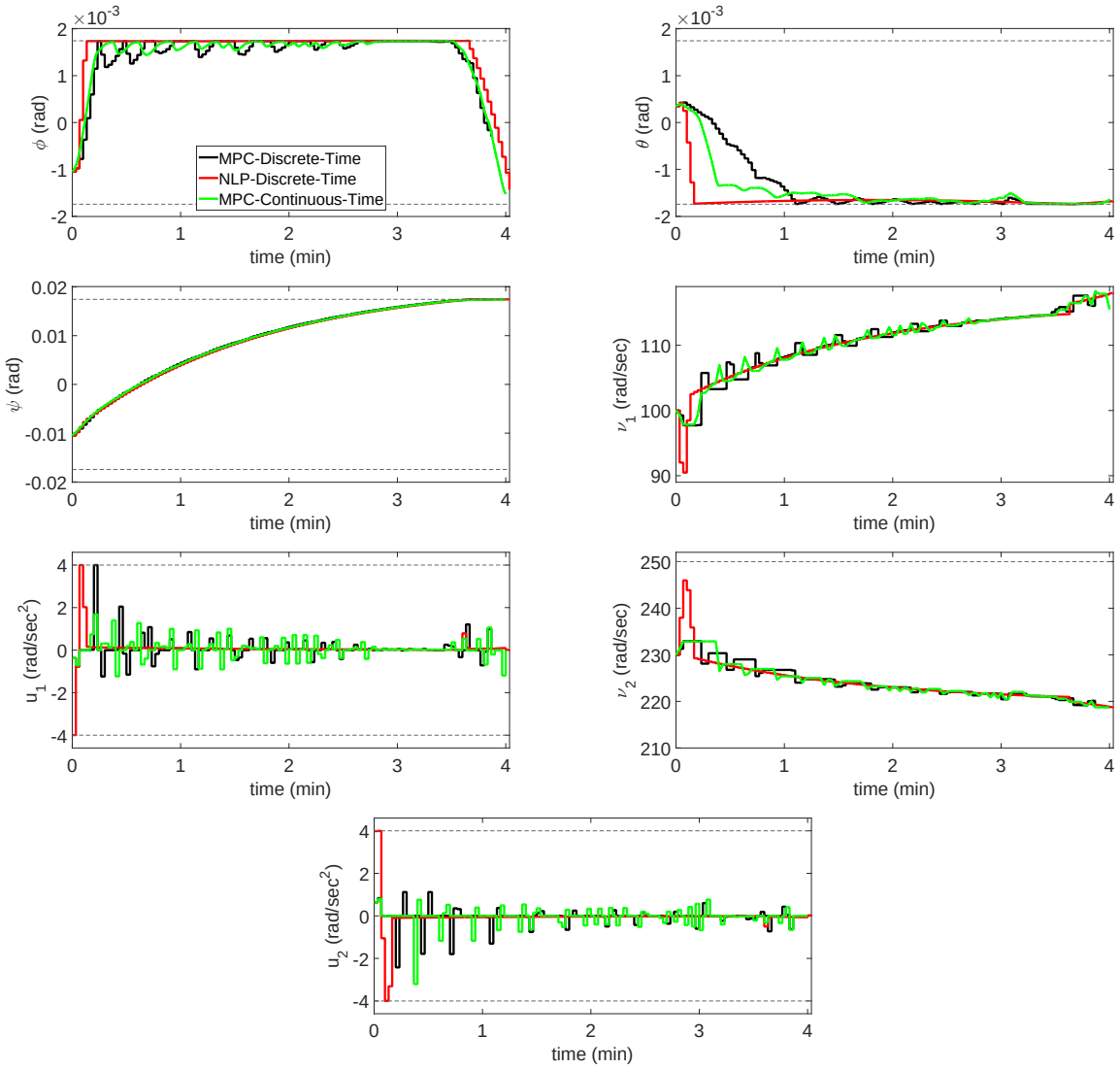

Figure 3.7: Spacecraft attitude control problem, two RWs ($p = 2$): Euler angles, RW speeds, and control inputs vs. time.

Now two operable RWs are assumed ($p = 2$), which increases the spacecraft's control authority compared to $p = 1$. The spacecraft has a second RW with spin axis $\bar{g}_2 = [0, 1, 0]^\top$ in addition to the RW with spin axis $\bar{g}_1 = [1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}]^\top$. The initial RW spin rates are given by $\nu_0 = [100, 230]^\top$ rad/sec.

The responses based on the LP-based MPC strategy (in MPC-Discrete-Time and MPC-

Continuous-Time simulations) as well as the NLP solution (in NLP-Discrete-Time simulation) for the state constraints and initial condition in (3.46) and (3.47), respectively, are plotted in Figure 3.7. The gray dashed lines in Figure 3.7 indicate the prescribed constraints. As for the case $p = 1$, the MPC solutions are close to the open-loop NLP solution. There are differences, however, as the MPC strategy exploits a linear model and control adjustments are required when the predicted trajectory differs from the actual trajectory due to unmodeled effects. The NLP solution violates constraints after 122 time steps or 244 sec (4.1 min). Similarly, the MPC-Discrete-Time simulation shows constraint violation after 117 time steps or 234 sec (3.9 min). Constraint violation occurs after 240 sec (4 min) in MPC-Continuous-Time simulation. In the worst-case, the MPC implementation (in both MPC-Discrete-Time and MPC-Continuous-Time simulations) requires 0.22 sec to compute the control u_t at a time instant $t \in \mathbb{Z}_{\geq 0}$ and 0.05 sec on average. The open-loop NLP solution, on the other hand, is computed in 136 sec.

Remark 3.3. *With two operable RWs, the linear spacecraft model with incorporated SRP torques becomes controllable under certain conditions [79]. Nevertheless, the spacecraft has not enough control authority (due to the constraints on the control input) in this case study to avoid violation of the relatively tight attitude constraints in* (3.46) *for the given initial condition in* (3.47)*. In addition, for all admissible control laws, the initial condition in* (3.47) *may be outside their respective regions of attraction.*

Results for Three RWs

In the case of three operable RWs ($p = 3$), the spacecraft is fully actuated. In this case, a third RW with spin axis $\bar{g}_3 = [0, 0, 1]^\top$ is added to the two RWs with spin axes $\bar{g}_1 = [1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}]^\top$ and $\bar{g}_2 = [0, 1, 0]^\top$. As before, the state constraints and initial condition are as in (3.46) and (3.47). Initial RW speeds of $\nu_0 = [100, 230, 249.7]^\top \, \mathrm{rad/sec}$ are assumed. Note that the third RW is initially near its saturation limit. Thus, the control authority is limited and, despite being fully actuated, constraint violation occurs in finite time for any admissible control law.

The NLP solution violates constraints after 216 time steps or 432 sec (7.2 min). For the MPC implementation, constraint violation occurs after 209 time steps or 418 sec (6.97 min) in the MPC-Discrete-Time case and after 419.6 sec (6.99 min) in the MPC-Continuous-Time case. These relatively large differences versus the NLP solution can be attributed to the weight w that emphasizes minimum control effort in LPs (3.19) and (3.20), see (3.50). The respective first exit-times are improved by reducing w from 0.005 to 0.001. This change results in constraint violation after 213 time steps or 424 sec (7.07 min) in

MPC-Discrete-Time simulation, which is within 1.5 % of the NLP solution. Furthermore, with the modified weight, constraint violation occurs after 426.9 sec (7.12 min) in MPC-Continuous-Time simulation. Further reducing w does not significantly improve the first exit-times. Figure 3.8 shows the trajectories of the NLP solution as well as of the MPC solutions (in MPC-Discrete-Time and MPC-Continuous-Time simulations) for $w = 0.001$.

A computation time of 461 sec is required to obtain the NLP solution. For the LP-based MPC implementation, the worst-case time to compute the control is 1.12 sec and 0.14 sec are required on average. Thus, for the cases considered ($p = 1$, $p = 2$, and $p = 3$), worst-case computation times are below the sampling time of $\Delta t = 2$ sec. Computation times can be further reduced by increasing Δt and/or reducing the upper bound, N_{ub}, on the time horizon of LP (3.19) in Algorithm 3.3, which may, however, reduce the control performance (i.e., lead to earlier constraint violation).

3.5 Summary

In this Chapter, mathematical programs were developed that lead to open-loop solutions of deterministic DCOC problems with the objective of maximizing the first exit-time (i.e., the time until prescribed constraints are violated for the first time). For DCOC problems where the system dynamics are described by a linear model, an MILP and a standard LP were developed that obtain optimal and good-quality suboptimal solutions, respectively. Similar programs were developed for the nonlinear case. Moreover, an iterative procedure was presented that efficiently updates the time horizon of the respective mathematical program until a proper solution is found. Based on linear model approximation of the system and the open-loop solution provided by the LP, a computationally efficient MPC strategy was developed, providing state feedback in order to compensate for unmodeled effects online. In several numerical case studies, involving a VDP oscillator, GEO satellite station keeping, and spacecraft attitude control, the developed mathematical programs and algorithmic procedures successfully obtained optimal as well as good-quality suboptimal solutions to the respective DCOC problems. In particular, the solutions of the LP-based MPC strategy were close to the nonlinear programming solutions.

Compared to the DP-based techniques in Chapter 2 which are quite general and can address a broad range of DCOC problems (see Section 2.1), the mathematical programs and MPC scheme developed in this chapter do not suffer from the curse of dimensionality of DP and higher-dimensional problems can readily be treated (e.g., problems with 9 states and 3 control variables such as the spacecraft attitude control problem with 3 RWs in Section 3.4.2.2). An additional advantage is that the mathematical programs and MPC scheme

facilitate the use of continuous control inputs in a numerical setting.

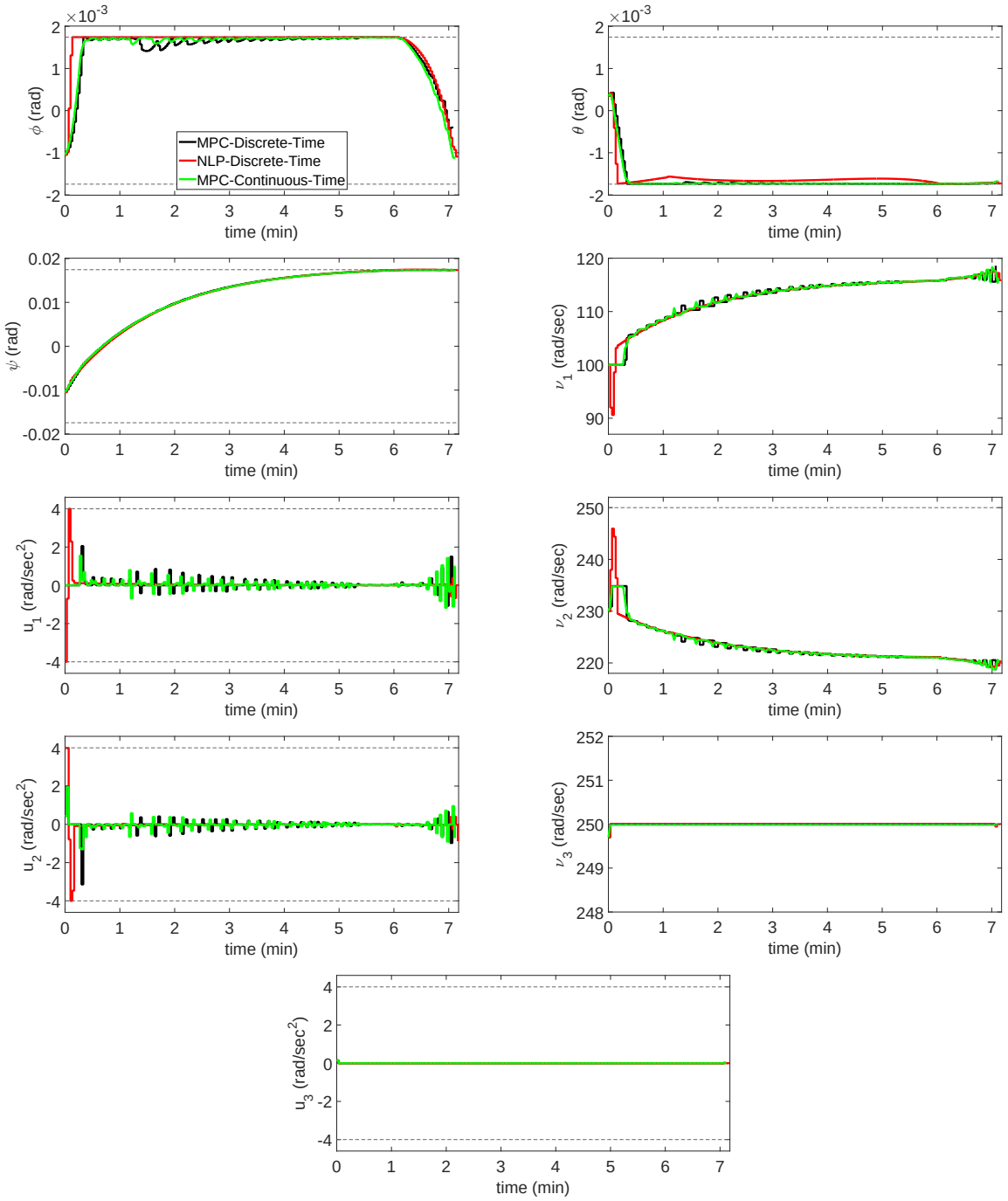

Figure 3.8: Spacecraft attitude control problem, three RWs ($p = 3$): Euler angles, RW speeds, and control inputs vs. time.

CHAPTER 4

Stochastic DCOC – DP Approaches

4.1 Problem Formulation

The problems considered here and in Chapter 5 are extensions of deterministic DCOC (Chapters 2 and 3) to the case of stochastic systems. The focus in this chapter is on the following class of discrete-time stochastic nonlinear systems,

$$x_{t+1} = f(x_t, u_t, w_t), \tag{4.1}$$

where f is a general nonlinear function, $t \in \mathbb{Z}_{\geq 0}$ denotes the time instant, $x \in \mathbb{R}^n$ is the state vector, and w is a random disturbance modeled by a Markov chain, taking values in the finite set

$$W = \{w^1, w^2, ..., w^{|W|}\}, \tag{4.2}$$

of cardinality $|W| > 0$. The transition probabilities of the Markov chain are given by

$$P_W(w^j|w^i) = P_W(w_{t+1} = w^j|w_t = w^i) \in [0, 1], \tag{4.3}$$

for all $w^i, w^j \in W$ and $t \in \mathbb{Z}_{\geq 0}$. The control input at a time instant t is given by

$$u_t = \pi(x_t, w_t) \in U \subset \mathbb{R}^p, \tag{4.4}$$

where π is a control policy and $\Pi = \{\pi : \mathbb{R}^n \times W \to U\}$ is the set of admissible control policies.

Let $G \subset \mathbb{R}^n$ be a set representing the prescribed state constraints. The random variable τ (also referred to as the first exit-time from G) for a given control policy $\pi \in \Pi$ and initial $x_0 \in G$ and $w_0 \in W$ is as follows

$$\tau(x_0, w_0, \pi) = \inf\{t \in \mathbb{Z}_{\geq 0} : x_t \notin G\}, \tag{4.5}$$

where x_t results from applying control policy π to system (4.1) with initial condition x_0 and w_0. The expected value of the first exit-time from G is given by

$$\bar{\tau}(x, w, \pi) = \mathrm{E}\{\tau(x, w, \pi) \,|\, x, w, \pi\}. \tag{4.6}$$

The stochastic DCOC problem reads

$$J(x, w, \pi) = \mathrm{E}\left\{ \sum_{t=0}^{\tau(x,w,\pi)-1} g(x_t, u_t) \,|\, x, w, \pi \right\} \to \max_{\pi \in \Pi}, \tag{4.7}$$

for $x = x_0 \in G$, $w = w_0 \in W$, and $\pi \in \Pi$, where $g : G \times U \to \mathbb{R}_+$ denotes the instantaneous yield.

Similar to (2.4) in the deterministic case, the value function is defined as follows

$$V(x, w) = \sup_{\pi \in \Pi} J(x, w, \pi). \tag{4.8}$$

4.2 Boundedness of Expected First Exit-Time and Value Function

The following theorems provide conditions under which $\bar{\tau}$ (Theorem 4.1) and V (Theorem 4.2) are bounded. The following assumption is made in this regard.

Assumption 4.1. There exists $T > 0$ and $\bar{w} \in W$ such that $\bar{w}$ overpowers any admissible control and the deterministic system,

$$x_{t+1} = f(x_t, \pi(x_t, \bar{w}), \bar{w}),$$

exits G in at most T steps for all $x_0 \in G$ and $\pi \in \Pi$. In addition, $P_W(\bar{w}|\bar{w}) > 0$ and $\bar{w}$ is accessible from each $w \in W$.

Theorem 4.1. *Suppose Assumption 4.1 holds. Then there exists $\bar{T} > 0$ such that*

$$\bar{\tau}(x, w, \pi) \leq \bar{T},$$

for all $x \in G$, $w \in W$, and $\pi \in \Pi$.

Proof. Let $x \in G$, $w \in W$, and $\pi \in \Pi$ be a given initial condition and admissible control policy, respectively, where the corresponding expected first-exit time may be expressed as

follows:

$$\begin{aligned}\bar{\tau}(x,w,\pi) &= \sum_{i=1}^{\infty} iP(\tau(x,w,\pi)=i) \\ &\leq \sum_{i=1}^{\infty} iP(\tau(x,w,\pi)\geq i),\end{aligned} \tag{4.9}$$

where $P(\tau(x,w,\pi)=i)$ and $P(\tau(x,w,\pi)\geq i)$ denote the probabilities that the first exit-time is equal to i or greater than or equal to i, respectively. Using Assumption 4.1 and

$$\rho_{\bar{w},T,i} = \text{Prob}(\bar{w} \text{ occurs } T \text{ times in a row prior to } t=i-1), \tag{4.10}$$

$P(\tau(x,w,\pi)\geq i)$ is bounded according to

$$P(\tau(x,w,\pi)\geq i) \leq 1-\rho_{\bar{w},T,i}, \tag{4.11}$$

where $\bar{w}$ and T are defined in Assumption 4.1. Using Assumption 4.1 (in particular: $\bar{w}$ is accessible from every $w^i \in W$) and denoting

$$q_T = \text{Prob}(\bar{w} \text{ is reached from } w \text{ in at most } |W| \text{ steps}) \times [P_W(\bar{w}|\bar{w})]^T, \tag{4.12}$$

which is greater than zero due to the accessibility of $\bar{w}$ and $P_W(\bar{w}|\bar{w})>0$ by Assumption 4.1, it follows that

$$\begin{aligned}\rho_{\bar{w},T,i} &\geq \sum_{k=0}^{\lfloor \frac{i-1}{T+|W|} \rfloor - 1} (1-q_T)^k q_T \\ &= q_T \left(\frac{1-(1-q_T)^{\lfloor \frac{i-1}{T+|W|} \rfloor}}{1-(1-q_T)} \right) \\ &= 1-(1-q_T)^{\lfloor \frac{i-1}{T+|W|} \rfloor},\end{aligned} \tag{4.13}$$

where $\lfloor \cdot \rfloor$ is the floor operator. Hence, (4.11) becomes

$$P(\tau(x,w,\pi)\geq i) \leq (1-q_T)^{\lfloor \frac{i-1}{T+|W|} \rfloor}, \tag{4.14}$$

and (4.9) may be written as follows

$$
\begin{aligned}
\bar{\tau}(x,w,\pi) &\leq \sum_{i=1}^{\infty} i(1-q_T)^{\left\lfloor \frac{i-1}{T+|W|} \right\rfloor} \\
&\leq \sum_{k=0}^{\infty} (k+1)(T+|W|)^2(1-q_T)^k \\
&= (T+|W|)^2 \left(\frac{1-q_T}{q_T^2} + \frac{1}{q_T} \right) \\
&= \left(\frac{T+|W|}{q_T} \right)^2 = \bar{T}.
\end{aligned} \tag{4.15}
$$

□

Theorem 4.2. *Suppose Assumptions 2.1 and 4.1 hold. Then there exists $\bar{V} > 0$ such that*

$$
J(x,w,\pi) \leq V(x,w) \leq \bar{V},
$$

for all $x \in G$, $w \in W$, and $\pi \in \Pi$.

Proof. In analogy to the proof of Theorem 4.1 and using Assumption 2.1,

$$
\begin{aligned}
J(x,w,\pi) &\leq \bar{g} \sum_{i=1}^{\infty} iP(\tau(x,w,\pi) = i) \\
&\leq \bar{g} \sum_{i=1}^{\infty} iP(\tau(x,w,\pi) \geq i),
\end{aligned} \tag{4.16}
$$

for a given $\pi \in \Pi$, $x \in G$, and $w \in W$. Following the same steps as in the proof of Theorem 4.1, one obtains that

$$
J(x,w,\pi) \leq \bar{g} \left(\frac{T+|W|}{q_T} \right)^2 = \bar{V}, \tag{4.17}
$$

where q_T is defined by (4.12). Together with (4.8), (4.17) implies that $V(x,w) \leq \bar{V}$. □

Remark 4.1. *Theorem 4.2 guarantees the existence of a maximizing sequence for all $x \in G$ and $w \in W$, i.e., a sequence $\{\pi_n\}$ in Π such that $J(x,w,\pi_n) \to \sup_{\pi \in \Pi} J(x,w,\pi)$ or, equivalently, $J(x,w,\pi_n) \to V(x,w)$ for all $x \in G$ and $w \in W$.*

4.3 Other Theoretical Results

Let the expected value function for a control input $u \in U$ be defined by

$$\bar{V}(x,u,w) = \sum_{w^i \in W} \left[V(f(x,u,w),w^i) P_W(w^i|w) \right]. \tag{4.18}$$

Moreover, define the expression $L^\pi V(x,w)$ as follows

$$L^\pi V(x,w) = V(x,w) - \bar{V}(x,\pi(x,w),w). \tag{4.19}$$

The next theorem provides conditions for a control policy to be optimal. It is similar to Theorem 2.2 in Section 2.2 (deterministic case).

Theorem 4.3. *Suppose Assumptions 2.1 and 4.1 hold. Then $\pi^* \in \Pi$ satisfies*

$$\begin{aligned} &L^{\pi^*} V(x,w) = g(x,\pi^*(x,w)), \ \text{ if } x \in G, \\ &L^\pi V(x,w) \geq g(x,\pi(x,w)), \ \text{ if } x \in G, \ \pi \neq \pi^*, \\ &V(x,w) = 0, \ \text{ if } x \notin G, \end{aligned} \tag{4.20}$$

for all $x \in \mathbb{R}^n$, $w \in W$, and $\pi \in \Pi$ iff π^* maximizes $J(x,w,\pi)$. Moreover, $V(x,w) = J(x,w,\pi^*)$ and

$$\pi^*(x,w) \in \Pi^*(x,w) = \arg\max_{u \in U} \{ g(x,u) + \bar{V}(x,u,w) \}. \tag{4.21}$$

Proof. Following similar steps as in the proof of Theorem 2.2 (the case $V(x,w) = 0$ if $x \notin G$ is trivial), let $x = x_0 \in G$ and $w = w_0 \in W$ be a given initial condition and $\pi \in \Pi$. Assume that $\pi^* \in \Pi$ satisfies (4.20). This implies that

$$\begin{aligned} J(x,w,\pi) &= \mathrm{E} \left\{ \sum_{t=0}^{\tau(x,w,\pi)-1} g(x_t,\pi(x_t,w_t)) \,|\, x,w,\pi \right\} \\ &\leq \mathrm{E} \left\{ \sum_{t=0}^{\tau(x,w,\pi)-1} L^\pi V(x_t,w_t) \,|\, x,w,\pi \right\} = V(x,w), \end{aligned} \tag{4.22}$$

since $V(x_{\tau(x,w,\pi)},\cdot) = 0$. Similarly, for π^*, by (4.7) and (4.20),

$$
\begin{aligned}
J(x,w,\pi^*) &= \mathrm{E}\left\{ \sum_{t=0}^{\tau(x,w,\pi^*)-1} g(x_t,\pi^*(x_t,w_t)) \,|\, x,w,\pi^* \right\} \\
&= \mathrm{E}\left\{ \sum_{t=0}^{\tau(x,w,\pi^*)-1} L^{\pi^*}V(x_t,w_t) \,|\, x,w,\pi^* \right\} = V(x,w).
\end{aligned}
\tag{4.23}
$$

Because V is bounded by Theorem 4.2 and Assumptions 2.1 and 4.1, (4.22) and (4.23) can be compared which implies that $J(x,w,\pi^*) \geq J(x,w,\pi)$ for all $x \in G$, $w \in W$, and $\pi \in \Pi$. It follows from (4.19) and (4.20) that $\pi^*(x,w) \in \Pi^*(x,w)$ according to (4.21).

Now assume that π^* maximizes $J(x,w,\pi)$ for all $x \in G$ and $w \in W$. Thus, $V(x,w) = J(x,w,\pi^*)$ and, similar to (2.10), it follows from (4.18) that

$$
\begin{aligned}
V(x,w) &= g(x,\pi^*(x,w)) + \sum_{w^i \in W} \left[J(f(x,\pi^*(x,w),w),w^i,\pi^*) P_W(w^i|w) \right] \\
&= g(x,\pi^*(x,w)) + \bar{V}(x,\pi^*(x,w),w),
\end{aligned}
\tag{4.24}
$$

which implies that $L^{\pi^*}V(x,w) = g(x,\pi^*(x,w))$. On the other hand, similar to (2.11), for any admissible $\pi \neq \pi^*$, it is

$$
\begin{aligned}
V(x,w) &\geq g(x,\pi(x,w)) + \sum_{w^i \in W} \left[J(f(x,\pi(x,w),w),w^i,\pi^*) P_W(w^i|w) \right] \\
&= g(x,\pi(x,w)) + \bar{V}(x,\pi(x,w),w).
\end{aligned}
\tag{4.25}
$$

Consequently, $L^{\pi}V(x,w) \geq g(x,\pi(x,w))$. □

Note that Remark 2.2 applies in the stochastic case as well. It is clear that a solution to the DCOC problem (4.7) exists if the set $\Pi^*(x,w)$ in (4.21) is nonempty for all $x \in G$ and $w \in W$, which is addressed in the following theorem that is similar to Theorem 2.4 (deterministic case).

Theorem 4.4. *A solution $\pi^* \in \Pi$ to the stochastic DCOC problem* (4.7) *exists for all $x \in G$ and $w \in W$ iff the set $\Pi^*(x,w)$ in* (4.21) *is nonempty for all $x \in G$ and $w \in W$.*

Proof. If $\Pi^*(x,w)$ is nonempty for all $x \in G$ and $w \in W$, then there exists $\pi^*(x,w) = u^* \in U$ such that $g(x,u^*) + \bar{V}(x,u^*,w) \geq g(x,u) + \bar{V}(x,u,w)$, for all $x \in G$ and $u \in U$. Hence, by Theorem 4.3, π^* is a solution to (4.7). Now assume that an optimal control policy π^* exists for all $x \in G$ and $w \in W$, implying that $V(x,w) = J(x,w,\pi^*)$ for all

$x \in G$ and $w \in W$. Consequently, by denoting $u^* = \pi^*(x, w) \in U$,

$$\begin{aligned} J(x, w, \pi^*) &= g(x, u^*) + \sum_{w^i \in W} \left[J(f(x, u^*, w), w^i, \pi^*) P_W(w^i|w) \right] \\ &\geq g(x, u) + \sum_{w^i \in W} \left[J(f(x, u, w), w^i, \pi^*) P_W(w^i|w) \right], \end{aligned} \tag{4.26}$$

for all $x \in G$, $w \in W$, and $u \in U$. Since $V(x, w) = J(x, w, \pi^*)$, it follows from (4.18) and (4.26) that, for all $x \in G$ and $w \in W$, there exists $u^* = u^*(x, w) \in U$ such that $g(x, u^*) + \bar{V}(x, u^*, w) \geq g(x, u) + \bar{V}(x, u, w)$ for all $u \in U$. □

Two separate conditions are provided in Theorem 4.5 that guarantee the existence of a solution to the stochastic DCOC problem (4.7), where condition 2 is based on Lemma 4.1. The proof of Lemma 4.1 follows the same steps as in the deterministic case (Lemma 2.1) and can be found in Appendix E.

Lemma 4.1. *If $V(x, w)$ is USC with respect to $x \in G$ for all $w \in W$ and $f(x, u, w)$ is continuous with respect to $u \in U$ for all $x \in G$ and $w \in W$, then $\bar{V}(x, u, w)$ is USC with respect to $u \in U$ for all $x \in G$ and $w \in W$.*

Proof. See Appendix E. □

Theorem 4.5. *Suppose either*

1. *U is finite and Assumptions 2.1 and 4.1 hold.*
2. *U is compact, $f(x, u, w)$ and $g(x, u)$ are continuous and USC with respect to $u \in U$, respectively, for all $x \in G$ and $w \in W$, and $V(x, w)$ is USC with respect to $x \in G$ for all $w \in W$.*

Then a solution to (4.7) *exists for all $x \in G$ and $w \in W$.*

Proof. It needs to be shown that $\Pi^*(x, w)$ in (4.21) is nonempty for all $x \in G$ and $w \in W$ since, by Theorem 4.4, this implies the existence of a solution. Assume that 1 holds. By Assumptions 2.1 and 4.1 and Theorem 4.2, both V (and thus $\bar{V}$) and g are bounded for all $x \in G$, $w \in W$, and $u \in U$, respectively. Consequently, their sum is bounded. Since U is finite and the maximum of a bounded function over a finite set exists, $\Pi^*(x, w)$ is nonempty for all $x \in G$ and $w \in W$. Now suppose 2 holds. By Lemma 4.1, $\bar{V}(x, u, w)$ is USC with respect to $u \in U$ for all $x \in G$ and $w \in W$. Since the sum of two USC functions is USC, the sum of g and $\bar{V}$ in (4.21) is USC with respect to $u \in U$ for all $x \in G$ and $w \in W$. Because U is compact, it follows from the extension of the Weierstrass theorem to USC functions [55] that $\Pi^*(x, w)$ is nonempty for all $x \in G$ and $w \in W$. □

The following Theorem 4.6 provides conditions under which the objective function J is USC. In order to prove upper semi-continuity of $J(x, w, \pi)$, the results from the deterministic case in Theorem 2.7 are used. In what follows, a deterministic w_t sequence (that is feasible according to the Markov chain for w_t) is denoted by $\{w_t\}$ and the set of disturbance sequences that lead to x_t deterministically exiting G in $i \in \mathbb{Z}_+$ steps, given $x = x_0 \in G$ and $\pi \in \Pi$, is defined as follows

$$\mathcal{W}_i(x, \pi) = \{\{w'_t\} : x_{t+1} = f(x_t, \pi(x_t, w'_t), w'_t) \text{ exits } G \text{ in } i \in \mathbb{Z}_+ \text{ steps}\}. \tag{4.27}$$

In addition, the deterministic total yield for any $\{w'_t\} \in \mathcal{W}_i(x, \pi)$, $i \in \mathbb{Z}_+$, is defined in analogy to (2.3) as follows

$$J_{\{w'_t\}}(x, \pi) = \sum_{t=0}^{i-1} g(x_t, u_t), \tag{4.28}$$

where x_t evolves according to $x_{t+1} = f(x_t, \pi(x_t, w'_t), w'_t)$.

The probability that, beginning at $w = w_0 \in W$, the sequence $\{w_t\}$ occurs, is denoted by $P(\{w_t\}) \in [0, 1]$. Hence, in analogy to (4.9) and (4.16), $J(x, w, \pi)$ may be expressed as

$$J(x, w, \pi) = \lim_{k\to\infty} \sum_{i=1}^{k} \sum_{\{w_t\}\in\mathcal{W}_i(x,\pi)} J_{\{w_t\}}(x, \pi) P(\{w_t\}), \tag{4.29}$$

for all $x \in G$, $w = w_0 \in W$, and $\pi \in \Pi$.

As in Chapter 2, for analyzing USC properties of the objective function, the set of admissible control policies is reduced to control policies that are continuous on G for all $w \in W$, i.e.,

$$\pi \in C_{G,W}(\Pi) = \{\pi \in \Pi \,|\, \pi \text{ is continuous on } G \text{ for all } w \in W\}. \tag{4.30}$$

Hence, by Theorems 2.1 and 2.7 and due to $\pi \in C_{G,W}(\Pi)$, the sequence in (4.29) is a bounded sequence of USC functions. In order to conclude that $J(x, w, \pi)$ is USC, it needs to be shown that the sequence in (4.29) converges uniformly, see Proposition B.1 in [93]. This is done in Lemma 4.2.

Lemma 4.2. *Suppose Assumptions 2.1 and 4.1 hold and $\pi \in C_{G,W}(\Pi)$. Then the sequence in* (4.29) *converges uniformly for each $x \in G$, $w \in W$, and $\pi \in \Pi$.*

Proof. Using the Weierstrass M-test [94], it needs to be shown that there exists a converging sequence $\{M_i\}$, i.e., $\lim_{k\to\infty} \sum_{i=1}^{k} M_i < \infty$, such that (absolute values are not required

since the expression is non-negative)

$$\sum_{\{w_t\}\in\mathcal{W}_i(x,\pi)} J_{\{w_t\}}(x,\pi)P(\{w_t\}) \leq M_i, \tag{4.31}$$

for each $x \in G$, $w \in W$, $\pi \in \Pi$, and $i \in \mathbb{Z}_+$. Note that

$$\sum_{\{w_t\}\in\mathcal{W}_i(x,\pi)} P(\{w_t\}) = P(\tau(x,w,\pi) = i),$$

which is the probability that the first exit-time is equal to i. Hence,

$$\begin{aligned}\sum_{\{w_t\}\in\mathcal{W}_i(x,\pi)} J_{\{w_t\}}(x,\pi)P(\{w_t\}) &\leq \sum_{\{w_t\}\in\mathcal{W}_i(x,\pi)} J_{\{w_t\}}(x,\pi)P(\tau(x,w,\pi)=i)\\ &\leq \sum_{\{w_t\}\in\mathcal{W}_i(x,\pi)} i\bar{g}P(\tau(x,w,\pi)=i),\end{aligned} \tag{4.32}$$

where the last step is due to Assumption 2.1. According to Assumption 4.1, (4.9), and (4.11), $P(\tau(x,w,\pi)=i) \leq (1-q_T)^{\left\lfloor \frac{i-1}{T+|W|}\right\rfloor}$, where $q_T \in (0,1]$ is defined by (4.12) and T is as in Assumption 4.1. Thus, based on (4.32), $M_i = i\bar{g}(1-q_T)^{\left\lfloor \frac{i-1}{T+|W|}\right\rfloor} \geq 0$ satisfies (4.31). Using (4.15),

$$\lim_{k\to\infty}\sum_{i=1}^{k} M_i \leq \bar{g}\left(\frac{T+|W|}{q_T}\right)^2 < \infty. \tag{4.33}$$

Hence, by the Weierstrass M-test, the sequence in (4.29) converges uniformly for each $x \in G$, $w \in W$, and $\pi \in \Pi$. □

Theorem 4.6. *Suppose Assumptions 2.1 and 4.1 hold. Furthermore, suppose that G is compact, $f(x,u,w)$ is continuous on $G\times U$ for all $w \in W$, and $g(x,u)$ is USC on $G\times U$. Then $J(x,w,\pi)$ is USC with respect to $x \in G$ for all $w \in W$ and $\pi \in C_{G,W}(\Pi)$.*

Proof. Upper semi-continuity of $J(x,w,\pi)$ with respect to $x \in G$ for all $w \in W$ and $\pi \in C_{G,W}(\Pi)$ follows from (4.29), Lemma 4.2, and Proposition B.1 in [93]. Since the sequence of bounded functions that are USC with respect to $x \in G$ for a given admissible π (by Theorems 2.1 and 2.7) in (4.29) is uniformly converging to $J(x,w,\pi)$ for each $x \in G$ and $w \in W$ (by Lemma 4.2), it follows from Proposition B.1 in [93] that $J(x,w,\pi)$ is USC with respect to $x \in G$ for all $w \in W$ and $\pi \in C_{G,W}(\Pi)$. □

Note that upper semi-continuity of the value function cannot be concluded from Theorem 4.6 since the supremum of infinitely many USC functions may not be USC.

4.4 Proportional Feedback VI

4.4.1 Theoretical Results

In this section, proportional feedback VI is extended to the case of stochastic systems. For each $x \in G$ and $w \in W$, the error $e_n(x,w)$ at iteration n is defined by

$$e_n(x,w) = \max_{u \in U} \left\{ g(x,u) + \sum_{w^i \in W} \left[V_n(f(x,u,w),w^i) P(w^i|w) \right] \right\} - V_n(x,w). \quad (4.34)$$

Note that V_n is equal to the value function V iff $e_n(x,w) = 0$ for all $(x,w) \in G \times W$, which follows from Theorem 4.3. In analogy to (2.25), proportional feedback VI for stochastic DCOC is given by

$$\begin{aligned} V_{n+1}(x,w) &= V_n(x,w) + k e_n(x,w), \text{ if } x \in G, \\ V_{n+1}(x,w) &= 0, \text{ if } x \notin G, \end{aligned} \quad (4.35)$$

for all $x \in G$ and $w \in W$, where $k \in \mathbb{R}$ is a proportional gain factor. Note that conventional VI follows for $k = 1$ in (4.35). In order to prove convergence of (4.35) to the value function, Assumption 4.2 is made. Extensive numerical studies (for $m \in \{1, 2, ..., 100, ...\}$) suggest that this assumption holds.

Assumption 4.2. *If $k \in (0,2)$, then*

$$\lim_{n \to \infty} k^m \sum_{j=0}^{n} (1-k)^j \prod_{i=1}^{m-1} (\frac{j}{i} + 1) = 1,$$

for all $m \in \mathbb{Z}_{\geq 1}$.

Theorem 4.7. *Suppose Assumption 4.2 holds, $k \in (0,2)$, and a solution to the stochastic DCOC problem* (4.7) *exists. Furthermore, let $V_0(x,w) \in \mathbb{R}$ be defined for all $x \in G$ and $w \in W$ and let $g(x_t, \pi(x_t, w_t)) = 0$ for all $t \geq \tau(x_0, w_0, \pi)$. Then the sequence of functions given by* (4.34) *and* (4.35) *converges pointwise to $V(x,w)$ for all $x \in G$ and $w \in W$.*

Proof. It is clear from (4.20) and (4.35) that $V_n(x,w) = V(x,w) = 0$ for all n if $x \notin G$. Now, for a given $x \in G$ and $w \in W$, (4.34) and (4.35) can be written as

$$V_{n+1}(x,w) = V_n(x,w) + k[V(x,w) + \beta_n(x,w) - V_n(x,w)]. \quad (4.36)$$

It needs to be shown that, if $k \in (0,2)$, then $\beta_n(x,w) \to 0$ as $n \to \infty$, which implies that V_n converges to V. By setting $x_0 = x$ and $w_0 = w$, it follows from (4.34), (4.35), and

(4.36) that

$$V(x,w)+\beta_n(x,w)=\mathrm{E}\{V_n(x_1,w_1)\,|\,x,w,u^n\}+g(x,u^n), \tag{4.37}$$

where, x_1 and w_1 are the state vector and disturbance at the next time instant and $u^n \in U$ is the maximizer in (4.34) based on V_n. The term $\mathrm{E}\{...\}$ denotes the expectation of $V_n(x_1,w_1)$ conditional on x, w, and u^n. According to (4.34) and (4.35), (4.37) may be written as

$$\begin{aligned} V(x,w)+\beta_n(x,w) &= g(x,u^n)+(1-k)\mathrm{E}\{V_{n-1}(x_1,w_1)\,|\,x,w,u^n\} \\ &+ k\mathrm{E}\{V_{n-1}(x_2,w_2)\,|\,x,x_1,w,w_1,u^n,u^{n-1}\}+k\mathrm{E}\{g(x_1,u^{n-1})\,|\,x,w,u^n\}, \end{aligned} \tag{4.38}$$

where x_2 and w_2 are the state vector and disturbance two time instants ahead. By continuing to apply (4.34) and (4.35), eventually (4.38) is expressed in terms of V_0, reading

$$V(x,w)+\beta_n(x,w)=g(x,u^n)+y_g(n)+y_{V_0}(n), \tag{4.39}$$

where $y_g(n)$ is a sum of expected values of g at the future states $x_1,x_2,...,x_n$ using the controls $u^0,...,u^{n-1}$. Similarly, $y_{V_0}(n)$ is a sum of expected values of V_0 at the future states $x_1,...,x_{n+1}$ and disturbances $w_1,...,w_{n+1}$, which can be bounded by

$$|y_{V_0}(n)| \le \left|\sum_{j=0}^{n}\left[cnk^j(1-k)^{n-j}\mathrm{E}V_0(x_{j+1},w_{j+1})\right]\right|, \tag{4.40}$$

where $c>0$ is a constant and the expectation is conditional on $x,x_1,...,x_n$, $w,w_1,...,w_n$, and $u^0,...,u^n$. It follows from $k\in(0,2)$ that the terms with small j in (4.40) approach zero. The remaining terms in (4.40) also vanish as $n\to\infty$ due to the existence of a solution to the DCOC problem (by assumption), as this implies $\mathrm{Prob}(x_n\notin G)\to 1$ as $n\to\infty$. Hence, $V_0(x_n,w_n)=0$ with probability one as $n\to\infty$ and $y_{V_0}(n)\to 0$. Consequently,

$$V(x,w)+\lim_{n\to\infty}\beta_n(x,w)=\lim_{n\to\infty}(y_g(n)+g(x,u^n)). \tag{4.41}$$

The terms $\mathrm{E}g(x_m,u^{n-m}),...,\mathrm{E}g(x_m,u^0)$, where $m\in\{1,2,...,n\}$, (indicating conditional dependence is omitted for brevity henceforth) are contained in $y_g(n)$. Based on (4.34) and (4.35), one can write

$$y_g(n)=\sum_{m=1}^{n}k^m\sum_{j=0}^{n-m}N_m(j)(1-k)^j\mathrm{E}g(x_m,u^{(n-m-j)}), \tag{4.42}$$

where $N_m(j)\in\mathbb{Z}_{\ge 0}$ denotes the number of times the term $(1-k)^j\mathrm{E}g(x_m,\cdot)$ appears in

$y_g(n)$ for a given n, $m \in \{1, 2, ..., n\}$, and $j \in \{0, 1, ..., n-m\}$. It follows from (4.34) and (4.35) that

$$N_m(j) = 1 \text{ if } j = 0, \tag{4.43a}$$

$$N_1(j) = 1 \text{ for } j \in \mathbb{Z}_{\geq 0}, \tag{4.43b}$$

$$N_m(j) = N_m(j-1) + N_{m-1}(j) \text{ for } j \in \mathbb{Z}_+ \text{ and } m \in \{2, 3, ...\}, \tag{4.43c}$$

which can be summarized by

$$N_m(j) = \prod_{i=1}^{m-1} \left(\frac{j}{i} + 1 \right). \tag{4.44}$$

As $n \to \infty$, u^0, u^1, etc. in (4.42) and (4.41) approach optimal values due to the assumption that a solution to the DCOC problem exists. Hence, by denoting an optimal control policy by π^*, it follows from (4.42) and (4.44) that

$$\lim_{n\to\infty} y_g(n) = \sum_{m=1}^{\infty} \mathrm{E} g(x_m, \pi^*(x_m, w_m)) k^m \sum_{j=0}^{\infty} (1-k)^j \prod_{i=1}^{m-1} \left(\frac{j}{i} + 1 \right). \tag{4.45}$$

Using Assumption 4.2 and $k \in (0, 2)$, one obtains from (4.45) that

$$\begin{aligned} \lim_{n\to\infty} y_g(n) &= \sum_{m=1}^{\infty} \mathrm{E} g(x_m, \pi^*(x_m, w_m)) \\ &= \mathrm{E} \left\{ \sum_{m=1}^{\infty} g(x_m, \pi^*(x_m, w_m)) \right\} \\ &= \mathrm{E} \left\{ \sum_{m=1}^{\tau(x,w,\pi^*)-1} g(x_m, \pi^*(x_m, w_m)) \right\}, \end{aligned} \tag{4.46}$$

since $g(x_t, \pi^*(x_t, w_t)) = 0$ for all $t \geq \tau(x, w, \pi^*)$ by assumption. Consequently, by (4.7) and (4.46), (4.41) becomes

$$V(x, w) + \lim_{n\to\infty} \beta_n(x, w) = J(x, w, \pi^*) = V(x, w), \tag{4.47}$$

implying $\beta_n(x, w) \to 0$ and thus $V_n(x, w) \to V(x, w)$ as $n \to \infty$. □

Note that Remark 2.3 about convergence of control policies also applies in the stochastic case.

4.4.2 Adaptive Proportional Feedback VI with Damping

In theory, as in the deterministic case, the optimal gain for fast convergence of iterations (4.35) is $k = 1$. However, in practice, iterations (4.35) are applied to a discrete subset $\tilde{G}$ of G, where a function approximator (such as NNs or linear interpolation) is used to evaluate $V(x, w)$ if $x \notin \tilde{G}$. In this regard, the same considerations as in Section 2.3.2 apply and $k = 1$ may not be the optimal gain. Instead, the optimal gain may depend on x and w. Moreover, it may depend on the selection of $\tilde{G}$ as well as of the function approximator and convergence may occur for $k \in (0, k_{\max})$, where $k_{\max}$ may be greater than 2 or, on the contrary, $k_{\max} < 2$. Therefore, individual adaptive gains $k_n : \tilde{G} \times W \to \mathbb{R}_+$ are introduced in this section. This approach is referred to as adaptive proportional feedback VI, which is given by

$$\begin{aligned} V_{n+1}(x, w) &= V_n(x, w) + k_n(x, w)e_n(x, w), \text{ if } x \in G \\ V_{n+1}(x, w) &= 0, \text{ if } x \notin G \\ k_{n+1}(x, w) &= k_n(x, w) + \delta e_n(x, w). \end{aligned} \tag{4.48}$$

where $\delta \geq 0$ is the learning rate.

In addition, the algorithm is extended by including damping. This is because numerical experiments with adaptive proportional feedback VI show that the error $e_n(x, w)$ oscillates between negative and positive values when diverging, i.e., when $k_n(x, w) > k_{\max}(x, w)$. Since $k_{\max}(x, w)$ may be unknown for a given $(x, w) \in \tilde{G} \times W$, this can be detected by comparing $e_n(x, w)$ and $e_{n+1}(x, w)$. If oscillations occur, the gain $k_{n+1}(x, w)$ is lowered using a damping factor $\zeta \in (0, 1]$. Adaptive proportional feedback VI with damping is as follows, where $\delta \geq 0$ is the learning rate as in (4.48),

$$\begin{aligned} V_{n+1}(x, w) &= V_n(x, w) + k_n(x, w)e_n(x, w), \text{ if } x \in G \\ V_{n+1}(x, w) &= 0, \text{ if } x \notin G \\ k_{n+1}(x, w) &= k_n(x, w) + \delta e_n(x, w), \text{ if } e_n(x, w)e_{n+1}(x, w) \geq 0 \\ k_{n+1}(x, w) &= \zeta k_n(x, w), \text{ if } e_n(x, w)e_{n+1}(x, w) < 0. \end{aligned} \tag{4.49}$$

In the theoretical case (where $\tilde{G} = G$), it is straightforward to show that iterations (4.48) and (4.49) converge to V if, in addition to the assumptions in Theorem 4.7, each individual gain is bounded by $k_n(x, w) \in (0, 2)$ for all $n \in \mathbb{Z}_{\geq 0}$. However, this may not hold in practice (where $\tilde{G} \subset G$) as explained above and it is expected that, for proper choices of δ and ζ, iterations (4.49) effectively adjust each individual gain to improve the convergence rate.

4.5 Application: Driving Policies for Autonomous Vehicles

This section focuses on using stochastic DCOC for high-level control and decision-making of autonomous cars. The presented approach for autonomous driving is based on a hierarchical control structure, where the DCOC-based controller provides optimal decision-making (high-level) and low-level controllers execute the decisions by regulating the longitudinal and lateral motion of the car. While the focus here is on high-level control, relevant low-level controllers are discussed in [95–97].

One of the earliest relevant studies on modeling decision-making in driving can be found in [98], where Markov chains calibrated from real traffic data are used. A set of deterministic rules for lane-changing decisions for cars traveling at a constant velocity is proposed in [99]. More complex probabilistic approaches can be found in [100–102]. A game theoretic approach for lateral and longitudinal decision-making is considered in [103], where the control problem is decomposed into car following and lane-changing subproblems and a cost function is minimized.

In contrast to previous work, the application of stochastic DCOC provides a systematic approach to generating driving policies that may enhance safety for autonomous cars by maximizing the expected time that a prescribed minimum (safe) headway to the in-front vehicle is maintained. In order to formulate the stochastic DCOC problem, a hybrid probabilistic model that describes the motion of a car and its surrounding traffic is developed in Section 4.5.1. Section 4.5.2 extends the DCOC framework to consider such hybrid systems. An ADP approach based on proportional feedback VI is proposed in Section 4.5.3 and a numerical case study is considered in Section 4.5.4.

4.5.1 Driving Model

A discrete-time stochastic hybrid model is formulated to describe the motion of a car and its surrounding traffic. While the developments in this dissertation are limited to roads with two lanes, which constitute the largest fraction of the multi-lane roads, the modeling framework may readily be extended to more than two lanes as explained in Remark 4.2.

The proposed model considers three cars. Subscript "m" denotes the controlled/ego car ("my car") and subscripts "c" and "o" denote the closest cars ahead of the controlled car in its current lane and in the other lane, respectively. The state vector at a time instant $t \in \mathbb{Z}_{\geq 0}$ is given by

$$x_t = \left[s_{\mathrm{c},t}, s_{\mathrm{o},t}, v_{\mathrm{m},t}\right]^\top,$$

where s_{c} and s_{o} are the respective headways relative to the closest cars ahead in each lane, and v_{m} is the velocity of the controlled car. In addition,

$$w_t = [v_{\mathrm{c},t}, v_{\mathrm{o},t}]^\top,$$

is a random disturbance, where v_{c} and v_{o} are the respective velocities of the two cars ahead. Note that the closest car ahead of the controlled car in the other lane is defined to be the closest car with a relative distance $s \geq -(d+\gamma)$, where d is the length of the controlled car and γ provides a margin of safety (here $\gamma = 0$ is used).

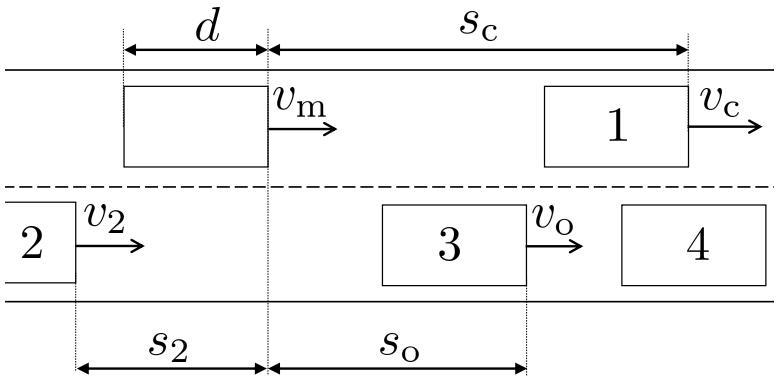

Figure 4.1: Driving model: traffic example.

Figure 4.1 shows a traffic situation at which the controlled/ego car is driving in the upper lane (current lane) and car 1 is the closest car ahead in its current lane. The closest car ahead in the other lane is car 3. If the velocity difference between car 2 and the controlled car, $v_2 - v_{\mathrm{m}}$, is positive, the distance between the two cars will eventually become $s_2 \geq -d$ and car 2 becomes the closest car ahead in the other lane, i.e., $s_{\mathrm{o}} = s_2$. Moreover, if car 3 cuts in between the controlled car and car 1, car 3 becomes the closest car ahead in the current lane and either car 2 (if $s_2 \geq -d$) or car 4 becomes the closest car ahead in the other lane. In addition to these scenarios, there are several other possible transitions, which are addressed by introducing a discrete-valued variable θ.

The variable θ models possible scenarios/transitions from time instant t to $t+1$. For the two-lane case, seven different transitions can be identified, $\theta \in \{1, 2, 3, 4, 5, 6, 7\}$. In the following, c_t and o_t denote the closest cars ahead in the current and in the other lane, respectively, at the time instant t. Similarly c_{t+1} and o_{t+1} denote the closest cars ahead in the respective lanes at the time instant $t+1$. The possible transitions are given by

- $\theta = 1$: c_t remains the closest car ahead in the current lane ($\mathrm{c}_t \to \mathrm{c}_{t+1}$) and o_t remains the closest car ahead in the other lane ($\mathrm{o}_t \to \mathrm{o}_{t+1}$).
- $\theta = 2$: c_t remains the closest car ahead in the current lane ($\mathrm{c}_t \to \mathrm{c}_{t+1}$) and a car other than o_t becomes the closest car ahead in the other lane (new car $\to \mathrm{o}_{t+1}$).
- $\theta = 3$: a car other than c_t or o_t becomes the closest car ahead in the current lane (new car $\to \mathrm{c}_{t+1}$) and a car other than c_t or o_t becomes the closest car ahead in the other lane (new car $\to \mathrm{o}_{t+1}$).
- $\theta = 4$: a car other than c_t becomes the closest car ahead in the current lane (new car $\to \mathrm{c}_{t+1}$) and o_t remains the closest car ahead in the other lane ($\mathrm{o}_t \to \mathrm{o}_{t+1}$).
- $\theta = 5$: o_t becomes the closest car ahead in the current lane ($\mathrm{o}_t \to \mathrm{c}_{t+1}$) and a car other than c_t becomes the closest car ahead in the other lane (new car $\to \mathrm{o}_{t+1}$).
- $\theta = 6$: o_t becomes the closest car ahead in the current lane ($\mathrm{o}_t \to \mathrm{c}_{t+1}$) and c_t becomes the closest car ahead in the other lane ($\mathrm{c}_t \to \mathrm{o}_{t+1}$).
- $\theta = 7$: a car other than o_t becomes the closest car ahead in the current lane (new car $\to \mathrm{c}_{t+1}$) and c_t becomes the closest car ahead in the other lane ($\mathrm{c}_t \to \mathrm{o}_{t+1}$).

The control input vector at a time instant t is given by

$$u_t = [a_{\mathrm{m},t}, l_{\mathrm{m},t}]^\top \in U = A \times \{0, 1\},$$

where $a_\mathrm{m} \in \mathcal{A}$ denotes the acceleration of the controlled car and $l_\mathrm{m} \in \{0, 1\}$ indicates whether to initiate a lane change ($l_\mathrm{m} = 1$) or not ($l_\mathrm{m} = 0$). In the case where $l_{\mathrm{m},t} = 1$, the current lane at the time instant t becomes the other lane at $t + 1$, whereas the other lane at t becomes the current lane at $t + 1$. Furthermore, the relative time gaps are defined as follows

$$T_{\mathrm{g,c}} = s_\mathrm{c}/v_\mathrm{m},\ T_{\mathrm{g,o}} = s_\mathrm{o}/v_\mathrm{m}. \tag{4.50}$$

The driving model is given by

$$x_{t+1} = f(x_t, u_t, w_t, \theta_t) = \begin{bmatrix} s_{\mathrm{c},t+1} \\ s_{\mathrm{o},t+1} \\ v_{\mathrm{m},t} + \Delta t a_{\mathrm{m},t} \end{bmatrix}, \tag{4.51}$$

where Δt is the sampling time. Introducing the relative velocities $\Delta v_\mathrm{c} = v_\mathrm{c} - v_\mathrm{m}$ and $\Delta v_\mathrm{o} = v_\mathrm{o} - v_\mathrm{m}$, $s_{\mathrm{c},t+1}$ and $s_{\mathrm{o},t+1}$ in (4.51) are given by

$$s_{\mathrm{c},t+1} = \begin{cases} \min\{s_{\max}, s_{\mathrm{c},t} + \Delta t \Delta v_{\mathrm{c},t}\}, \text{ if } \theta_t \in \{1,2\}, \\ \mathrm{init}_{\mathrm{c}}(T_{\mathrm{g,c},t}, T_{\mathrm{g,o},t}, \theta_t), \text{ if } \theta_t \in \{3,4,7\}, \\ \min\{s_{\max}, s_{\mathrm{o},t} + \Delta t \Delta v_{\mathrm{o},t}\}, \text{ if } \theta_t \in \{5,6\}, \end{cases} \tag{4.52}$$

$$s_{\mathrm{o},t+1} = \begin{cases} \min\{s_{\max}, s_{\mathrm{o},t} + \Delta t \Delta v_{\mathrm{o},t}\}, \text{ if } \theta_t \in \{1,4\}, \\ \mathrm{init}_{\mathrm{o}}(T_{\mathrm{g,c},t}, T_{\mathrm{g,o},t}, \theta_t), \text{ if } \theta_t \in \{2,3,5\}, \\ \min\{s_{\max}, s_{\mathrm{c},t} + \Delta t \Delta v_{\mathrm{c},t}\}, \text{ if } \theta_t \in \{6,7\}, \end{cases} \tag{4.53}$$

where $s_{\max}$ is the maximum headway at which a car can be detected. If no car is ahead of the controlled vehicle in the respective lane, $s_{\mathrm{c}} = s_{\max}$ or $s_{\mathrm{o}} = s_{\max}$, respectively. The functions $\mathrm{init}_{\mathrm{c}}$ and $\mathrm{init}_{\mathrm{o}}$ in (4.52) and (4.53) set the value for s_{c} in case $\theta_t \in \{3,4,7\}$ and for s_{o} in case $\theta_t \in \{2,3,5\}$, respectively, depending on the current relative time gaps. Both $\mathrm{init}_{\mathrm{c}}$ and $\mathrm{init}_{\mathrm{o}}$ are defined below by (4.56) and (4.57), respectively.

As in [37], the velocities v_{c} and v_{o} are random variables that are modeled as Markov chains and take values in the discrete set $\mathcal{V} = \{v^j : j \in I_v\}$. The probability of transitioning from $w_t = [v^i, v^j]^\top$ to $w_{t+1} = [v^q, v^r]^\top$, given $\theta_t = p \in \{1,2,...,7\}$, is given by

$$P_W(v^q, v^r | v^i, v^j, p) \in [0,1], \tag{4.54}$$

for all $i, j, q, r \in I_v$. Similarly, θ is a random variable and the probability that $\theta_t = p \in \{1,2,...,7\}$, given $T_{\mathrm{g,c},t} = T^i \in \mathcal{T}$, $T_{\mathrm{g,o},t} = T^j \in \mathcal{T}$, and $l_{\mathrm{m},t} = q \in \{0,1\}$, is denoted by $P_\theta(p|T^i, T^j, q) \in [0,1]$ for all $i, j \in I_T$, where $\mathcal{T} = \{T^j : j \in I_T\}$ is a discrete set.

Unlike v_{c} and v_{o}, $T_{\mathrm{g,c}}$ and $T_{\mathrm{g,o}}$ are continuous variables and nearest-neighbor interpolation is used to map $T_{\mathrm{g,c}} \notin \mathcal{T}$ and $T_{\mathrm{g,o}} \notin \mathcal{T}$ onto $\mathcal{T}$ when computing P_θ. The nearest-neighbor operator that maps a point $r \in \mathcal{R}$, where $\mathcal{R}$ is a continuous set, onto the discrete set $\tilde{\mathcal{R}} = \{r^j : j \in I_r\} \subset \mathcal{R}$, is defined as

$$\mathrm{nn}_{\tilde{\mathcal{R}}}(r) \in \{r^j \in \tilde{\mathcal{R}} : \left\|r^j - r\right\|_2 \leq \left\|r^i - r\right\|_2 \text{ for all } r^i \in \tilde{\mathcal{R}}\}. \tag{4.55}$$

Consequently, the probability that $\theta_t = p \in \{1,2,...,7\}$, given $T_{\mathrm{g,c},t}$, $T_{\mathrm{g,o},t}$, and $l_{\mathrm{m},t} = q \in \{0,1\}$, is

$$P_\theta(p | \mathrm{nn}_{\mathcal{T}}(T_{\mathrm{g,c},t}), \mathrm{nn}_{\mathcal{T}}(T_{\mathrm{g,o},t}), q).$$

Similar to [37], the probabilities P_W and P_θ may be calibrated based on observations made in real or simulated traffic. Likewise, traffic observations may be used to calibrate the functions $\mathrm{init}_{\mathrm{c}}$ and $\mathrm{init}_{\mathrm{o}}$ in (4.52) and (4.53), respectively. For a given $T^i, T^j \in \mathcal{T}$ and $\theta = p \in \{1,2,...,7\}$, the respective output of $\mathrm{init}_{\mathrm{c}}$ and $\mathrm{init}_{\mathrm{o}}$ is the average value from

observations, i.e.,

$$\text{init}_{\text{c}}(T^i, T^j, p) = \bar{s}_{\text{c}}(T^i, T^j, p), \tag{4.56}$$

$$\text{init}_{\text{o}}(T^i, T^j, p) = \bar{s}_{\text{o}}(T^i, T^j, p), \tag{4.57}$$

where $\bar{s}_{\text{c}}(T^i, T^j, p)$ and $\bar{s}_{\text{o}}(T^i, T^j, p)$ are the average values from observations for the given T^i, T^j, and p. As for P_θ, nearest-neighbor interpolation according to (4.55) is used to map $T_{\text{g,c}} \notin \mathcal{T}$ and $T_{\text{g,o}} \notin \mathcal{T}$ onto the discrete set $\mathcal{T}$.

Remark 4.2. *The driving model may readily be extended to multiple lanes, which will be investigated in future work. Moreover, additional cars may be considered (e.g., closest cars behind the ego car). This increases the complexity of the model due to the additional states, control inputs ($l_{\text{m}} \in \{-1, 0, 1\}$, where $l_{\text{m}} = 0$ to stay in the current lane, $l_{\text{m}} = -1$ to switch to the left lane, and $l_{\text{m}} = 1$ to switch to the right lane), and, especially, the increase in possible transitions/scenarios θ. The increased complexity may be addressed by only considering the most likely scenarios θ for a given condition (e.g., only consider the 5 most likely scenarios to compute/estimate the states at the next time instant). This provides a balance between model complexity and accuracy.*

4.5.2 Extension of DCOC Framework to Hybrid Systems

The stochastic DCOC problem in (4.7) is extended to stochastic hybrid systems, where the focus in this section is on time maximization problems (i.e., the objective is to maximize the expected value of the first exit-time).

4.5.2.1 Problem Formulation

The DCOC (time maximization) problem for stochastic hybrid systems is given by

$$\begin{aligned} &\max_{\pi \in \Pi} \bar{\tau}(x_0, w_0, \pi) \\ &\text{subject to } x_{t+1} = f(x_t, u_t, w_t, \theta_t), \end{aligned} \tag{4.58}$$

where $\bar{\tau}(x_0, w_0, \pi)$ is the expected first exit-time defined in (4.6). As before, $x \in \mathbb{R}^n$ and $u \in U$ denote the state and control input vector, respectively, where $u_t = \pi(x_t, w_t)$ and $\pi \in \Pi$. Furthermore, $w \in W = \{w^j : j \in I_w\} \subset \mathbb{R}^{n_w}$ and $\theta \in I_\theta \subset \mathbb{Z}_+$ are random variables. The evolution of w is modeled as a Markov chain where the probability of transitioning from w^i to w^j, given $\theta = p \in I_\theta$, is $P_W(w^j | w^i, p)$ for all $i, j \in I_w$. The

probability of $\theta = p \in I_\theta$, given $x \in G$, $u \in U$, and $w \in W$, is given by

$$P_\theta(p|\mathrm{nn}_{\tilde{G}}(x), \mathrm{nn}_{\tilde{U}}(u), w), \tag{4.59}$$

where $\tilde{G} \subset G$ and $\tilde{U} \subset U$ are prescribed discrete sets and nn is the nearest-neighbor operator defined in (4.55).

Remark 4.3. *For DCOC problems with stochastic hybrid systems as in* (4.58)*, in order to guarantee boundedness of the expected first exit-time and value function in analogy to Theorems 4.1 and 4.2, respectively, Assumption 4.1 is extended as follows. In addition to the disturbance $\bar{w} \in W$ that overpowers any admissible control, for all $x_0 \in G$ and $\pi \in \Pi$, there exists a scenario trajectory $\{\bar{\theta}_t\}$ that occurs with nonzero probability such that the deterministic system,*

$$x_{t+1} = f(x_t, \pi(x_t, \bar{w}), \bar{w}, \bar{\theta}_t),$$

exits G in at most T steps. Moreover, $P_W(\bar{w}|\bar{w}, \bar{\theta}_t) > 0$ and $\bar{w}$ is accessible from each $w \in W$ for all $\bar{\theta}_t \in \{\bar{\theta}_t\}$.

4.5.2.2 Optimal Control and Proportional Feedback VI

The value function is as in (4.8). In analogy to (4.18), using (4.58) and (4.59), the expected value function for a control input $u \in U$ is defined as follows

$$\bar{V}(x, u, w) = \sum_{p \in I_\theta} \left[\sum_{j \in I_w} V(f(x, u, w, p), w^j) P_W(w^j|w, p) \right] P_\theta(p|\mathrm{nn}_{\tilde{G}}(x), \mathrm{nn}_{\tilde{U}}(u), w). \tag{4.60}$$

According to (4.21), an optimal control policy π^* (assuming one exists) maximizes the expected value of the value function at the next time instant. Hence, with $g \equiv 1$ in this case,

$$\pi^*(x, w) \in \max_{u \in U} \bar{V}(x, u, w). \tag{4.61}$$

The value function V is computed with proportional feedback VI (4.35). Similar to (4.34), the error $e_n(x, w)$ at iteration n is given by

$$e_n(x, w) = \max_{u \in U} \bar{V}_n(x, u, w) + 1 - V_n(x, w), \tag{4.62}$$

where

$$\bar{V}_n(x,u,w) = \sum_{p\in I_\theta}\left[\sum_{j\in I_w} V_n(f(x,u,w,p),w^j)P_W(w^j|w,p)\right] P_\theta(p|\mathrm{nn}_{\tilde{G}}(x),\mathrm{nn}_{\tilde{U}}(u),w). \tag{4.63}$$

Note that the convergence properties in Theorem 4.7 also hold for DCOC problems of the form (4.58) and the practical considerations stated in Section 2.3.2 apply here as well. Hence, in practice, V is approximated on a mesh of discrete points (grid) $\tilde{G}\times W$ chosen in the set $G\times W$ and iterations (4.35) are applied with V_n approximated between the grid points through interpolation. In Section 4.5.4, this approach is referred to as conventional DP. Conventional DP is limited to lower-dimensional problems due to the curse of dimensionality (the grid size and computational effort grow exponentially with the dimension). Hence, a procedure based on ADP that mitigates the curse of dimensionality is proposed in the next section (Section 4.5.3).

4.5.3 ADP Approach

A feedforward NN is used to approximate the value function, i.e., $\tilde{V}(x,w)\approx V(x,w)$ for all $x\in G$ and $w\in W$, and $\tilde{V}(x,w)=0$ if $x\notin G$. The NN consists of one input layer with $n+n_w$ inputs, potentially several hidden layers, and an output layer with one linear output neuron (1L). The neurons of the hidden layers are activated by logistic functions, see [104]. Standard notation is used to denote an NN with, for example, two hidden layers by 12-6-1L, where the first hidden layer contains 12 neurons and the second hidden layer contains 6 neurons.

The NN is trained by the back-propagation algorithm with the momentum term, see [104]. The set of training points is denoted by

$$X_{\mathrm{train}} = \{X^1_{\mathrm{train}}, X^2_{\mathrm{train}}, ..., X^{n_{\mathrm{train}}}_{\mathrm{train}}\},$$

with $X^j_{\mathrm{train}}\in G\times W$ for each $j\in\{1,2,...,n_{\mathrm{train}}\}$. The output target values that correspond to X_{train} are contained in

$$V_{\mathrm{train}} = \{V^1_{\mathrm{train}}, V^2_{\mathrm{train}}, ..., V^{n_{\mathrm{train}}}_{\mathrm{train}}\}.$$

In combination with proportional feedback VI (4.35) [see line 9 in Algorithm 4.1], the ADP-based procedure to approximate V is given by Algorithm 4.1. The NN approximation at iteration n is $\tilde{V}_n$ and, for each $j\in\{1,2,...,n_{\mathrm{train}}\}$ and $(x,w)=X^j_{\mathrm{train}}$, $\tilde{e}_n(X^j_{\mathrm{train}},V^j_{\mathrm{train}})$

is obtained in analogy to (4.62) and (4.63) by

$$\tilde{e}_n(X^j_{\text{train}}, V^j_{\text{train}}) = \max_{u\in U}\Bigg\{\sum_{p\in I_\theta}\Big[\sum_{q\in I_w}\tilde{V}_n(f(x,u,w,p),w^q) \\ \times P_w(w^q|w,p)\Big]P_\theta(p|\text{nn}_{\tilde{G}}(x),\text{nn}_{\tilde{U}}(u),w)\Bigg\} + 1 - V^j_{\text{train}}. \tag{4.64}$$

To combine the advantages of online and batch learning, a mini-batch approach is employed in Algorithm 4.1, where the NN is updated based on average values of the output error $\tilde{V}_n(X^j_{\text{train}}) - V^j_{\text{train}}$ over subsets of size n_{mb} of the training data; $n_{\text{train}}/n_{\text{mb}} \in \mathbb{Z}_+$ is assumed.

Algorithm 4.1 ADP procedure to approximately obtain the value function

1: $X_{\text{train}} \leftarrow$ randomly generate n_{train} points
2: $V^j_{\text{train}} \leftarrow 0$ for all $j \in \{1, 2, ..., n_{\text{train}}\}$
3: $\tilde{V}_0 \leftarrow 0$
4: $n \leftarrow 0$
5: **while** $\max_{j\in\{1,2,...,n_{\text{train}}\}} |\tilde{e}_n(X^j_{\text{train}}, V^j_{\text{train}})| > \varepsilon$ **do**
6: **for** $i \leftarrow 1$ to $n_{\text{train}}/n_{\text{mb}}$ **do**
7: $q \leftarrow (i-1)n_{\text{mb}}$
8: **for** $j \leftarrow 1$ to n_{mb} **do**
9: $V^{q+j}_{\text{train}} \leftarrow V^{q+j}_{\text{train}} + k\tilde{e}_n(X^{q+j}_{\text{train}}, V^{q+j}_{\text{train}})$
10: **end for**
11: $\tilde{V}_{n+1} \leftarrow$ update NN using $\{X^{q+1}_{\text{train}}, ..., X^{q+n_{\text{mb}}}_{\text{train}}\}$ and $\{V^{q+1}_{\text{train}}, ..., V^{q+n_{\text{mb}}}_{\text{train}}\}$
12: **end for**
13: $n \leftarrow n + 1$
14: **end while**
15: $\tilde{V} \leftarrow \tilde{V}_n$

The approximation $\tilde{\pi}^*$ of an optimal control policy follows from (4.61), where V [in (4.60)] is replaced by $\tilde{V}$. The parameter ε in Algorithm 4.1 is a convergence threshold that is set to $\varepsilon = 0.01$. In some cases in Section 4.5.4, the maximum error $\max|\tilde{e}_n|$ in line 5 of Algorithm 4.1 approaches a value close to (but greater than) ε before increasing to large values. While the maximum error may eventually decrease to small values again and satisfy $\max|\tilde{e}_n| < \varepsilon$, this may take considerably long. Hence, the convergence criterion is slightly modified for the case study in Section 4.5.4. In addition to $\max|\tilde{e}_n| < 0.01$, Algorithm 4.1 is considered to be converged if $\max|\tilde{e}_n| < 0.05$ and $\max|\tilde{e}_{n+1}| > \max|\tilde{e}_n|$. On the other hand, if $\max|\tilde{e}_{100+n}| > 1$ for some $n \in \mathbb{Z}_{\geq 0}$, Algorithm 4.1 is considered not to be converged. Numerical studies show that the difference between the control policies based

on $\max|\tilde{e}_n| < 0.01$ and the modified convergence criterion is small, except the former may require substantially more computation time.

4.5.4 Numerical Case Study

Closed-loop simulation results based on the driving model developed in Section 4.5.1 are presented, where the respective model probabilities P_W and P_θ are calibrated using traffic observations. For each possible event, the respective probability is approximated by dividing the number of times that the event is observed by the total number of observations. The traffic data is obtained using the traffic simulator in [105] for a two-lane road involving 15 cars in the immediate vicinity of the controlled/ego car (car length: $d = 6$ m). The driving simulator in [105] considers discrete vehicle velocities and accelerations. In this case study, the surrounding cars can travel with five different velocities defined by

$$\mathcal{V} = \{17.\bar{2}, 19.7\bar{2}, 22.\bar{2}, 24.7\bar{2}, 27.\bar{2}\}\,\mathrm{m/sec},$$

which corresponds to $\mathcal{V} = \{62, 71, 80, 89, 98\}\,\mathrm{km/h}$. Note that a bar over a digit indicates that the digit is repeating, e.g., $0.\bar{2} = 0.2222...$. The study of more realistic traffic situations (involving a denser or continuous velocity set) is left for future work.

For the relative time gaps $T_{\mathrm{g,c}}$ and $T_{\mathrm{g,o}}$, $\mathcal{T} = \{0.5, 1, 1.5, 2, 2.5, 3\}\,\mathrm{sec}$ is chosen as the corresponding discrete set and the maximum headway at which a car can be detected is $s_{\max} = 90\,\mathrm{m}$. The acceleration a_{m} of the controlled car can take values in the set $\mathcal{A} = \{-2.5, 0, 2.5\}\,\mathrm{m/sec^2}$. Hence, together with the lane change indicator $l_{\mathrm{m}} \in \{0, 1\}$, the control constraints are given by $U = \mathcal{A} \times \{0, 1\}$, and $\tilde{U} = U$. The model sampling time is set to $\Delta t = 1\,\mathrm{sec}$, whereas the time for completing lane changes may be greater than 1 sec; this is the case in the chosen traffic simulator [105]. However, the driving model does not depend on the actual time for lane changes because a lane change, initiated at time instant t, is always considered to be completed at $t + 1$ (after Δt), even though the car may still be traveling towards the new lane at $t + 1$.

Initial results showed that the DCOC policy frequently initiates lane changes, which may be dissatisfying for passengers. Therefore, similar to [103], a fourth state L is introduced in order to limit the number of lane changes. This additional state evolves according to

$$L_{t+1} = \min\{L_{\max}, L_t + \Delta t/\mathrm{sec}\} - L_{\max} l_{\mathrm{m},t}, \tag{4.65}$$

where $L_{\max}/\Delta t$ is equivalent to the minimum amount of time between two lane changes,

which is enforced by the constraint $L \geq 0$. Thus, the state vector is given by

$$x = [s_{\mathrm{c}}, s_{\mathrm{o}}, v_{\mathrm{m}}, L]^{\top},$$

and the constraints for the DCOC problem (4.58) considered in this case study are given by the following set

$$G = \{x : T_{\mathrm{g,c}} \geq 0.5\,\mathrm{sec}, L \geq 0, v_{\min} \leq v_{\mathrm{m}} \leq v_{\max}\}, \tag{4.66}$$

where $v_{\min}$ and $v_{\max}$ are the controlled car's minimum and maximum velocity, respectively. Instead of using the headway directly, the relative time gap $T_{\mathrm{g,c}}$, defined in (4.50), is used to acknowledge the influence of different traveling speeds. Note that in real traffic applications, an additional controller would be required to resolve cases of constraint violation, i.e., when $T_{\mathrm{g,c}} < 0.5\,\mathrm{sec}$, and recover constraint satisfaction. This may be achieved by ignoring the constraints on L and v_{m} and allowing for larger decelerations/accelerations. As the design of such a controller is not considered in this chapter, simulations are terminated when constraints are violated.

In the following, $n_{\mathrm{train}} = 2500$ training points are used with a mini-batch size of $n_{\mathrm{mb}} = 50$ and the momentum rate in the back-propagation algorithm is set to 0.85. The time t_{comp} to compute the DCOC policy and the required number of iterations N_{iter} until convergence (the DCOC policy is computed offline by Algorithm 4.1) are reported. For the ADP approach, these values are average values over 50 samples to account for different training sets X_{train}, since X_{train} is generated randomly in Algorithm 4.1. Moreover, for each training set, 200 different closed-loop simulations are performed and the average first exit-time $\bar{\tau}$ (averaged over $50 \times 200 = 10000$ samples) is reported. The initial condition for the closed-loop simulations is given by $s_{\mathrm{c},0} = 90\,\mathrm{m}$, $s_{\mathrm{o},0} = 50\,\mathrm{m}$, $v_{\mathrm{m},0} = 22.\bar{2}\,\mathrm{m/sec}$, $L_0 = L_{\max}$, $v_{\mathrm{c},0} = 22.\bar{2}\,\mathrm{m/sec}$, and $v_{\mathrm{o},0} = 19.7\bar{2}\,\mathrm{m/sec}$, and the minimum and maximum velocities for the controlled/ego car in (4.66) are $v_{\min} = 19.7\bar{2}\,\mathrm{m/sec}$ and $v_{\max} = 27.\bar{2}\,\mathrm{m/sec}$.

The procedure is parallelized and implemented in C, where all computations are performed on a computing node with 12 cores (2.67 GHz Intel Xeon X5650 processors) and 48 GB RAM.[1] For a learning rate of $\lambda = 5 \times 10^{-4}$ in the back-propagation algorithm and a proportional gain of $k = 1.6$, Table 4.1 shows the computation time, number of iterations until convergence, and the average first exit-time for different NN structures for the case $L_{\max} = 10$. Out of the four NNs in Table 4.1, the 18-12-6-1L NN yields the best

[1] Flux computing cluster, Advanced Research Computing - Technology Services, University of Michigan, http://arc-ts.umich.edu.

control performance and fastest computation time. The performance of the more complex 24-18-12-6-1L NN may be improved by increasing the number of training points, which, however, would increase the computation time. The 18-12-6-1L NN is used for all subsequent computations in this section.

NN	t_{comp} (sec)	N_{iter}	$\bar{\tau}$ (sec)
6-1L	46.5	173.5	36.4
12-6-1L	14.8	64.9	285.4
18-12-6-1L	14	45.9	814.6
24-18-12-6-1L	23.8	40.2	698.8

Table 4.1: Autonomous driving problem – ADP ($L_{\max} = 10$), performance of different NN: computation time t_{comp} to obtain DCOC policy, number of iterations N_{iter} until convergence, and average first exit-time $\bar{\tau}$.

Table 4.2 summarizes t_{comp}, N_{iter}, and $\bar{\tau}$ for $L_{\max} = 10$, different learning rates λ (back-propagation algorithm), and different k (proportional feedback VI). Moreover, the table shows the number of samples that did not converge (out of 50 total samples). While $\bar{\tau}$ is nearly the same for all configurations of k and λ ($\bar{\tau} \approx 820$ sec, except for $k = 1.9$ and $\lambda = 7.5 \times 10^{-4}$), increasing k and λ improves the convergence rate and computation time. If λ is too large, convergence may not be achieved. Of the configurations in Table 4.2, the fastest convergence is achieved with $\lambda = 5 \times 10^{-4}$ and $k = 1.9$ (10 sec and 32.4 iterations) or with $\lambda = 7.5 \times 10^{-4}$ and $k = 1.8$ (10.6 sec and 22 iterations). For the three learning rates in Table 4.2 and $L_{\max} = 10$, N_{iter} is plotted against k in Figure 4.2, where $\lambda = 7.5 \times 10^{-4}$ and $k = 1.725$ yield the fastest convergence rate (7.9 sec and 18.2 iterations).

For $L_{\max} = 10$, Table 4.3 shows t_{comp}, N_{iter}, and $\bar{\tau}$ (also averaged over 10000 samples) for conventional DP [with $k = 1$ in (4.35)], using different grid sizes. It can be seen that the first exit-time improves with increasing grid size. On the other hand, increasing the grid size increases the computation time nearly exponentially. While, for denser grids, conventional DP achieves slightly better results than the ADP approach, computation times are significantly longer as expected due to the curse of dimensionality. This is a limiting factor for applications of DP to more complex and higher-dimensional traffic scenarios, which may be treated with the ADP approach.

Example trajectories for the ADP approach and conventional DP are plotted in Figures 4.3 and 4.4, respectively, showing the relative time gaps $T_{\mathrm{g,c}}$ and $T_{\mathrm{g,o}}$, the velocity v_{m} of the controlled car, and the lane change indicator l_{m}. The dashed lines indicate the state and

control constraints. A distinguishing feature of the two approaches, which can be observed in other simulation samples as well, is that the ADP-based policy regulates the velocity to the prescribed minimum value, whereas the conventional DP policy changes the velocity more frequently. Since both approaches provide approximations of the optimal solution (yielding similar exit-times), it is not clear which strategy is more effective in maximizing the expected first exit-time. Moreover, the optimal solution (if one exists) may not be unique.

k	t_{comp} (sec)	N_{iter}	no convergence	$\bar{\tau}$ (sec)
1.0	36.4 [18.6] (18.1)	118 [60.1] (40.7)	0 [0] (36)	828.4 [828] (817.9)
1.1	35.9 [17.5] (15.9)	115 [56.9] (37.5)	0 [0] (15)	819.8 [819.2] (814.8)
1.2	35 [16.6] (14.8)	112.5 [54.2] (34.6)	0 [0] (1)	819.8 [816.3] (819.2)
1.3	34.4 [15.9] (13.8)	110.2 [51.9] (31.9)	0 [0] (0)	826.6 [823.2] (817.8)
1.4	33.7 [15.3] (13.5)	108.2 [49.6] (29.3)	0 [0] (0)	838.1 [804] (834.5)
1.5	32.9 [14.8] (13.1)	106.5 [47.7] (26.9)	0 [0] (0)	825.1 [810] (828.6)
1.6	32.7 [14] (11.5)	104.9 [45.9] (24.6)	0 [0] (0)	828.2 [814.6] (813.8)
1.7	32 [13.6] (10.3)	102.9 [44.3] (20.7)	0 [0] (0)	816.6 [821.6] (818.2)
1.8	31.3 [13.2] (10.6)	100.2 [42.8] (22)	0 [0] (0)	818.1 [816.5] (813.9)
1.9	27.6 [10] (82.5)	88.2 [32.4] (157.3)	0 [0] (38)	828.5 [817.6] (82)
1.92	22.7 [11.7] (n/a)	54.6 [37.5] (n/a)	0 [0] (50)	815.2 [826.5] (n/a)

Table 4.2: Autonomous driving problem – ADP ($L_{\max} = 10$): computation time t_{comp} to obtain DCOC policy, number of iterations N_{iter} until convergence, number of samples without convergence, and average first exit-time $\bar{\tau}$ for $\lambda = 2.5 \times 10^{-4}$ [$\lambda = 5 \times 10^{-4}$] ($\lambda = 7.5 \times 10^{-4}$).

The influence of $L_{\max}$ and $v_{\min}$ on the first exit-time can be seen in Table 4.4 and Figure 4.5. Figure 4.5 shows $\bar{\tau}$ plotted over $L_{\max}$. As expected, increasing $L_{\max}$, i.e., decreasing the allowed maximum frequency for lane changes, yields smaller $\bar{\tau}$. In contrast to the previous results where $v_{\min} = 19.7\bar{2}\,\mathrm{m/sec}$, Table 4.4 lists the average first exit-time, based on the ADP approach, for different $v_{\min}$ and $L_{\max} = 10$. When $v_{\min}$ is equal to the lowest possible velocity of the surrounding cars, the optimal first exit-time is indefinite. For $v_{\min} > 17.\bar{2}\,\mathrm{m/sec}$, $\bar{\tau}$ decreases with increasing $v_{\min}$.

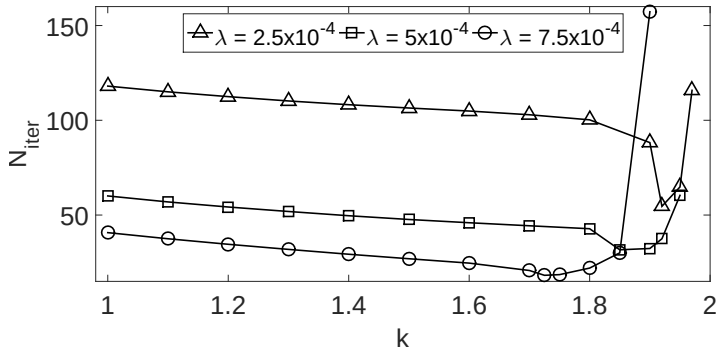

Figure 4.2: Autonomous driving problem – ADP ($L_{\max} = 10$): number of iterations N_{iter} until convergence vs. proportional gain k for different λ.

$n_{s_\mathrm{c}} \times n_{s_\mathrm{o}} \times n_L$	t_{comp} (sec)	N_{iter}	$\bar{\tau}$ (sec)
$2 \times 2 \times 2$	19.8	1352	472.9
$3 \times 3 \times 3$	83	2427	545.4
$4 \times 4 \times 4$	100.3	1555	709
$5 \times 5 \times 5$	230.7	2246	779.7
$6 \times 6 \times 6$	367.8	2424	825.2
$7 \times 7 \times 7$	561.7	2634	857.5
$8 \times 8 \times 8$	988.1	3126	879.4

Table 4.3: Autonomous driving problem – conventional DP ($L_{\max} = 10$): computation time t_{comp} to obtain DCOC policy, number of iterations N_{iter} until convergence, and average first exit-time $\bar{\tau}$ for different grids $n_{s_\mathrm{c}} \times n_{s_\mathrm{o}} \times n_{v_\mathrm{m}} \times n_L \times n_{v_\mathrm{c}} \times n_{v_\mathrm{o}}$ (number of discrete values considered for each variable), where $n_{v_\mathrm{m}} = 4$ and $n_{v_\mathrm{c}} = n_{v_\mathrm{o}} = 5$ are fixed.

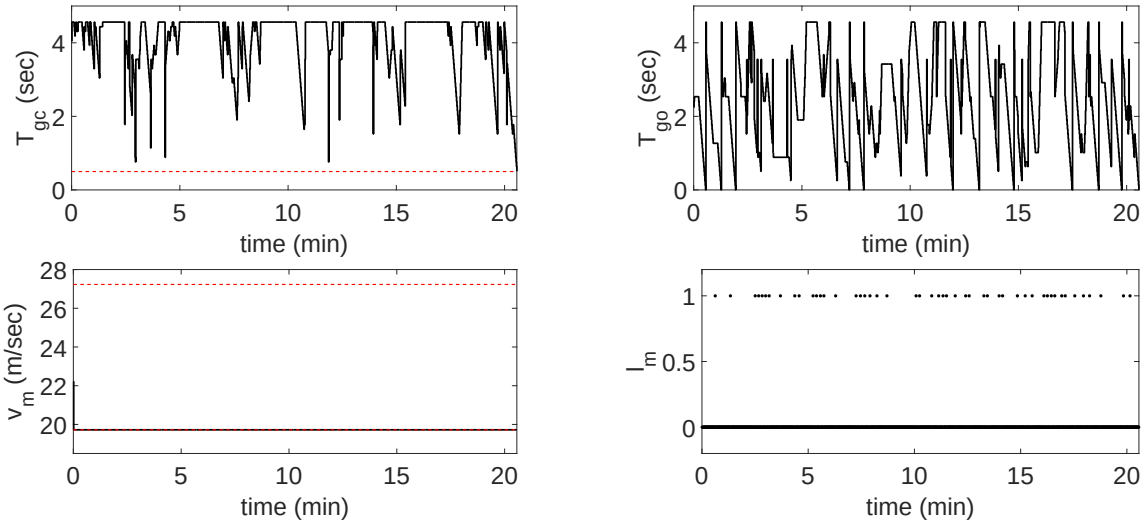

Figure 4.3: Autonomous driving problem – ADP ($L_{\max} = 10$): sample trajectories of relative time gap $T_{\mathrm{g,c}}$ for the ego car's current lane (top left), relative time gap $T_{\mathrm{g,o}}$ for other lane (top right), velocity v_{m} of ego car (bottom left), and lane change indicator l_{m} of ego car (bottom right) over time.

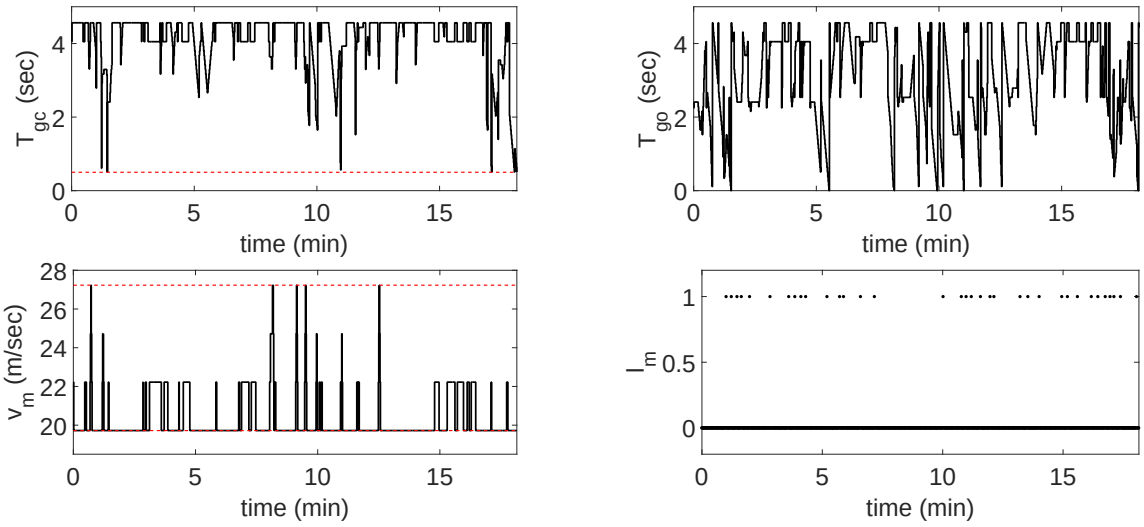

Figure 4.4: Autonomous driving problem – conventional DP ($L_{\max} = 10$): sample trajectories of relative time gap $T_{\mathrm{g,c}}$ for the ego car's current lane (top left), relative time gap $T_{\mathrm{g,o}}$ for other lane (top right), velocity v_{m} of ego car (bottom left), and lane change indicator l_{m} of ego car (bottom right) over time.

$v_{\min}$ (m/sec)	$17.\bar{2}$	$19.7\bar{2}$	$22.\bar{2}$	$24.7\bar{2}$	$27.\bar{2}$
$\bar{\tau}$ (sec)	∞	820	200.7	85.4	36.9

Table 4.4: Autonomous driving problem – ADP ($L_{\max} = 10$): average first exit-time $\bar{\tau}$ for different $v_{\min}$.

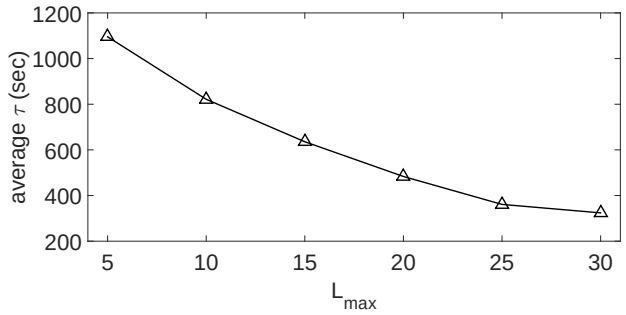

Figure 4.5: Autonomous driving problem – ADP: average first exit-time $\bar{\tau}$ vs. $L_{\max}$.

4.6 Other Numerical Case Studies

Other numerical case studies of stochastic DCOC problems are considered in this section. The first problem in Section 4.6.1 is the control of a pendulum under random disturbance. Section 4.6.2 considers glider flight management. An ACC problem is treated in Section 4.6.3. Each respective problem is solved using adaptive proportional feedback VI with damping from Section 4.4.2, where iterations (4.49) are applied to a discrete subset (grid) $\tilde{G}$ of G, using linear interpolation to evaluate V_n between the grid points. The convergence criterion is set to $\max_{(x,w)\in\tilde{G}\times W} |e_n(x,w)| \leq 0.01$ and $V_0 \equiv 1$.

4.6.1 Stochastic Pendulum

Consider a pendulum subject to a random horizontal disturbance force F_w, where F_w is modeled by a Markov chain. The objective is to maximize the expected value of the first exit-time (i.e., $g \equiv 1$). The discrete-time model is obtained from the continuous-time model using Euler's forward method. By denoting the pendulum's angle by ϕ and its angular velocity by ω, the discrete-time model reads

$$\begin{aligned}\phi_{t+1} &= \phi_t + \Delta t \omega_t, \\ \omega_{t+1} &= \omega_t + \Delta t \left[\frac{M_{u,t}}{ml^2} - \frac{g_{\mathrm{E}}}{l} \sin(\phi_t) - \frac{F_{w,t}}{ml} \cos(\phi_t) \right],\end{aligned} \tag{4.67}$$

with a sampling time of $\Delta t = 0.1$ sec. The control input at a time instant t is $M_{u,t} \in U = [-1, 1]$ Nm. The length of the pendulum is $l = 1$ m, its mass is $m = 1$ kg, and $g_{\mathrm{E}} = 9.81$ $\mathrm{m/sec}^2$. The disturbance takes values from the set $W = \{-1.75, -0.75, 0, 0.75, 1.75\}$ N and the associated matrix of transition probabilities is

$$P_W = \begin{bmatrix} 0.25 & 0.4 & 0.35 & 0 & 0 \\ 0.35 & 0.2 & 0.25 & 0.2 & 0 \\ 0.2 & 0.3 & 0.25 & 0.15 & 0.1 \\ 0.1 & 0.1 & 0.5 & 0.2 & 0.1 \\ 0.2 & 0.1 & 0.3 & 0.2 & 0.2 \end{bmatrix}.$$

The state constraints for this case study are given by the set

$$G = \{\phi, \omega : \ \phi \in [-0.4, 0.4] \text{ rad}, \ \omega \in [-1.2, 1.2] \text{ rad/sec}\},$$

which is discretized using an equidistant grid of 9 points ranging from -0.4 to 0.4 rad for ϕ and an equidistant grid of 25 points from -1.2 to 1.2 rad/sec for ω. For numerical reasons, the set U needs to be discretized as well, where an equidistant grid of 21 points from -1 to 1 Nm is used. The initial gains are set to $k_0 \equiv 1$.

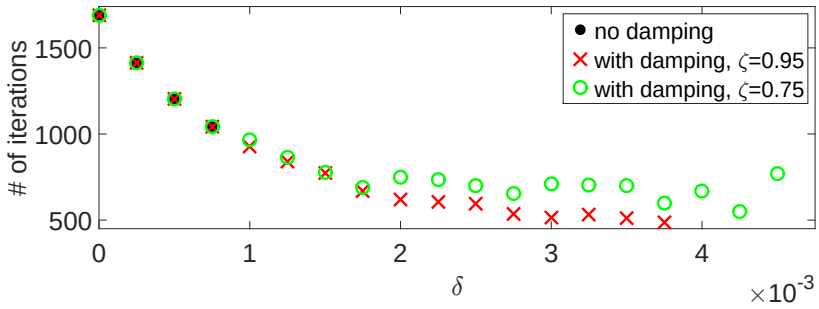

Figure 4.6: Stochastic pendulum problem: number of iterations until convergence vs. learning rate δ.

Figure 4.6 shows the required number of iterations until convergence for different values of the learning rate δ using no damping (black dots), moderate damping with $\zeta = 0.95$

(red x's), and increased damping with $\zeta = 0.75$ (green circles). For $\delta = 0$ and no damping, the algorithm is identical to conventional VI which requires 1690 iterations to converge. The convergence rate improves with increasing δ. There is no difference between the damped and undamped algorithms up to $\delta \geq 0.001$ for which the undamped algorithm fails to converge. With damping, however, the algorithm continuous to converge. The convergence rate with moderate damping ($\zeta = 0.95$) is faster than with increased damping ($\zeta = 0.75$). The moderate damping approach fails to converge for $\delta \geq 0.004$, whereas the increased damping approach converges up to $\delta = 0.00475$. For $\delta = 0.00375$ and $\zeta = 0.95$, the algorithm requires 488 iterations to converge, which is more than three times as fast as with conventional VI.

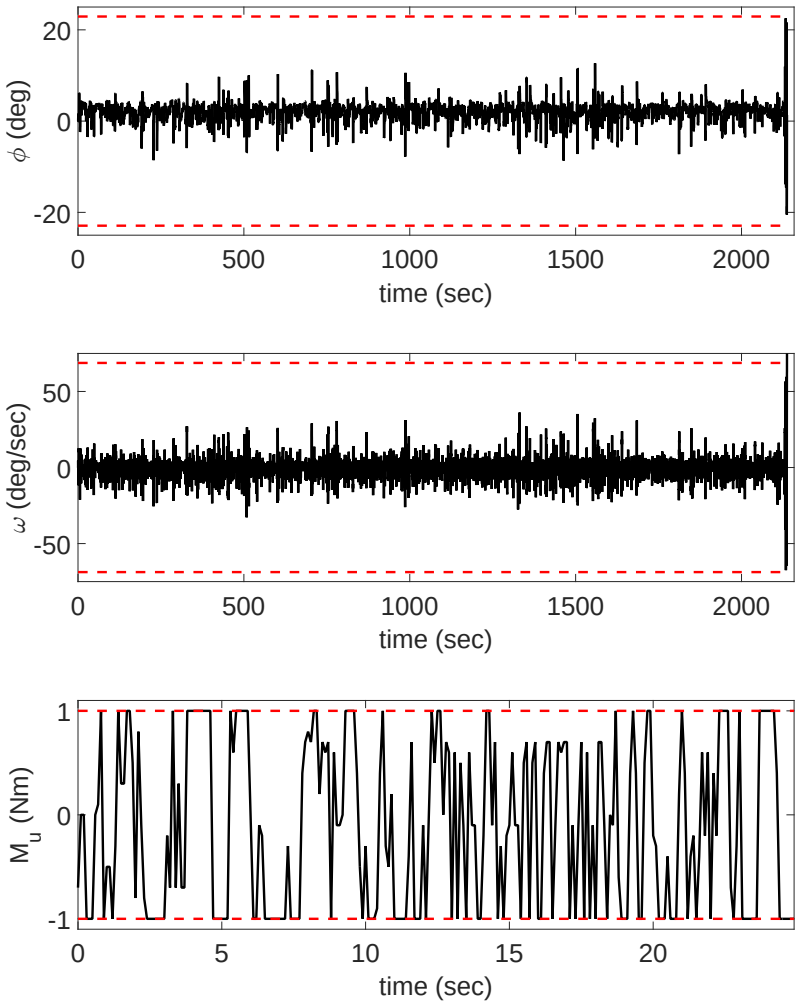

Figure 4.7: Stochastic pendulum problem – sample results for some random disturbance profile. Top: angle ϕ vs. time. Middle: angular velocity ω vs. time. Bottom: control input during the first 25 sec vs. time.

Figure 4.7 shows sample trajectories of the states ϕ and ω over time (top and middle plot) that result from applying the DCOC-based control policy to the pendulum for some random disturbance profile. The disturbance is counteracted while the system stays inside the prescribed set (dashed red lines in Figure 4.7) until about 2100 sec elapsed time. Note that the value functions, approximated based on adaptive proportional feedback VI with damping, are all identical (within numerical tolerances) for the different parameter settings in Figure 4.6. The corresponding control input is shown in Figure 4.7 (bottom) for the first 25 sec, where the dashed red lines indicate the control constraints.

4.6.2 Glider Flight Management

In this section, a stochastic DCOC problem of glider flight management is treated in order to maximize the expected time of flight. Let $h\,[\mathrm{ft}]$ be the altitude of the glider and $s\,[\mathrm{miles}]$ its range (relative to a reference point). The following simplified model is considered to describe the glider's flight,

$$\begin{aligned} h_{t+1} &= h_t + (1-\sigma_t)(\tau_t w_{\mathrm{th},t} + \Delta t(w_{\mathrm{d},t} - v_{z,t})(1-\tau_t)) \\ &\quad - \sigma_t(\Delta h_{\mathrm{turn}}(1-\tau_t) - \tau_t w_{\mathrm{th},t}), \\ s_{t+1} &= s_t + \Delta t(1-\sigma_t)(1-\tau_t) d_t v_t/3600, \\ d_{t+1} &= d_t - 2\sigma_t \mathrm{sgn}(d_t), \end{aligned} \tag{4.68}$$

where $\Delta t = 60$ sec and $d \in \{-1, 1\}$ indicates the direction of flight. The variables $w_{\mathrm{th}} \in \{0, 100, 200\}$ ft (thermal strength) and $w_{\mathrm{d}} \in \{-6.67, -3.33, 0, 3.33\}$ ft/sec (updraft strength) are random inputs modeled by a Markov chain. In this case study, the transition probabilities are state-dependent. The developed DCOC framework is readily extended to consider state-dependent transition probabilities by replacing $P_W(w^i|w)$ by $P_W(w^i|w,x)$ in (4.18), where x is the state vector. In this case, $x = [h, s, d]^\top$ and $w = [w_{\mathrm{th}}, w_{\mathrm{d}}]^\top$.

For w_{th}, there are two transition probability matrices $P_{\mathrm{th},s\leq 2}$ and $P_{\mathrm{th},s>2}$, where $P_{\mathrm{th},s\leq 2}$ is the transition probability matrix for $s \leq 2$ miles and $P_{\mathrm{th},s>2}$ for $s > 2$ miles. $P_{\mathrm{th},s\leq 2}$ and $P_{\mathrm{th},s>2}$, respectively, are given by

$$\begin{bmatrix} 0.7 & 0.25 & 0.05 \\ 0.3 & 0.55 & 0.15 \\ 0.2 & 0.4 & 0.4 \end{bmatrix}, \begin{bmatrix} 0.75 & 0.05 & 0.2 \\ 0.3 & 0.3 & 0.4 \\ 0.55 & 0.2 & 0.25 \end{bmatrix}.$$

Similarly, two state-dependent transition probability matrices $P_{\mathrm{d},s\leq 3}$ and $P_{\mathrm{d},s>3}$ are consid-

ered for w_{d}, which are, respectively, given by

$$\begin{bmatrix} 0.04 & 0.25 & 0.7 & 0.01 \\ 0.1 & 0.25 & 0.55 & 0.1 \\ 0.05 & 0.1 & 0.7 & 0.15 \\ 0.01 & 0.04 & 0.75 & 0.2 \end{bmatrix}, \begin{bmatrix} 0.02 & 0.07 & 0.83 & 0.08 \\ 0.01 & 0.08 & 0.81 & 0.1 \\ 0.01 & 0.03 & 0.65 & 0.31 \\ 0.01 & 0.05 & 0.54 & 0.4 \end{bmatrix}.$$

The control variables are $\tau \in \{0, 1\}$ (indicating whether to climb a thermal or not), $\sigma \in \{0, 1\}$ (indicating whether to turn 180 deg or not), and the horizontal velocity

$$v \in \{34.0, 36.0, 38.0, 40.0, 42.0, 44.0, 46.0, 48.0, 52.0, 60.0, 66.0\}\,\mathrm{mph}.$$

The glider is assumed to perform a turn within 60 sec and the altitude decreases by $\Delta h_{\mathrm{turn}} = 250$ ft during each turn. Note that it is also possible to turn while climbing a thermal. The sink rate v_z in (4.68) is a function of the horizontal velocity [39]: $v_z = 8.2582 - 0.287v + 0.0038v^2$.

The objective is to maximize the expected first exit-time from the set

$$G = \{x :\, h \in [1000, 3000]\,\mathrm{ft},\; s \in [0, 5]\,\mathrm{miles}\},$$

where both altitude and range constraints are taken into account. The set G is discretized with an equidistant grid comprising 15 points for each h and s. The initial gains are set to $k_0 \equiv 1.9$.

Two different damping configurations are analyzed: $\zeta = 0.95$ (less damping) and $\zeta = 0.93$ (more damping). The number of iterations until convergence is shown for different learning rates δ in the top of Figure 4.8. As in the previous case study (Section 4.6.1), larger δ are possible with more damping. The fastest convergence is achieved with $\delta = 0.0022$ for $\zeta = 0.95$ (53 iterations) and with $\delta = 0.014$ for $\zeta = 0.93$ (53 iterations). This corresponds to a computation time of about 0.3 sec for a C implementation on a computing cluster with 12 cores. In contrast, conventional VI requires 205 iterations to converge (about 4 times as long). Note that the algorithm does not converge (for $\delta \geq 0$) without damping due to the large initial gains, $k_0 \equiv 1.9$.

Sample trajectories of the altitude h and range s when using the computed DCOC policy are plotted over time in the middle and bottom, respectively, of Figure 4.8, where the constraints are indicated by dotted red lines.

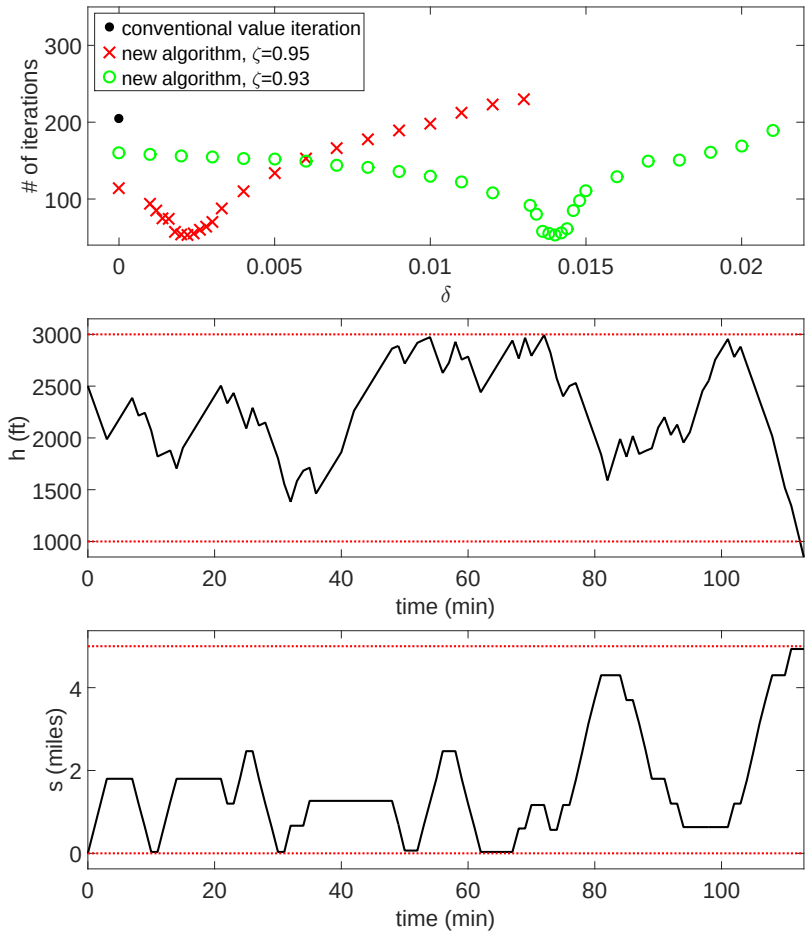

Figure 4.8: Glider flight management problem: number of iterations until convergence vs. learning rate δ (top) and example trajectories showing the altitude h vs. time (middle) and the range s vs. time (bottom).

4.6.3 Adaptive Cruise Control

A stochastic ACC problem, similar to the problem in [37], is treated. Two vehicles are involved, the lead vehicle and the follower vehicle. The objective is to control the speed of the follower vehicle, v_{f}, such that the time gap between the two vehicles, $T_{\mathrm{g}} = s/v_{\mathrm{f}}$, where s is the relative distance, stays within prescribed bounds for as long as possible. The speed of the lead vehicle v_{l} is modeled by a Markov chain that takes values in the set $W = \{26.4, 26.645, ..., 31.3\}$ m/sec, containing 21 elements. The model is given by

$$\begin{aligned} s_{t+1} &= s_t + \Delta t(v_{\mathrm{l},t} - v_{\mathrm{f},t}), \\ v_{\mathrm{f},t+1} &= v_{\mathrm{f},t} + \Delta t a_t, \end{aligned} \tag{4.69}$$

and the sampling time is $\Delta t = 1$ sec. The control input at a time instant t is

$$a_t \in \{-0.25, -0.125, 0, 0.125, 0.25\} \text{ m/sec}^2,$$

which is the acceleration of the follower vehicle. The transition probabilities of the lead vehicle velocity are similar to the values in [37], which are based on experimental data. Moreover, as in [37], the possibility of another vehicle cutting in upfront is taken into account by slightly modifying the model. Such an event may occur with a probability of 0.1 if the time gap T_{g} is greater than 2.2 sec. In case of another vehicle cutting in upfront, the distance between the vehicles is set to half of the previous distance. Moreover, the vehicle is assumed to cut in with a speed of 28.85 m/sec.

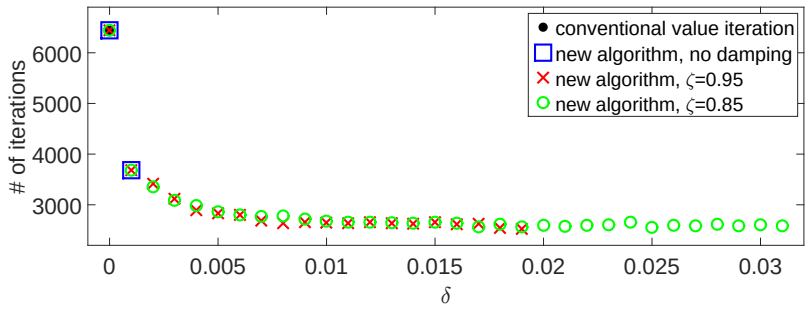

Figure 4.9: Adaptive cruise control problem, DP-based solution: number of iterations until convergence vs. learning rate δ.

The state constraints for this problem are given by the set

$$G = \{s, v_{\text{f}} : T_{\text{g}} = s/v_{\text{f}} \in [0.5, 2.5] \text{ sec}, \ v_{\text{f}} \leq 30 \,\text{m/sec}\},$$

which is discretized using an equidistant grid of 15 points from 10 to 80 m for s and 10 points from 25.4 to 30 m/sec for v_{f}.

The initial gains are set to $k_0 \equiv 1$. Figure 4.9 shows the number of iterations until convergence for different learning rates δ and damping configurations, where the algorithm fails to converge for $\delta > 0.001$ without using damping. With damping, on the other hand, δ can be further increased which improves the convergence rate. The fastest convergence with a damping factor of $\zeta = 0.95$ occurs for $\delta = 0.019$ (2521 iterations, 3.2 sec computation time for a C implementation on a 12-core cluster) before the algorithm fails to converge ($\delta > 0.019$). Larger learning rates are possible with more damping ($\zeta = 0.85$), where the fastest convergence is achieved with $\delta = 0.025$ (2549 iterations). In contrast, conventional VI ($\delta = 0$, no damping) requires 6442 iterations, which is more than twice

as long. Note that the different configurations in Figure 4.9 converge to the same value function (within the prescribed tolerance).

Sample simulation results are plotted in Figure 4.10, including the time history of the time gap T_{g} (top left), the velocity of the follower vehicle v_{f} (top right), the control input (bottom left), and the disturbance, i.e., lead vehicle velocity v_{l} (bottom right). The respective constraints are indicated by dotted red lines. On average (1000 random simulations), constraint violation occurs after 2591 sec (43.2 min). As can be seen from the follower vehicle velocity and acceleration plots (top right and bottom left plots in Figure 4.10), frequent velocity changes are performed by the control policy, which may be uncomfortable for passengers and inefficient in terms of fuel consumption. This issue may be addressed by adding a control input penalty to the objective function, which is done in Section 5.5.2.

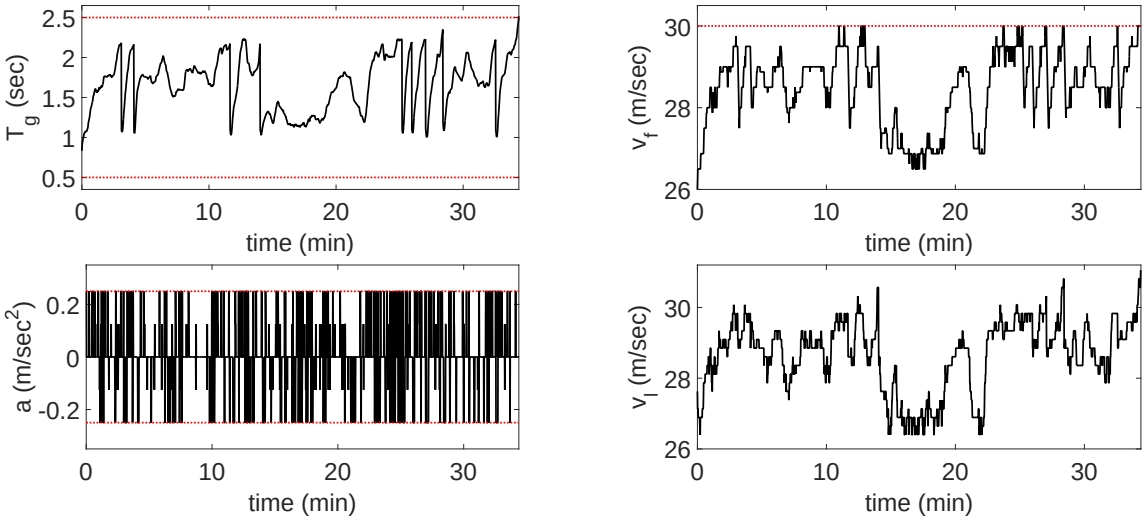

Figure 4.10: Adaptive cruise control problem, DP-based solution: sample trajectories over time of time gap T_{g} between the two vehicles (top left), follower vehicle velocity v_{f} (top right), acceleration of follower vehicle a (bottom left), and lead vehicle velocity v_{l} (bottom right).

4.7 Summary

This chapter treated the DCOC problem for stochastic systems using DP techniques. Theoretical properties of the problem were studied, including conditions under which the expected value of the first exit-time and the value function are bounded from above, which are necessary conditions for the existence of a solution, as well as characteristics of a solution and additional conditions under which a solution exists. An enhanced version of the VI algorithm was proposed. Numerical case studies of stochastic DCOC problems of

ACC, glider flight management, and pendulum control demonstrated that the enhanced VI algorithm is able to obtain an approximation of the value function faster than conventional VI for proper parameter settings.

Furthermore, based on stochastic DCOC, an approach for high-level control / decision-making for autonomous cars was presented. A probabilistic driving model for a two-lane road, which may be extended to multiple lanes in future work, was developed. Based on this model, the stochastic DCOC problem was formulated with the objective of maintaining a prescribed safe headway to the car in front for as long as possible (on average). An ADP approach based on proportional feedback VI was proposed to obtain an approximate solution. Numerical results showed that this approach can be advantageous compared to conventional DP techniques. Moreover, in contrast to conventional DP, the ADP approach may be able to treat higher-dimensional problems (e.g., multiple lanes and more complex traffic scenarios).

CHAPTER 5

Stochastic DCOC – Tree-Based SMPC

5.1 Problem Formulation

The focus of this chapter is on stochastic DCOC problems with the objective of maximizing the expected value of the first exit-time, i.e., $g \equiv 1$ in (4.7). Moreover, stochastic linear systems of the form

$$x_{t+1} = A_t x_t + B_t u_t + w_t, \tag{5.1}$$

are considered, where A_t and B_t are time-dependent matrices of proper dimension and w_t denotes a random disturbance at a time instant $t \in \mathbb{Z}_{\geq 0}$. As in Chapter 4, w is modeled by a Markov chain that takes values in the finite set W, see (4.2), with transition probabilities given by P_W, see (4.3). Using the results of Sections 5.2 and 5.3, an SMPC scheme is proposed in Section 5.4, where the linear-model-based [see (5.1)] solution is recomputed over a moving time horizon based on the current state and disturbance to compensate for unmodeled effects. Thus, the proposed SMPC scheme can be effective in obtaining approximate solutions to DCOC problems involving more general stochastic nonlinear system models.

In analogy to (4.6) and (4.7), the stochastic DCOC problem considered in this chapter is as follows

$$\max_{\pi \in \Pi} \bar{\tau}(x, w, \pi), \tag{5.2}$$

where $\bar{\tau}(x, w, \pi)$ is the average (i.e., the expected value of the) first exit-time, which is defined similarly to (4.5) by

$$\tau(x_0, w_0, \pi) = \inf\{t \in \mathbb{Z}_{\geq 0} : x_t \notin G_t\}, \tag{5.3}$$

where x_t is the response of (5.1) to the initial condition $x_0 \in G_0$ and $w_0 \in W$ when using the control policy

$$\pi \in \Pi = \{\pi : G_t \times W \times \mathbb{Z}_{\geq 0} \to U_t \text{ for all } t \in \mathbb{Z}_{\geq 0}\},$$

i.e., $u_t = \pi(x_t, w_t, t)$ in (5.1). Note that, in contrast to Chapter 4, the state constraints (given by the sets G_t) and the control constraints (given by the sets U_t) can be time-dependent.

Throughout this chapter, the following assumption about the sets G_t and U_t is made.

Assumption 5.1. The sets G_t and U_t are polytopes for all $t \in \mathbb{Z}_{\geq 0}$, where G_t is expressed as follows

$$G_t = \{x : C_t x \leq b_t\}. \tag{5.4}$$

5.2 Scenario Tree

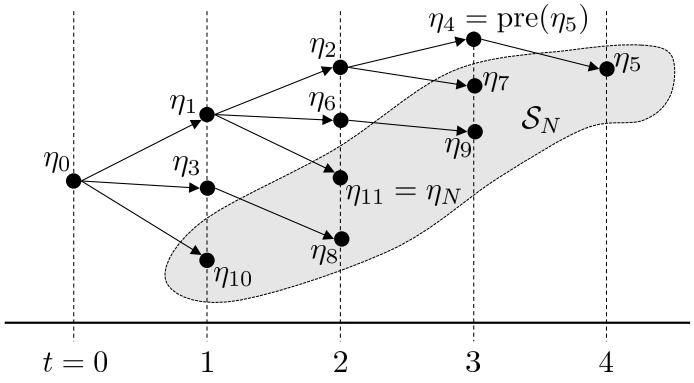

Figure 5.1: Scenario tree example for 12 nodes, including $|\mathcal{S}_N| = 6$ leaf nodes.

In order to optimize over a subset of all possible disturbance scenarios, similar to the work in [106], a scenario tree is constructed that contains the most likely disturbance scenarios for a given number of tree nodes. A tree node is denoted by $\eta \in \mathcal{T}_N$, where

$$\mathcal{T}_N = \{\eta_0, \eta_1, ..., \eta_N\},$$

denotes a tree with $N + 1$ nodes. The node η_0 is the root node of the tree. The predecessor of a node $\eta \in \mathcal{T}_N$ is given by $\mathrm{pre}(\eta)$. The set of successors of a node $\eta \in \mathcal{T}_N$ is denoted by

$$\mathrm{succ}(\eta) = \{\eta_1^{\mathrm{succ}(\eta)}, \eta_2^{\mathrm{succ}(\eta)}, ..., \eta_{|W|}^{\mathrm{succ}(\eta)}\},$$

and the set of leaf nodes of $\mathcal{T}_N$ has the form,

$$\mathcal{S}_N = \{\eta \in \mathcal{T}_N : \mathrm{succ}(\eta) \cap \mathcal{T}_N = \emptyset\}.$$

Figure 5.1 shows an example scenario tree $\mathcal{T}_{11} = \{\eta_0, \eta_1, ..., \eta_{11}\}$ for a given Markov chain with $|W| = 3$. For example, $\mathrm{succ}(\eta_1) = \{\eta_2, \eta_6, \eta_{11}\}$ and $\eta_1^{\mathrm{succ}(\eta_1)} = \eta_2$, $\eta_2^{\mathrm{succ}(\eta_1)} = \eta_6$, and $\eta_3^{\mathrm{succ}(\eta_1)} = \eta_{11}$ in Figure 5.1. The set of leaf nodes is $\mathcal{S}_{11} = \{\eta_5, \eta_7, \eta_8, \eta_9, \eta_{10}, \eta_{11}\}$ in Figure 5.1.

A state vector x^η, control input u^η, disturbance w^η, and time instant t^η are associated with each $\eta \in \mathcal{T}_N$, where $x^{\eta_0} = x_0$, $w^{\eta_0} = w_0$, and $t^{\eta_0} = 0$ for the root node. Moreover, for each $\eta \in \mathcal{T}_N \setminus \{\eta_0\}$, x^η satisfies the dynamics in (5.1),

$$x^\eta = A_{t^{\mathrm{pre}(\eta)}} x^{\mathrm{pre}(\eta)} + B_{t^{\mathrm{pre}(\eta)}} u^{\mathrm{pre}(\eta)} + w^{\mathrm{pre}(\eta)}. \tag{5.5}$$

Algorithm 5.1 Generation of scenario tree $\mathcal{T}_N$

$\mathcal{T}_N \leftarrow \{\eta_0\}$
$\mathcal{C} \leftarrow \emptyset$
$\rho^{\eta_0} \leftarrow 1$
$t^{\eta_0} \leftarrow 0$
$x^{\eta_0} \leftarrow x_0$
$w^{\eta_0} \leftarrow w_0$
$i \leftarrow 0$
while $i < N$ **do**
    for $j \in \{1, 2, ..., |W|\}$ **do**
        $w^{\eta_j^{\mathrm{succ}(\eta_i)}} \leftarrow w^j$ $(w^j \in W)$
        $t^{\eta_j^{\mathrm{succ}(\eta_i)}} \leftarrow t^{\eta_i} + 1$
        $\rho^{\eta_j^{\mathrm{succ}(\eta_i)}} \leftarrow \rho^{\eta_i} P_W(w^j | w^{\eta_i})$
    end for
    $\mathcal{C} \leftarrow \mathcal{C} \cup \mathrm{succ}(\eta_i)$
    $\eta_{i+1} \leftarrow \arg\max_{\eta \in \mathcal{C}} \rho^\eta$ (pick any maximizer)
    $\mathcal{T}_N \leftarrow \mathcal{T}_N \cup \{\eta_{i+1}\}$
    $\mathcal{C} \leftarrow \mathcal{C} \setminus \{\eta_{i+1}\}$
    $i \leftarrow i + 1$
end while

The probability of reaching node $\eta \in \mathcal{T}_N$, starting from the root node, is given by

$$\rho^\eta = \rho^{\mathrm{pre}(\eta)} P_W(w^\eta | w^{\mathrm{pre}(\eta)}) \in [0, 1], \tag{5.6}$$

where $\rho^{\eta_0} = 1$. Algorithm 5.1 implements the scenario tree generation scheme. The set $\mathcal{C}$ contains the candidate nodes that are considered when adding a node to the tree. At each iteration, the node $\eta \in \mathcal{C}$ with the greatest probability ρ^η is chosen from the set of candidate nodes, and the successors of η are added to the list of candidate nodes.

In general, the scenario tree $\mathcal{T}_N$ contains $|\mathcal{S}_N| \geq 1$ unique disturbance trajectories that

are denoted by

$$\{w_t\}^\eta = \{w_t : t \in \mathbb{Z}_{[0,t^\eta]}\}^\eta = (w_0, ..., w^{\mathrm{pre}(\mathrm{pre}(\eta))}, w^{\mathrm{pre}(\eta)}, w^\eta), \tag{5.7}$$

for each leaf node $\eta \in \mathcal{S}_N$. For example, $\{w_t\}^{\eta_9} = (w_0, w^{\eta_1}, w^{\eta_6}, w^{\eta_9})$ in Figure 5.1.

For a given tree $\mathcal{T}_N$ with initial $x = x_0 \in G_0$ and $w = w_0 \in W$ and control policy $\pi_N \in \Pi$, the deterministic first exit-time corresponding to the disturbance trajectory $\{w_t\}^\eta$, see (5.7), is defined by

$$\tau_N^\eta(x, w, \pi_N) = \min\{\min\{t \in \mathbb{Z}_{[0,t^\eta]} : x_t \notin G_t\} \cup \{t^\eta + 1\}\}, \tag{5.8}$$

for each $\eta \in \mathcal{S}_N$, where x_t is the deterministic response of (5.1) under $\{w_t\}^\eta$ when using the control policy $\pi_N \in \Pi$. Note that for some $\{w_t\}^\eta$, x_t may not exit G_t for $t \in \mathbb{Z}_{[0,t^\eta]}$; in this case, $\tau_N^\eta(x, w, \pi_N) = t^\eta + 1$ in line with (5.8). The average first exit-time for a given scenario tree $\mathcal{T}_N$ and control policy $\pi_N \in \Pi$ is given by

$$\bar{\tau}_N(x, w, \pi_N) = \sum_{\eta \in \mathcal{S}_N} \tau_N^\eta(x, w, \pi_N)\rho^\eta. \tag{5.9}$$

In analogy to the stochastic DCOC problem (5.2), the control problem of maximizing the average first exit-time over a subset of disturbance scenarios defined by $\mathcal{T}_N$ is as follows

$$\max_{\pi_N \in \Pi} \bar{\tau}_N(x, w, \pi_N). \tag{5.10}$$

The following sets are defined

$$\mathcal{H}_N^\eta = \{\eta_0, ..., \mathrm{pre}(\mathrm{pre}(\eta)), \mathrm{pre}(\eta), \eta\}, \text{ for all } \eta \in \mathcal{S}_N, \tag{5.11}$$

$$\mathcal{K}_N^\xi = \{\eta \in \mathcal{S}_N : \xi \in \mathcal{H}_N^\eta\}, \text{ for all } \xi \in \mathcal{T}_N, \tag{5.12}$$

where $\mathcal{H}_N^\eta$ is the set of nodes of the disturbance scenario associated with leaf node $\eta \in \mathcal{S}_N$ and $\mathcal{K}_N^\xi$ is the set of leaf nodes whose associated disturbance scenarios contain the node $\xi \in \mathcal{T}_N$. For example, in Figure 5.1,

$$\mathcal{H}_{11}^{\eta_7} = \{\eta_0, \eta_1, \eta_2, \eta_7\} \text{ and } \mathcal{K}_{11}^{\eta_1} = \{\eta_5, \eta_7, \eta_9, \eta_{11}\}.$$

Moreover, for a given control policy $\pi \in \Pi$ and scenario tree $\mathcal{T}_N$, $N \in \mathbb{Z}_+$, with initial condition $x = x_0 \in G_0$ and $w = w_0 \in W$, the set of leaf nodes $\eta \in \mathcal{S}_N$ with associated

first exit-time $\tau_N^\eta(x, w, \pi) = i \in \mathbb{Z}_+$ is given by

$$\mathcal{Z}_N(\pi, i) = \{\eta \in \mathcal{S}_N : \tau_N^\eta(x, w, \pi) = i\}. \tag{5.13}$$

The next result (Theorem 5.1) shows that, in terms of the average first exit-time, a solution to (5.10) is arbitrarily close to a solution (if one exists) of problem (5.2) for sufficiently large N. Theorem 5.1 is based on Lemma 5.1.

Lemma 5.1.

$$\lim_{N\to\infty} \bar{\tau}_N(x, w, \pi) = \bar{\tau}(x, w, \pi), \tag{5.14}$$

for all $x \in G_0$, $w \in W$, and $\pi \in \Pi$.

Proof. Let $\pi \in \Pi$ be a given control policy and $x \in G_0$ and $w \in W$ be a given initial condition. Then, by (5.9),

$$\begin{aligned} \lim_{N\to\infty} \bar{\tau}_N(x, w, \pi) &= \lim_{N\to\infty} \sum_{\eta\in\mathcal{S}_N} \tau_N^\eta(x, w, \pi)\rho^\eta \\ &= \lim_{N\to\infty} \left(\sum_{i=1}^{t_N} i \sum_{\eta\in\mathcal{Z}_N(\pi,i)} \rho^\eta \right), \end{aligned} \tag{5.15}$$

where $t_N = \max\{t^\eta : \eta \in \mathcal{T}_N\} + 1$. Since W is a finite set, it follows from the tree generation procedure (Algorithm 5.1) that eventually every branch corresponding to non-zero probability of next disturbance value continuous. Thus, for each $i \in \mathbb{Z}_+$,

$$\lim_{N\to\infty} \sum_{\eta\in\mathcal{Z}_N(\pi,i)} \rho^\eta = \mathrm{Prob}(\tau(x, w, \pi) = i). \tag{5.16}$$

Moreover, $t_N \to \infty$ as $N \to \infty$. Consequently, (5.15) and (5.16) imply that

$$\begin{aligned} \lim_{N\to\infty} \bar{\tau}_N(x, w, \pi) &= \sum_{i=1}^{\infty} i\mathrm{Prob}(\tau(x, w, \pi) = i) \\ &= \bar{\tau}(x, w, \pi). \end{aligned} \tag{5.17}$$

□

Theorem 5.1. *Suppose a solution to the stochastic DCOC problem* (5.2) *exists for all $x \in G_0$ and $w \in W$. Then, for each $x \in G_0$, $w \in W$, and $\varepsilon > 0$, there exists $\bar{N} > 0$ such that*

$$\bar{\tau}(x, w, \pi_N^*) + \varepsilon \geq \max_{\pi\in\Pi} \bar{\tau}(x, w, \pi), \tag{5.18}$$

where $\pi_N^* \in \arg\max_{\pi_N \in \Pi} \bar{\tau}_N(x, w, \pi_N)$*, for all* $N \geq \bar{N}$.

Proof. For a given initial $x \in G_0$ and $w \in W$, let $\mathcal{T}_N$ be the scenario tree for a given $N \in \mathbb{Z}_+$. Moreover, let $\pi^* \in \Pi$ be a solution to the stochastic DCOC problem (5.2), which exists by assumption, and let $\pi_N^* \in \Pi$ be a control policy that maximizes the average first exit-time associated with $\mathcal{T}_N$ according to (5.10), which exists due to the existence of a solution to (5.2). It follows that

$$\bar{\tau}_N(x, w, \pi_N^*) \geq \bar{\tau}_N(x, w, \pi^*). \tag{5.19}$$

The optimal average first exit-time of the DCOC problem may be written as follows

$$\bar{\tau}(x, w, \pi^*) = \bar{\tau}_N(x, w, \pi^*) + \bar{\tau}_{\mathrm{Rest},N}(x, w, \pi^*), \tag{5.20}$$

where $\bar{\tau}_{\mathrm{Rest},N}$ is the average first exit-time of all scenarios not described by $\mathcal{T}_N$. By Lemma 5.1, $\bar{\tau}_N(x, w, \pi^*)$ approaches $\bar{\tau}(x, w, \pi^*)$ as $N \to \infty$ and thus $\bar{\tau}_{\mathrm{Rest},N} \to 0$. This implies that for every $\varepsilon > 0$, there exists $\bar{N} > 0$ such that

$$\bar{\tau}(x, w, \pi^*) \leq \bar{\tau}_N(x, w, \pi^*) + \varepsilon, \tag{5.21}$$

for all $N \geq \bar{N}$. It follows from (5.19) and (5.21) that

$$\bar{\tau}_N(x, w, \pi_N^*) + \varepsilon \geq \bar{\tau}(x, w, \pi^*), \tag{5.22}$$

for all $N \geq \bar{N}$. In analogy to (5.20), it follows from adding $\bar{\tau}_{\mathrm{Rest},N}(x, w, \pi_N^*)$ to (5.22) that

$$\bar{\tau}(x, w, \pi_N^*) + \varepsilon \geq \bar{\tau}(x, w, \pi^*), \tag{5.23}$$

for all $N \geq \bar{N}$, which proves (5.18). □

5.3 MILP Formulation

In this section, an MILP is proposed that solves (5.10), where, by Theorem 5.1, the average first exit-time of a solution to (5.10) is arbitrarily close to the average first exit-time of a solution to the stochastic DCOC problem (5.2) for a sufficiently large N.

In what follows, a set of control inputs for a given tree $\mathcal{T}_N$ is denoted by

$$\mathcal{U}_N = \{u^\eta \in U_{t^\eta} \,:\, \eta \in \mathcal{T}_N \setminus \mathcal{S}_N\}. \tag{5.24}$$

Moreover, a given $\mathcal{U}_N$ defines a control policy $\pi_{\mathcal{U}_N}$ according to

$$\pi_{\mathcal{U}_N}(x^\eta, w^\eta, t^\eta) = u^\eta \in \mathcal{U}_N, \tag{5.25}$$

for each $\eta \in \mathcal{T}_N \setminus \mathcal{S}_N$ and x^η satisfying (5.5) where $u^{\mathrm{pre}(\eta)} \in \mathcal{U}_N$. Likewise, a control policy $\pi_N^* \in \Pi$ defines a set of control inputs for a given tree $\mathcal{T}_N$ by

$$\mathcal{U}_N(\pi_N^*) = \{u^\eta = \pi_N^*(x^\eta, w^\eta, t^\eta) \,:\, \eta \in \mathcal{T}_N \setminus \mathcal{S}_N\}, \tag{5.26}$$

where x^η satisfies (5.5) for $u^{\mathrm{pre}(\eta)} \in \mathcal{U}_N(\pi_N^*)$.

Using (5.4) [see (5.27e)], (5.5) [see (5.27b)], and (5.24) [see (5.27a)], the MILP for a given tree $\mathcal{T}_N$ is as follows

$$\min_{\mathcal{U}_N, \mathcal{D}_N} \sum_{\eta \in \mathcal{T}_N} \sum_{\xi \in \mathcal{K}_\eta} \delta^\eta \rho^\xi \quad \text{s.t.} \tag{5.27a}$$

$$x^\eta = A_{t^{\mathrm{pre}(\eta)}} x^{\mathrm{pre}(\eta)} + B_{t^{\mathrm{pre}(\eta)}} u^{\mathrm{pre}(\eta)} + w^{\mathrm{pre}(\eta)},\ \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.27b}$$

$$\delta^\eta \geq \delta^{\mathrm{pre}(\eta)},\ \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.27c}$$

$$\delta^\eta \in \{0, 1\} \subset \mathbb{Z},\ \forall \eta \in \mathcal{T}_N \tag{5.27d}$$

$$C_{t^\eta} x^\eta \leq b_{t^\eta} + \mathbf{1} M \delta^\eta,\ \forall \eta \in \mathcal{T}_N, \tag{5.27e}$$

where

$$\mathcal{D}_N = \{\delta^\eta \in \{0, 1\} \,:\, \eta \in \mathcal{T}_N\}, \tag{5.28}$$

denotes a set of δ^η values for tree $\mathcal{T}_N$. Moreover, M is a large positive number, $\mathbf{1}$ denotes the n-dimensional row vector of ones, and the control constraints $u^\eta \in U_{t^\eta}$ for all $\eta \in \mathcal{T}_N \setminus \mathcal{S}_N$ are satisfied due to (5.24). The next result states conditions for the existence of a solution to MILP (5.27).

Lemma 5.2. *For a given $\mathcal{T}_N$, $N \in \mathbb{Z}_+$, suppose $M > 0$ is sufficiently large such that $C_{t^\eta} x^\eta \leq b_{t^\eta} + \mathbf{1}M$ for all $\eta \in \mathcal{T}_N$ and x^η according to* (5.27b) *for any $\mathcal{U}_N$. Then a solution to MILP* (5.27) *exists.*

Proof. Because M is assumed to be sufficiently large, for a given $\mathcal{T}_N$, $N \in \mathbb{Z}_+$, $\delta^\eta = 1$ for all $\eta \in \mathcal{T}_N$ satisfies the constraints of the MILP for any $\mathcal{U}_N$. Since $\delta^\eta \in \{0, 1\}$, the number of possible $\mathcal{D}_N$ is finite. Furthermore, $\rho^\xi \in [0, 1]$ for all $\xi \in \mathcal{T}_N$. Thus, a feasible solution exists for at least one of the $\mathcal{D}_N$ sets and the existence of a solution to MILP (5.27) follows. □

The following theorem shows that, under suitable assumptions and based on (5.25) and

(5.26), a solution to MILP (5.27) is equivalent to a solution to (5.10).

Theorem 5.2. *Suppose Assumption 5.1 holds, a solution to* (5.10) *exists for all* $x \in G_0$ *and* $w \in W$*, and* M *is sufficiently large as in Lemma 5.2. Then* $\mathcal{U}_N^*$ *is a solution to MILP* (5.27) *if the control policy* $\pi_{\mathcal{U}_N^*}$ *according to* (5.25) *is a solution to* (5.10)*. Likewise,* $\pi_N^* \in \Pi$ *is a solution to* (5.10) *if* $\mathcal{U}_N(\pi_N^*)$ *according to* (5.26) *is a solution to MILP* (5.27)*.*

Proof. Let $x = x_0 \in G_0$ and $w = w_0 \in W$ be a given initial condition and $\mathcal{T}_N$ be the corresponding scenario tree, $N \in \mathbb{Z}_+$. For the first part of the proof, suppose π_N^* is a solution to (5.10). Thus,

$$\bar{\tau}_N(x, w, \pi_N^*) \geq \bar{\tau}_N(x, w, \pi_N^\#), \tag{5.29}$$

for all $\pi_N^\# \in \Pi$. A solution to MILP (5.27) exists due to the assumptions and Lemma 5.2. Using (5.26), fix $\mathcal{U}_N = \mathcal{U}_N(\pi_N^*)$ in MILP (5.27) and denote the resulting $\mathcal{D}_N$ by $\mathcal{D}_N^* = \{\delta^{\eta *} \in \{0,1\} : \eta \in \mathcal{T}_N\}$. Similarly, let $\mathcal{D}_N^\# = \{\delta^{\eta \#} \in \{0,1\} : \eta \in \mathcal{T}_N\}$ denote the MILP solution when $\mathcal{U}_N = \mathcal{U}_N(\pi_N^\#)$ is fixed. Hence, by (5.27c)–(5.27e), for each $\eta \in \mathcal{S}_N$, $\delta^{\xi *} = 1$ iff $t^\xi \geq \tau_N^\eta(x, w, \pi_N^*)$, $\delta^{\xi \#} = 1$ iff $t^\xi \geq \tau_N^\eta(x, w, \pi_N^\#)$, $\delta^{\xi *} = 0$ iff $t^\xi < \tau_N^\eta(x, w, \pi_N^*)$, and $\delta^{\xi \#} = 0$ iff $t^\xi < \tau_N^\eta(x, w, \pi_N^\#)$ for all $\xi \in \mathcal{H}_N^\eta$. Consequently, according to (5.8), it follows that

$$\tau_N^\eta(x, w, \pi_N^*) = t^\eta + 1 - \sum_{\xi \in \mathcal{H}_N^\eta} \delta^{\xi *}, \tag{5.30a}$$

$$\tau_N^\eta(x, w, \pi_N^\#) = t^\eta + 1 - \sum_{\xi \in \mathcal{H}_N^\eta} \delta^{\xi \#}, \tag{5.30b}$$

for all $\eta \in \mathcal{S}_N$. Then, using (5.9), (5.29), and (5.30), one obtains

$$\begin{aligned} \sum_{\eta \in \mathcal{S}_N} (t^\eta + 1 - \sum_{\xi \in \mathcal{H}_N^\eta} \delta^{\xi *}) \rho^\eta &= \bar{\tau}_N(x, w, \pi_N^*) \\ &\geq \bar{\tau}_N(x, w, \pi_N^\#) = \sum_{\eta \in \mathcal{S}_N} (t^\eta + 1 - \sum_{\xi \in \mathcal{H}_N^\eta} \delta^{\xi \#}) \rho^\eta. \end{aligned} \tag{5.31}$$

Consequently,

$$\sum_{\eta \in \mathcal{S}_N} \sum_{\xi \in \mathcal{H}_N^\eta} \delta^{\xi *} \rho^\eta \leq \sum_{\eta \in \mathcal{S}_N} \sum_{\xi \in \mathcal{H}_N^\eta} \delta^{\xi \#} \rho^\eta. \tag{5.32}$$

By (5.11) and (5.12), $\eta \in \mathcal{S}_N$ and $\xi \in \mathcal{H}_N^\eta$ iff $\xi \in \mathcal{T}_N$ and $\eta \in \mathcal{K}_N^\xi$. Therefore, (5.32) is equivalent to

$$\sum_{\xi \in \mathcal{T}_N} \sum_{\eta \in \mathcal{K}_N^\xi} \delta^{\xi *} \rho^\eta \leq \sum_{\xi \in \mathcal{T}_N} \sum_{\eta \in \mathcal{K}_N^\xi} \delta^{\xi \#} \rho^\eta, \tag{5.33}$$

which shows that $\mathcal{U}_N(\pi_N^*), \mathcal{D}_N^*$ is a solution to MILP (5.27). This completes the first part of the proof.

For the second part of the proof, let $\mathcal{U}_N^*, \mathcal{D}_N^*$ be a solution to MILP (5.27), which exists by Lemma 5.2, where $\mathcal{D}_N^* = \{\delta^{\eta *} \in \{0,1\} \,:\, \eta \in \mathcal{T}_N\}$. Hence,

$$\sum_{\eta \in \mathcal{T}_N} \sum_{\xi \in \mathcal{K}_N^\eta} \delta^{\eta *} \rho^\xi \leq \sum_{\eta \in \mathcal{T}_N} \sum_{\xi \in \mathcal{K}_N^\eta} \delta^{\eta \#} \rho^\xi, \tag{5.34}$$

for any $\mathcal{U}_N = \mathcal{U}_N^\#$ fixed in MILP (5.27) with corresponding solution $\mathcal{D}_N^\# = \{\delta^{\eta \#} \in \{0,1\} \,:\, \eta \in \mathcal{T}_N\}$. Now define $\pi_{\mathcal{U}_N^*}$ according to (5.25). Since the dynamics in (5.1) and (5.27b) are equivalent, it follows from (5.8) and (5.27c)–(5.27e) that, for each $\eta \in \mathcal{S}_N$,

$$\begin{aligned} \tau_N^\eta(x, w, \pi_{\mathcal{U}_N^*}) &= \min\{\min\{t^\xi \in \mathbb{Z}_{[0,t^\eta]} : \delta^{\xi *} = 1, \xi \in \mathcal{H}_N^\eta\} \cup \{t^\eta + 1\}\} \\ &= t^\eta + 1 - \sum_{\xi \in \mathcal{H}_N^\eta} \delta^{\xi *}. \end{aligned} \tag{5.35}$$

Thus, by (5.9), the average first exit-time of tree $\mathcal{T}_N$ with control policy $\pi_{\mathcal{U}_N^*}$ is given by

$$\bar{\tau}_N(x, w, \pi_{\mathcal{U}_N^*}) = \sum_{\eta \in \mathcal{S}_N} (t^\eta + 1 - \sum_{\xi \in \mathcal{H}_N^\eta} \delta^{\xi *}) \rho^\eta. \tag{5.36}$$

In analogy, define $\pi_{\mathcal{U}_N^\#}$ according to (5.25). Hence,

$$\bar{\tau}_N(x, w, \pi_{\mathcal{U}_N^\#}) = \sum_{\eta \in \mathcal{S}_N} (t^\eta + 1 - \sum_{\xi \in \mathcal{H}_N^\eta} \delta^{\xi \#}) \rho^\eta. \tag{5.37}$$

Using (5.11) and (5.12), it follows from (5.34), (5.36), and (5.37) that

$$\begin{aligned} \bar{\tau}_N(x, w, \pi_{\mathcal{U}_N^*}) - \bar{\tau}_N(x, w, \pi_{\mathcal{U}_N^\#}) &= \sum_{\eta \in \mathcal{S}_N} \sum_{\xi \in \mathcal{H}_N^\eta} (\delta^{\xi \#} - \delta^{\xi *}) \rho^\eta \\ &= \sum_{\xi \in \mathcal{T}_N} \sum_{\eta \in \mathcal{K}_N^\xi} (\delta^{\xi \#} - \delta^{\xi *}) \rho^\eta \geq 0, \end{aligned} \tag{5.38}$$

implying that $\pi_{\mathcal{U}_N^*}$ is a solution to (5.10). □

5.4 SMPC Strategy

5.4.1 Theoretical Results

For a given scenario tree $\mathcal{T}_N$ with initial $w \in W$ and root node $w^{\eta_0} = w$, the control policy $\pi_{\mathcal{U}_N^*}$, derived from the MILP solution $\mathcal{U}_N^*$ according to (5.25), maximizes the average first exit-time $\bar{\tau}_N$ for a given $\mathcal{T}_N$ (Theorem 5.2) and achieves average first exit-times $\bar{\tau}$ arbitrarily close to the optimal value of the stochastic DCOC problem (5.2) for sufficiently large N (Theorem 5.1). However, $\pi_{\mathcal{U}_N^*}$ is only defined for the disturbance scenarios encoded by tree $\mathcal{T}_N$, which are the most likely scenarios for the specified N according to Algorithm 5.1. Thus, starting at $w_0 = w$, $w_t \notin \{w^\eta : \eta \in \mathcal{T}_N, t^\eta = t\}$ may occur at some $t \in \mathbb{Z}_+$, i.e., a disturbance scenario may occur that is not included in $\mathcal{T}_N$.

Therefore, an SMPC scheme is proposed using MILP (5.27), where the solution of the MILP is recomputed at each time instant for an updated tree $\mathcal{T}_N$ based on the current state vector. This approach furthermore provides feedback to compensate for unmodeled effects and can be effective in the context of controlling a nonlinear system and/or when the exact disturbance model is unknown. In this case, the stochastic linear model in (5.1) and the Markov chain for w_t serve as an approximation of the nonlinear system and/or the unknown disturbance model.

For a given $x \in G_{t_0}$, $w \in W$, and $t_0 \in \mathbb{Z}_{\geq 0}$, the SMPC scheme defines the following control policy $\pi_{\mathrm{SMPC},N} \in \Pi$,

$$\pi_{\mathrm{SMPC},N}(x, w, t_0) = u^{\eta_0} \in \mathcal{U}_N^*, \tag{5.39}$$

where $\mathcal{U}_N^*$ is a solution to MILP (5.27) for the scenario tree $\mathcal{T}_N$ with root node η_0 and $t^{\eta_0} \leftarrow t_0$, $x^{\eta_0} \leftarrow x$, and $w^{\eta_0} \leftarrow w$ in Steps 4–6 of Algorithm 5.1.

It follows from Theorems 5.1 and 5.2 that, in terms of first exit-time performance, $\pi_{\mathrm{SMPC},N}$ in (5.39) is arbitrarily close to a solution (assuming one exists) of the stochastic DCOC problem (5.2) for sufficienlty large N. This is summarized in Theorem 5.3.

Theorem 5.3. *Suppose Assumption 5.1 holds, $\pi_{\mathrm{SMPC},N}$ is as in* (5.39)*, M is sufficiently large as in Lemma 5.2, and a solution to* (5.2) *exists for all $x \in G_0$ and $w \in W$. Then, for each $x \in G_0$, $w \in W$, and $\varepsilon > 0$, there exists $\bar{N} > 0$ such that*

$$\bar{\tau}(x, w, \pi_{\mathrm{SMPC},N}) + \varepsilon \geq \max_{\pi \in \Pi} \bar{\tau}(x, w, \pi), \tag{5.40}$$

for all $N \geq \bar{N}$.

Proof. The proof follows from the proofs of Theorems 5.1 and 5.2. □

5.4.2 Implementation

Algorithm 5.2 SMPC implementation

1: $t \leftarrow 0$
2: $x \leftarrow$ obtain current $x(t)$
3: $w \leftarrow$ obtain current $w(t)$
4: $\mathcal{T}_N \leftarrow$ output of Algorithm 5.1 with $t^{\eta_0} \leftarrow t$, $x^{\eta_0} \leftarrow x$, and $w^{\eta_0} \leftarrow w$ in Steps 4–6
5: $t_{\text{comp}} \leftarrow 0$
6: **while** computing solution of MILP (5.27) **do**
7: **if** $t_{\text{comp}} > t_{\max}$ **then**
8: go to Step 13
9: **end if**
10: $t_{\text{comp}} \leftarrow$ update t_{comp}
11: **end while**
12: $\mathcal{U}_N^* \leftarrow$ solution of MILP (5.27); go to Step 14
13: $\mathcal{U}_N^* \leftarrow$ solution of LP (5.41)
14: $u(t) \leftarrow$ apply root node control $u^{\eta_0} \in \mathcal{U}_N^*$ to the system
15: $t \leftarrow t + 1$
16: go to Step 2

In practice, the SMPC strategy may be implemented as in Algorithm 5.2. At each time instant t, the current state vector and disturbance are obtained in Steps 2 and 3 of Algorithm 5.2. Based on these values, a new scenario tree is constructed in Step 4 using Algorithm 5.1. Then a solution $\mathcal{U}_N^*$ of MILP (5.27) is computed. Since MILP is NP-complete [75–77] and computing a solution may take considerably long in the worst-case, an upper bound $t_{\max}$ on the MILP computation time is specified. If the computation time t_{comp} is greater than $t_{\max}$, computation of an MILP solution is terminated (Steps 7–9) and a relaxed version of the MILP, a standard LP, is solved instead. The LP for a given tree $\mathcal{T}_N$ is obtained by replacing the integer variables δ^η in MILP (5.27) by non-negative real variables ε^η for all $\eta \in \mathcal{T}_N$. Thus, the LP is as follows

$$\min_{\mathcal{U}_N, \mathcal{E}_N} \sum_{\eta \in \mathcal{T}_N} \sum_{\xi \in \mathcal{K}_N^\eta} \varepsilon^\eta \rho^\xi \quad \text{s.t.} \tag{5.41a}$$

$$x^\eta = A_{t^{\text{pre}(\eta)}} x^{\text{pre}(\eta)} + B_{t^{\text{pre}(\eta)}} u^{\text{pre}(\eta)} + w^{\text{pre}(\eta)},\ \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.41b}$$

$$\varepsilon^\eta \geq \varepsilon^{\text{pre}(\eta)} \geq 0,\ \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.41c}$$

$$C_{t^\eta} x^\eta \leq b_{t^\eta} + \mathbf{1}\varepsilon^\eta,\ \forall \eta \in \mathcal{T}_N, \tag{5.41d}$$

where $\mathcal{U}_N$ is as in (5.24) and a set of ε^η values for a tree $\mathcal{T}_N$ is denoted by

$$\mathcal{E}_N = \{\varepsilon^\eta \geq 0 \,:\, \eta \in \mathcal{T}_N\}. \tag{5.42}$$

Note that a solution to LP (5.41) always exists because $\varepsilon^\eta \geq 0$ can always be chosen sufficiently large such that (5.41d) is satisfied for all $\eta \in \mathcal{T}_N$.

Remark 5.1. *In contrast to the deterministic case in Chapter 3 where the solution of the LP (i.e., the relaxed version of the MILP) provides good quality suboptimal solutions of the DCOC problem for proper choices of the time horizon (i.e., the time horizon being close to the optimal first exit-time), this may not hold here. This is because, for a given tree $\mathcal{T}_N$, there are different scenarios $\{w_t\}^\eta$, $\eta \in \mathcal{S}_N$, with different time horizons t^η. Thus, $\mathcal{T}_N$ may include scenarios with time horizons much greater than the associated optimal first-exit time, i.e., $t^\eta \gg \tau_N^\eta(x, w, \pi_N^*)$ for some $\{w_t\}^\eta$, for which the solution of LP* (5.41) *may not be close to the solution of* (5.2) *or* (5.10)*, respectively. Therefore, for the SMPC implementation (Algorithm 5.2), the solution of LP* (5.41) *is only used if computing the MILP solution requires longer than specified ($t_{\text{comp}} > t_{\max}$).*

The root node control input u^{η_0} of the MILP solution $\mathcal{U}_N^*$ (or the LP solution in case $t_{\text{comp}} > t_{\max}$) is applied to the system in Step 14 of Algorithm 5.2 and the procedure is repeated at the next time instant $t + 1$.

Note that similar to the deterministic case and Figure 3.1, the state constraints defined by the sets G_t may be tightened for control computation when there are unmodeled effects in order to avoid premature constraint violation. This is not pursued in this chapter.

5.5 Numerical Case Studies

Numerical case studies of stochastic DCOC problems of the form (5.2) are treated using the SMPC strategy given by Algorithm 5.2. The first case study in Section 5.5.1 considers a second-order linear system and investigates the influence of the number of tree nodes on the solution. In the second case study (Section 5.5.2), the ACC problem that was solved with DP techniques in Section 4.6.3 is solved with the SMPC strategy and results are compared. Likewise, in the third case study in Section 5.5.3, the SMPC strategy serves as a driving policy for an autonomous car and the driving problem on a two-lane road from Section 4.5 is considered. In order to reduce computation times, the scenario trees $\mathcal{T}_N(w^i)$ with $w^{\eta_0} = w^i$ are precomputed in all case studies and stored for each $w^i \in W$ instead of constructing $\mathcal{T}_N$ at each time instant. Hence, $\mathcal{T}_N \leftarrow \mathcal{T}_N(w)$ in Step 4 of Algorithm 5.2.

All computations involving the SMPC strategy are performed in MATLAB 2015a on a laptop with an i5-6300 processor and 8 GB RAM, where the Hybrid Toolbox [78] is used to solve LPs and MILPs.

5.5.1 Influence of Number of Tree Nodes

In this case study, the influence of N on the solution is investigated, where a tree $\mathcal{T}_N$ contains $N+1$ nodes. The following stochastic linear time-varying system is considered

$$\begin{bmatrix} r_{1,t+1} \\ r_{2,t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0.1 \\ -0.1 & 1.2 \end{bmatrix} \begin{bmatrix} r_{1,t} \\ r_{2,t} \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5\sin(t/2) \end{bmatrix} u_t + \begin{bmatrix} 0 \\ w_t \end{bmatrix}, \tag{5.43}$$

where $x = [r_1, r_2]^\top$ denotes the state vector and the control input is $u \in [-1, 1]$. The disturbance w takes values in the set $W = \{-1, 0, 1\}$ with transition probabilities $P_W(w^i|w^j) = [P_{W,\mathrm{Mat}}]_{j,i}$ ($j =$ row number and $i =$ column number) given by the matrix

$$P_{W,\mathrm{Mat}} = \begin{bmatrix} 0 & 0.8 & 0.2 \\ 0.3 & 0.5 & 0.2 \\ 0.35 & 0.4 & 0.25 \end{bmatrix},$$

for each $i, j \in \{1, 2, 3\}$. The constraints for the stochastic DCOC problem (5.2) are given by the set

$$G_t \equiv \{x : -2 \leq r_1 \leq 2, -2 \leq r_2 \leq 2\}.$$

The time limit for computing a solution to the MILP in Algorithm 5.2 is set to $t_{\max} = 10$ sec. The following results are for an initial $x_0 = [0, 0]^\top$ and $w_0 = -1$. Figure 5.2 shows sample trajectories, where the dashed red lines indicate the respective constraints. The average first exit-time $\bar{\tau}$ for 1000 random simulations is plotted against N in Figure 5.3. For comparison, a DP solution with conventional VI applied to a discrete grid of the state space using linear interpolation between the grid points (the set defining the control constraints is discretized as well, using an equidistant grid with 21 points) is shown as a reference in Figure 5.3 (dashed blue line), achieving $\bar{\tau} = 32.41$ sec. This DP solution is obtained for a relatively dense grid of 900000 points, which requires about 1.63 hours to compute the control policy offline when implemented in C on a desktop computer (compiled code). Due to the dense grids (for both G_t and U_t), the DP reference solution is expected to be close to an optimal solution of the stochastic DCOC problem.

In line with Theorem 5.3, it can be seen in Figure 5.3 that the SMPC solution improves with increasing N and approaches the DP reference solution (which is expected to be close

to an optimal solution), where the DP reference is slightly exceeded for $N \geq 500$. Note that the average first exit-time achieved by the SMPC strategy appears to be monotonically non-decreasing when increasing N in this case, which may not hold in general.

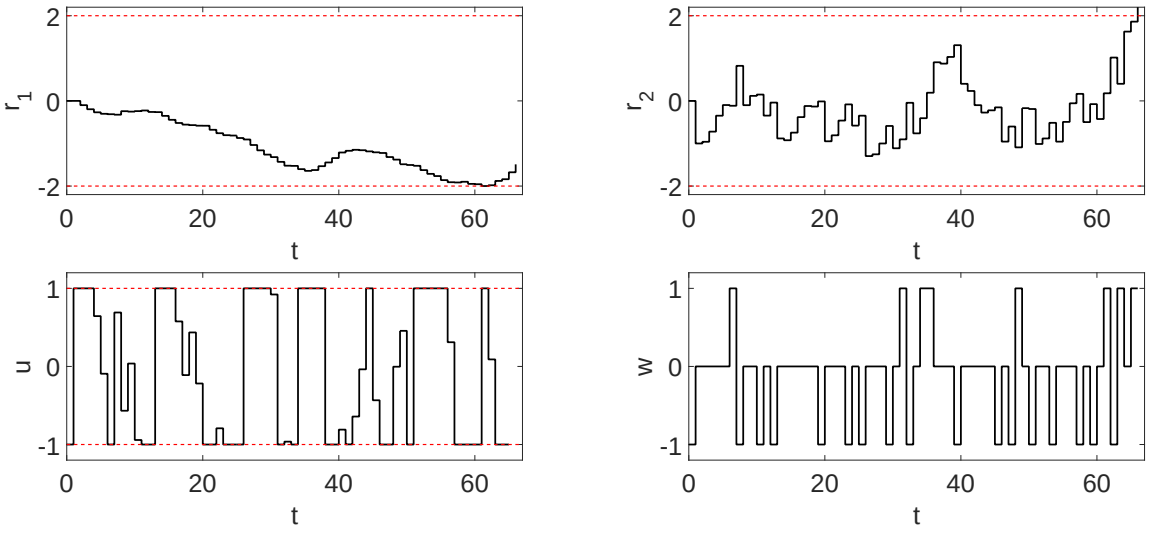

Figure 5.2: Numerical case study on SMPC strategy and influence of number of tree nodes: sample trajectories showing the states r_1 (top left) and r_2 (top right) as well as the control input u (bottom left) and disturbance w (bottom right) vs. t.

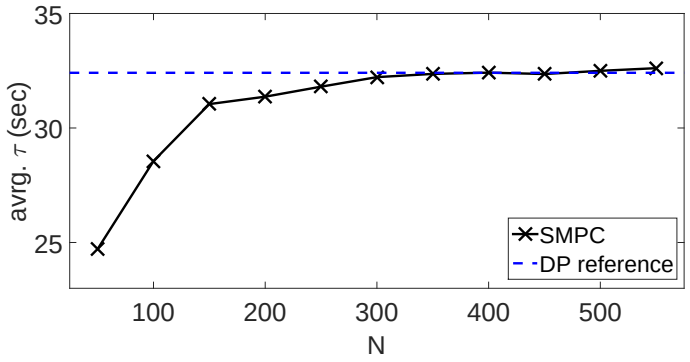

Figure 5.3: Numerical case study on SMPC strategy and influence of number of tree nodes: average first exit-time $\bar{\tau}$ vs. N (1000 random simulations for each N).

The computation time (in MATLAB) of the SMPC scheme for computing the control input at each time instant according to Algorithm 5.2 (Steps 2–14) is shown in Figure 5.4 for different N. The left plot in Figure 5.4 shows the average computation time, which increases nearly exponentially with N. The worst-case / maximum computation time is

shown in Figure 5.4 (right), where the prescribed limit on the MILP computation time $t_{\max} = 10$ sec is reached for $N \geq 400$.

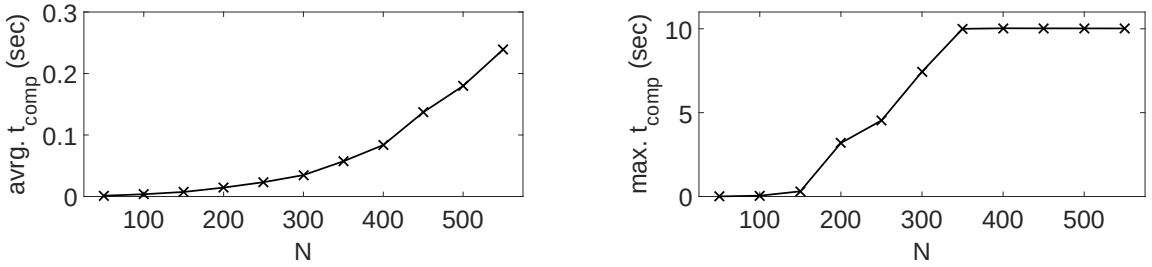

Figure 5.4: Numerical case study on SMPC strategy and influence of number of tree nodes: average (left) and worst-case time (right) to compute control u_t (Steps 2–14 in Algorithm 5.2) vs. N (1000 random simulations for each N).

5.5.2 Adaptive Cruise Control

The same ACC problem (same model, constraints, initial condition, etc.) as in Section 4.6.3 is now solved with the SMPC scheme. The results are compared against the DP-based solution from Section 4.6.3, which achieves an average first exit-time of $\bar{\tau} = 2591$ sec (1000 random simulations). Note that the simulation model is a stochastic hybrid model with state-dependent probabilities for mode switches since there is a 10 % chance of another vehicle cutting in upfront if the time gap T_{g} between the two vehicles is greater than 2.2 sec (see Section 4.6.3). The DP approach in Section 4.6.3 is able to explicitly consider such hybrid models. On the other hand, the SMPC strategy assumes a linear model with an additive random disturbance modeled by a Markov chain and neglects the possibility of another vehicle cutting in upfront. It compensates for the unmodeled effects through feedback (see Algorithm 5.2).

The parameter N is set to 100, meaning that scenario trees with 101 nodes are employed. With this, the SMPC strategy achieves an average first exit-time of $\bar{\tau} = 3120.7$ sec (1000 random simulations), which is an improvement of 20 % compared to the DP solution. The DP solution can be improved by using denser state space discretizations, which, however, would increase computation times exponentially (curse of dimensionality). For the SMPC strategy, 5 msec are required on average to compute the control input at each time instant and a worst-case computation time of 60 msec is recorded.

Similar to the DP solution, see bottom left plot in Figure 4.10, the SMPC strategy generates frequent velocity changes, which may be uncomfortable for passengers and inefficient (wasting fuel/energy). Hence, in order to avoid excessive control input (acceleration/deceleration), control inputs are penalized by considering the weighted sum of $|u^{\eta}|$

values as an additional objective to be minimized. As for example in [92], this is achieved by introducing new variables $\gamma^\eta \geq 0$ for each $\eta \in \mathcal{T}_N \setminus \mathcal{S}_N$. The weighted sum of γ^η values is added to the objective functions of MILP (5.27) and LP (5.41). Moreover, for each $\eta \in \mathcal{T}_N \setminus \mathcal{S}_N$, the constraint $-\gamma^\eta \leq u^\eta \leq \gamma^\eta$ is added to MILP (5.27) and LP (5.41).

In what follows, the factor for weighting the control input penalty is denoted by β_a. Note that for numerical reasons, the probability of each scenario, given by ρ^η for all $\eta \in \mathcal{S}_N$, is normalized by dividing ρ^η by the sum of the probabilities of all scenarios of tree $\mathcal{T}_N$, i.e., $\rho^\eta_{\text{norm}} = \rho^\eta / \sum_{\xi \in \mathcal{S}_N} \rho^\xi$. Instead of ρ^η, ρ^η_{norm} is used in (5.27a) and (5.41a).

β_a	0	0.01	0.05	0.1
$\bar{\tau}$ (sec)	3120.7	1662	1074.1	46.1

Table 5.1: Adaptive cruise control problem, SMPC solution: influence of control input penalty weight β_a on average first exit-time $\bar{\tau}$ (for 1000 random simulations).

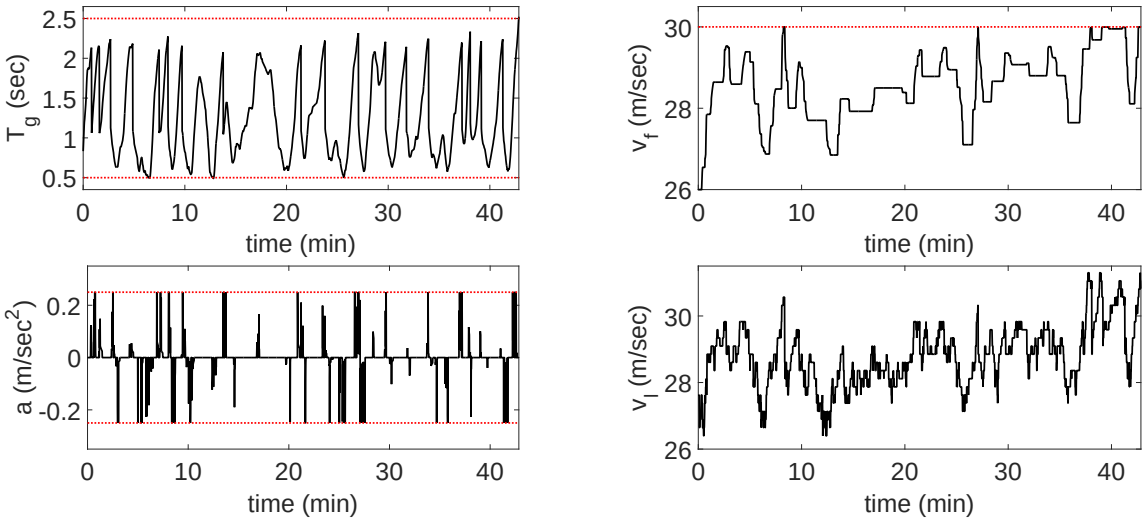

Figure 5.5: Adaptive cruise control problem, SMPC solution with additional penalty on control input (weight $\beta_a = 0.01$): sample trajectories over time of time gap T_{g} between the two vehicles (top left), follower vehicle velocity v_{f} (top right), acceleration of follower vehicle a (bottom left), and lead vehicle velocity v_{l} (bottom right).

Table 5.1 shows the average first exit-time $\bar{\tau}$ for different control input penalty weights β_a. As expected, $\bar{\tau}$ decreases with increasing β_a since a large β_a emphasizes less intense and less frequent acceleration/deceleration. The weight $\beta_a = 0.01$ appears to provide a good balance between a large $\bar{\tau}$ and a smoother velocity profile. Using the SMPC strategy with $\beta_a = 0.01$, Figure 5.5 shows the time history of the time gap T_{g} (top left), the velocity

of the follower vehicle v_f (top right), the control input (bottom left), and the disturbance, i.e., lead vehicle velocity v_l (bottom right) for a random simulation, where the respective constraints are indicated by dotted red lines. Compared to the DP solution (see sample simulation in Figure 4.10) and SMPC with $\beta_a = 0$, the intensity of the control inputs (acceleration/deceleration) is considerably reduced in Figure 5.5 (bottom left), resulting in a smoother trajectory of the follower vehicle velocity v_f.

5.5.3 Driving Policies for Autonomous Vehicles

In this section, the two-lane road driving problem from Section 4.5 is solved with the SMPC strategy. The simulation model (same as in Section 4.5) contains state-dependent transition probabilities, which cannot be considered by the SMPC strategy due to the assumed transition probabilities for w in (4.3). Thus, in order to implement the SMPC strategy, the simulation model is approximated by considering only the most likely scenario $\theta \in \{1, 2, ..., 7\}$ (see Section 4.5.1), while the other scenarios are neglected. In this case, assuming no lane change is performed by the ego car ($l_\mathrm{m} = 0$), the most likely scenario is $\theta = 1$, meaning that the closest car ahead in each lane remains the closest car ahead in the respective lane at the next time instant, i.e., $\mathrm{c}_t \to \mathrm{c}_{t+1}$ and $\mathrm{o}_t \to \mathrm{o}_{t+1}$, because $P_\theta(1|T^i, T^j, 0) \geq P_\theta(q|T^i, T^j, 0)$ for all $q \in \{1, 2, ..., 7\}$ and $T^i, T^j \in \mathcal{T}$ (see Section 4.5.1).

In what follows, the car that is initially ahead of the ego car in its current lane is referred to as car 1 and the car that is initially ahead of the ego car in the other lane is referred to as car 2. The disturbance vector for the linear model in (5.1) is $w = [v_1, v_2]^\top \in W = \mathcal{V} \times \mathcal{V}$ and, with a slight abuse of notation, the following transition probabilities for w are assumed,

$$P_{W,\mathrm{SMPC}}(w^i|w^j) = P_W(v_1^i, v_2^i|v_1^j, v_2^j, 1), \tag{5.44}$$

where P_W is according to (4.54) and $w^i = [v_1^i, v_2^i]^\top \in W$ and $w^j = [v_1^j, v_2^j]^\top \in W$. Note that throughout this section, the parameters of the simulation model, including P_θ, P_W, $\mathcal{T}$, and $\mathcal{V}$, are as in Section 4.5.4.

5.5.3.1 MILP Formulation

MILP (5.27) needs to be extended in order to consider lane changes. Following the developments in [75], this is achieved by introducing the variable $z = [z_1, z_2]^\top \in \mathbb{R}^2$ and the binary variable $l \in \{0, 1\}$. Moreover, the headway to the vehicle in front of the ego car in its current lane is modeled by

$$s_{\mathrm{m},t+1} = s_{1,t+1} - z_{1,t} + z_{2,t}, \tag{5.45}$$

where s_1 denotes the headway to car 1. Likewise, s_2 denotes the headway to car 2. Based on (4.52), s_1 and s_2 are modeled as follows

$$s_{1,t+1} = s_{1,t} + v_{1,t} - v_{\mathrm{m},t} \tag{5.46a}$$

$$s_{2,t+1} = s_{2,t} + v_{2,t} - v_{\mathrm{m},t}, \tag{5.46b}$$

where v_{m} is the velocity of the ego car. As in (4.51), v_{m} evolves according to

$$v_{\mathrm{m},t+1} = v_{\mathrm{m},t} + a_{\mathrm{m},t}, \tag{5.47}$$

where $a_{\mathrm{m},t}$ is the instantaneous acceleration of the ego car.

If $l_t = 0$, the ego car is in the lane of car 1 at the next time instant (and car 1 is the closest car ahead in the ego car's current lane), which requires $z_{1,t} = z_{2,t} = 0$ in (5.45). On the other hand, if $l_t = 1$, the ego car is in the lane of car 2 at the next time instant (and car 2 is the closest car ahead in the ego car's current lane), requiring $z_{1,t} = s_{1,t+1}$ and $z_{2,t} = s_{2,t+1}$ in (5.45). This is encoded by the following inequalities [75]

$$\begin{aligned} z_{1,t} &\le M_1 l_t, & z_{2,t} &\le M_2 l_t \\ z_{1,t} &\ge m_1 l_t, & z_{2,t} &\ge m_2 l_t \\ z_{1,t} &\le s_{1,t+1} - m_1(1-l_t), & z_{2,t} &\le s_{2,t+1} - m_2(1-l_t) \\ z_{1,t} &\ge s_{1,t+1} - M_1(1-l_t), & z_{2,t} &\ge s_{2,t+1} - M_2(1-l_t), \end{aligned} \tag{5.48}$$

where $M_1 \gg s_1$ and $M_2 \gg s_2$ are large numbers and $m_1 \ll s_1$ and $m_2 \ll s_2$ are small numbers. Using (5.45)–(5.48), the extension of MILP (5.27) for the autonomous driving problem is given by MILP (5.51), where, in analogy to (5.24),

$$\mathcal{U}_N = \{u^\eta = [a_{\mathrm{m}}^\eta, l^\eta]^\top \in U_{t^\eta} = \mathcal{A}_{t^\eta} \times \{0,1\} \,:\, \eta \in \mathcal{T}_N \setminus \mathcal{S}_N\}, \tag{5.49}$$

$\Gamma_N = \{\gamma^\eta \ge 0 \,:\, \eta \in \mathcal{T}_N \setminus \mathcal{S}_N\}$, and $\mathcal{D}_N$ is given by (5.28). The acceleration constraints for the ego car are defined by the set

$$\mathcal{A}_t \equiv \mathcal{A} = [a_{\min}, a_{\max}], \tag{5.50}$$

where $a_{\min} = -2.5\,\mathrm{m/sec^2}$ and $a_{\max} = 2.5\,\mathrm{m/sec^2}$ in line with Section 4.5.4.

Note that (5.51m) and the sum of γ^η values in (5.51a) are added to MILP (5.51) to penalize acceleration/deceleration as in Section 5.5.2, where $\beta_a \ge 0$ is the weight factor for the cost in (5.51a).

$$\min_{\mathcal{U}_N, \mathcal{D}_N, \Gamma_N} \sum_{\eta \in \mathcal{T}_N} \sum_{\xi \in \mathcal{K}_N^\eta} \delta^\eta \rho^\xi + \sum_{\eta \in \mathcal{T}_N \setminus \mathcal{S}_N} \beta_a \gamma^\eta \quad \text{s.t.} \tag{5.51a}$$

$$s_1^\eta = s_1^{\mathrm{pre}(\eta)} + v_1^{\mathrm{pre}(\eta)} - v_{\mathrm{m}}^{\mathrm{pre}(\eta)},\ \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.51b}$$

$$s_2^\eta = s_2^{\mathrm{pre}(\eta)} + v_2^{\mathrm{pre}(\eta)} - v_{\mathrm{m}}^{\mathrm{pre}(\eta)},\ \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.51c}$$

$$s_{\mathrm{m}}^\eta = s_1^\eta - z_1^{\mathrm{pre}(\eta)} + z_2^{\mathrm{pre}(\eta)},\ \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.51d}$$

$$v_{\mathrm{m}}^\eta = v_{\mathrm{m}}^{\mathrm{pre}(\eta)} + a_{\mathrm{m}}^{\mathrm{pre}(\eta)},\ \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.51e}$$

$$z_i^\eta \leq M_i l^\eta,\ i \in \{1,2\},\ \forall \eta \in \mathcal{T}_N \setminus \mathcal{S}_N \tag{5.51f}$$

$$z_i^\eta \geq m_i l^\eta,\ i \in \{1,2\},\ \forall \eta \in \mathcal{T}_N \setminus \mathcal{S}_N \tag{5.51g}$$

$$z_i^{\mathrm{pre}(\eta)} \leq s_i^\eta - m_i(1 - l^{\mathrm{pre}(\eta)}),\ i \in \{1,2\},\ \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.51h}$$

$$z_i^{\mathrm{pre}(\eta)} \geq s_i^\eta - M_i(1 - l^{\mathrm{pre}(\eta)}),\ i \in \{1,2\},\ \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.51i}$$

$$\delta^\eta \geq \delta^{\mathrm{pre}(\eta)},\ \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.51j}$$

$$\delta^\eta \in \{0,1\} \subset \mathbb{Z},\ \forall \eta \in \mathcal{T}_N \tag{5.51k}$$

$$C_{t^\eta}[s_{\mathrm{m}}^\eta, v_{\mathrm{m}}^\eta]^\top \leq b_{t^\eta} + \mathbf{1} M \delta^\eta,\ \forall \eta \in \mathcal{T}_N \tag{5.51l}$$

$$-\gamma^\eta \leq a_{\mathrm{m}}^\eta \leq \gamma^\eta,\ \forall \eta \in \mathcal{T}_N \setminus \mathcal{S}_N. \tag{5.51m}$$

The initial values for the root node η_0 are given by the states at the current time instant $t \in \mathbb{Z}_{\geq 0}$: $s_1^{\eta_0} = s_{\mathrm{c},t}$, $s_2^{\eta_0} = s_{\mathrm{o},t}$, $s_{\mathrm{m}}^{\eta_0} = s_{\mathrm{c},t}$, and $v_{\mathrm{m}}^{\eta_0} = v_{\mathrm{m},t}$, where s_{c} and s_{o} denote the headways of the two cars ahead in the ego car's current lane and in the other lane, respectively. The initial disturbance is given by the current velocities of the two cars ahead: $w^{\eta_0} = [v_1^{\eta_0}, v_2^{\eta_0}]^\top = [v_{\mathrm{c},t}, v_{\mathrm{o},t}]^\top$. Based on w^{η_0} and the transition probabilities $P_{W,\mathrm{SMPC}}$ defined by (5.44), the scenario tree $\mathcal{T}_N$ can be built according to Algorithm 5.1.

The implementation of the SMPC strategy for the autonomous driving problem follows from Algorithm 5.2 and is summarized by Algorithm 5.3, where, instead of MILP (5.27), MILP (5.51) is solved in Step 12. At each time instant $t \in \mathbb{Z}_{\geq 0}$, the root node control values $u^{\eta_0} = [a_{\mathrm{m}}^{\eta_0}, l^{\eta_0}]^\top$ of the current MILP solution are applied to the ego car, setting its acceleration and lane change indicator l_{m} at t.

In contrast to MILP (5.27) and LP (5.41), due to the binary variables l^η, MILP (5.51) cannot be transformed into an LP similar to (5.41). Hence, if no MILP solution is found within the prescribed computation time (Steps 6–11 in Algorithm 5.3), $l_{\mathrm{m},t} = 0$ (no lane change is performed at t) and $a_{\mathrm{m},t}$ is obtained by solving the LP of an ACC problem for the ego car's current lane, while neglecting the other lane. In analogy to LP (5.41), the LP of the ACC problem for the ego car's current lane (with additional penalty for acceleration/deceleration) reads

$$\min_{\mathcal{U}_N, \mathcal{E}_N, \Gamma_N} \sum_{\eta \in \mathcal{T}_N} \sum_{\xi \in \mathcal{K}_N^\eta} \varepsilon^\eta \rho^\xi + \sum_{\eta \in \mathcal{T}_N \setminus \mathcal{S}_N} \beta_a \gamma^\eta \quad \text{s.t.} \tag{5.52a}$$

$$s_1^\eta = s_1^{\mathrm{pre}(\eta)} + v_1^{\mathrm{pre}(\eta)} - v_{\mathrm{m}}^{\mathrm{pre}(\eta)},\ \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.52b}$$

$$v_{\mathrm{m}}^\eta = v_{\mathrm{m}}^{\mathrm{pre}(\eta)} + a_{\mathrm{m}}^{\mathrm{pre}(\eta)},\ \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.52c}$$

$$\varepsilon^\eta \geq \varepsilon^{\mathrm{pre}(\eta)} \geq 0,\ \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.52d}$$

$$C_{t^\eta}[s_1^\eta, v_{\mathrm{m}}^\eta]^\top \leq b_{t^\eta} + \mathbf{1}\varepsilon^\eta,\ \forall \eta \in \mathcal{T}_N \tag{5.52e}$$

$$-\gamma^\eta \leq a_{\mathrm{m}}^\eta \leq \gamma^\eta,\ \forall \eta \in \mathcal{T}_N \setminus \mathcal{S}_N. \tag{5.52f}$$

Algorithm 5.3 SMPC implementation for autonomous driving problem

1: $t \leftarrow 0$
2: $[s_1^{\eta_0}, s_2^{\eta_0}, s_{\mathrm{m}}^{\eta_0}, v_{\mathrm{m}}^{\eta_0}]^\top \leftarrow$ obtain current states $[s_{\mathrm{c}}(t), s_{\mathrm{o}}(t), s_{\mathrm{c}}(t), v_{\mathrm{m}}(t)]^\top$
3: $[v_1^{\eta_0}, v_2^{\eta_0}]^\top \leftarrow$ obtain current disturbances $[v_{\mathrm{c}}(t), v_{\mathrm{o}}(t)]^\top$
4: $\mathcal{T}_N \leftarrow$ output of Algorithm 5.1 with $\rho^{\eta_j^{\mathrm{succ}(\eta_i)}} \leftarrow \rho^{\eta_i} P_{W,\mathrm{SMPC}}(w^j | w^{\eta_i})$ [see (5.44)] in Step 12 and $t^{\eta_0} \leftarrow t$, $x^{\eta_0} \leftarrow [s_1^{\eta_0}, s_2^{\eta_0}, s_{\mathrm{m}}^{\eta_0}, v_{\mathrm{m}}^{\eta_0}]^\top$, and $w^{\eta_0} \leftarrow [v_1^{\eta_0}, v_2^{\eta_0}]^\top$ in Steps 4–6
5: $t_{\mathrm{comp}} \leftarrow 0$
6: **while** computing solution of MILP (5.51) **do**
7: **if** $t_{\mathrm{comp}} > t_{\max}$ **then**
8: go to Step 14
9: **end if**
10: $t_{\mathrm{comp}} \leftarrow$ update t_{comp}
11: **end while**
12: $\mathcal{U}_N^* \leftarrow$ solution of MILP (5.51)
13: $[a_{\mathrm{m}}(t), l_{\mathrm{m}}(t)]^\top \leftarrow$ apply root node control $u^{\eta_0} \in \mathcal{U}_N^*$ to the system; go to Step 16
14: $\mathcal{U}_N^* \leftarrow$ solution of LP (5.52)
15: $a_{\mathrm{m}}(t) \leftarrow$ apply root node control $a_{\mathrm{m}}^{\eta_0} \in \mathcal{U}_N^*$ to the system; $l_{\mathrm{m}}(t) \leftarrow 0$
16: $t \leftarrow t + 1$
17: go to Step 2

5.5.3.2 Numerical Results

The DCOC driving problem for a two-lane road from Section 4.5 is now solved with the SMPC scheme given by Algorithm 5.3. The simulation model, parameters, initial condition, etc. are as in the numerical case study in Section 4.5.4. However, the constraint on

the lane change frequency and the corresponding state L [see (4.65)] are not considered because, in contrast to the DP-based policies (without constraints on L) in Section 4.5.4, the SMPC strategy does not generate an excessive amount of lane changes. This may be due to the fact that, unlike the DP-based policies, the SMPC strategy assumes only a subset ($\theta = 1$) of all possible scenarios and the different model assumption leads to control policies with less frequent lane changes. Without the constraint on lane change frequency, the constraints for the stochastic DCOC problem (5.2) in this section are given by the set

$$G_t \equiv G = \{x : T_{\mathrm{g,c}} \geq 0.5\,\mathrm{sec}, v_{\min} \leq v_{\mathrm{m}} \leq v_{\max}\}, \tag{5.53}$$

where $T_{\mathrm{g,c}} = s_{\mathrm{c}}/v_{\mathrm{m}}$ is the time gap to the vehicle in front of the ego car in its current lane and $v_{\min} = 19.7\bar{2}\,\mathrm{m/sec}$ and $v_{\max} = 27.\bar{2}\,\mathrm{m/sec}$. The state constraints are encoded for the SMPC strategy [see (5.51l) and (5.52e)] by

$$C_t \equiv \begin{bmatrix} -1 & 0.5 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \quad b_t \equiv \begin{bmatrix} 0 \\ v_{\max} \\ -v_{\min} \end{bmatrix}. \tag{5.54}$$

The parameters in Algorithm 5.3 are set to $N = 50$ and $t_{\max} = 1$ sec and $\beta_a = 0.01$ is chosen (see Section 5.5.2). Note that that due to the unmodeled effects (only considering a subset of all possible scenarios), Theorem 5.3 may not hold in this case and increasing N may not improve the average first exit-time.

Figure 5.6 shows sample trajectories of the relative time gaps of the vehicles in front of the ego car in its current lane and in the other lane and of the ego car velocity and lane change indicator when using the SMPC strategy. The dashed red lines in Figure 5.6 indicate the prescribed constraints, which are violated for the first time after about 23 min in this case. On average (1000 random simulations), the SMPC strategy achieves a first exit-time of $\bar{\tau} = 912$ sec, which is considerably better than the values achieved by the DP-based policies, see Tables 4.2 and 4.3. Note, however, that the DP-based policies constrain the number of lane changes to at most 1 lane change per 10 sec. Without this hard constraint on the lane change frequency, with the advantage of complete knowledge of the simulation model (i.e., there are no unmodeled effects), the ADP approach (similar results hold for conventional DP) achieves an average first exit-time of $\bar{\tau} = 1703.8$ sec, where, however, a lane change is performed every 2.9 sec on average. In contrast, the SMPC strategy yields an average lane change frequency of 1 lane change per 144.2 sec.

While the DP-based policies (without constraints on the lane change frequency) are able to achieve a better control performance due to taking into account all possible scenarios

of the simulation model, the computational complexity is higher compared to the SMPC scheme. With both conventional DP and ADP, the respective control policy needs to be computed offline, where conventional DP is limited due to the curse of dimensionality and the ADP approach requires extensive tuning of the NN structure and training algorithm until a proper solution is obtained. The SMPC strategy, on the other hand, is based on a simpler model and the control input is computed online by solving either an MILP or a standard LP. On average, computing the control input according to Algorithm 5.3 at each time instant in MATLAB requires 81 msec and the worst case computation time is 1.03 sec (in line with $t_{\max} = 1$ sec), which can be further reduced by reducing $t_{\max}$ and/or N.

The SMPC strategy regulates the ego car velocity to $v_{\min}$ as can be seen in Figure 5.6 (bottom right) and in other simulation samples. On average, this results in large time gaps to the vehicle in front of the ego car (see top left in Figure 5.6), which, however, increases the possibility of another vehicle cutting in upfront. Since the SMPC strategy does not consider such an event, this may reduce the average first exit-time. Therefore, a hybrid SMPC strategy is proposed in the following section (Section 5.5.3.3) that combines MILP (5.51) with two car following controllers (one for each lane).

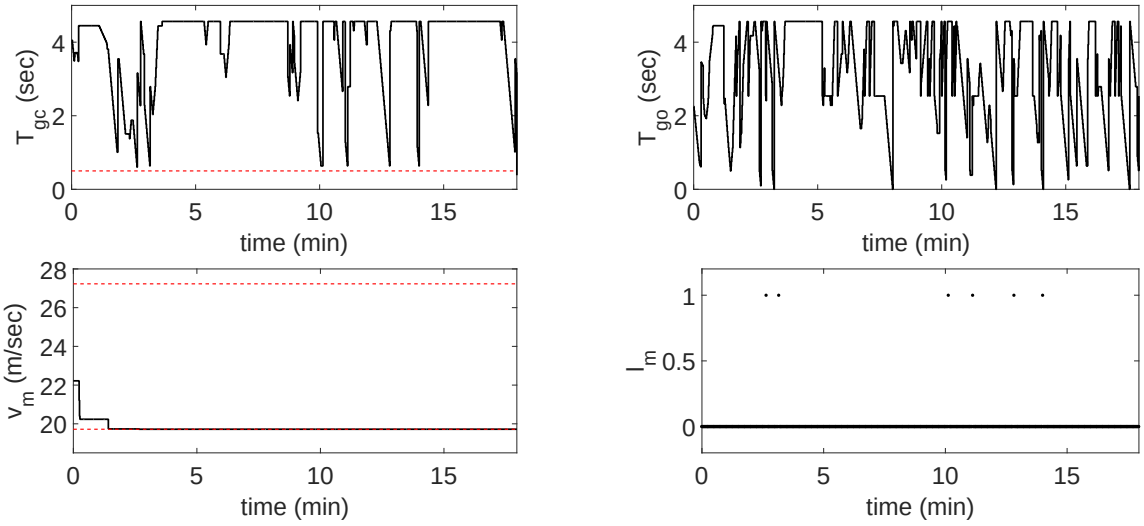

Figure 5.6: SMPC – autonomous driving case study: sample trajectories of relative time gap $T_{\mathrm{g,c}}$ for the ego car's current lane (top left), relative time gap $T_{\mathrm{g,o}}$ for other lane (top right), velocity v_{m} of ego car (bottom left), and lane change indicator l_{m} of ego car (bottom right) over time.

5.5.3.3 Hybrid SMPC Strategy

In order to avoid large time gaps to the vehicle ahead and reduce the chances of another vehicle cutting in upfront, the SMPC strategy from the previous section is combined with

car following problems for each lane. An MILP similar to MILP (5.51) is solved to make lane change decisions and, for each lane, an additional MILP similar to the ACC case in Section 5.5.2 is solved to obtain the acceleration of the ego car. The ACC-MILP for each lane $j \in \{1, 2\}$ follows from MILP (5.27) and is given by

$$\min_{\mathcal{U}_{\mathrm{ACC},N,j}, \mathcal{D}_N, \Gamma_N} \sum_{\eta \in \mathcal{T}_N} \sum_{\xi \in \mathcal{K}_N^\eta} \delta^\eta \rho^\xi + \sum_{\eta \in \mathcal{T}_N \setminus \mathcal{S}_N} \beta_a \gamma^\eta \quad \text{s.t.} \tag{5.55a}$$

$$s_j^\eta = s_j^{\mathrm{pre}(\eta)} + v_j^{\mathrm{pre}(\eta)} - v_{\mathrm{m}}^{\mathrm{pre}(\eta)}, \, \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.55b}$$

$$v_{\mathrm{m}}^\eta = v_{\mathrm{m}}^{\mathrm{pre}(\eta)} + a_{\mathrm{m},j}^{\mathrm{pre}(\eta)}, \, \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.55c}$$

$$\delta^\eta \geq \delta^{\mathrm{pre}(\eta)}, \, \forall \eta \in \mathcal{T}_N \setminus \{\eta_0\} \tag{5.55d}$$

$$\delta^\eta \in \{0, 1\} \subset \mathbb{Z}, \, \forall \eta \in \mathcal{T}_N \tag{5.55e}$$

$$C_{\mathrm{ACC},t^\eta} [s_j^\eta, v_{\mathrm{m}}^\eta]^\top \leq b_{\mathrm{ACC},t^\eta} + \mathbf{1} M \delta^\eta, \, \forall \eta \in \mathcal{T}_N \tag{5.55f}$$

$$-\gamma^\eta \leq a_{\mathrm{m},j}^\eta \leq \gamma^\eta, \, \forall \eta \in \mathcal{T}_N \setminus \mathcal{S}_N, \tag{5.55g}$$

where

$$\mathcal{U}_{\mathrm{ACC},N,j} = \{a_{\mathrm{m},j}^\eta \in \mathcal{A} \, : \, \eta \in \mathcal{T}_N \setminus \mathcal{S}_N\}. \tag{5.56}$$

In this case study, ACC-MILP (5.55) considers the following state constraints,

$$\text{for } j \in \{1, 2\}, \, 0.5\,\mathrm{sec} \leq s_j / v_{\mathrm{m}} \leq 3\,\mathrm{sec} \text{ and } v_{\min} \leq v_{\mathrm{m}} \leq v_{\max}, \tag{5.57}$$

which, according to (5.55f), are encoded by

$$C_{\mathrm{ACC},t} \equiv C_{\mathrm{ACC}} = \begin{bmatrix} -1 & 0.5 \\ 1 & -3 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, \quad b_{\mathrm{ACC},t} \equiv b_{\mathrm{ACC}} = \begin{bmatrix} 0 \\ 0 \\ v_{\max} \\ -v_{\min} \end{bmatrix}. \tag{5.58}$$

After the respective acceleration is obtained for each lane by solving ACC-MILP (5.55) for $j = 1$ and $j = 2$, the MILP for making lane change decisions is solved. Based on the ACC-MILP solutions $\mathcal{U}_{\mathrm{ACC},N,1}$ and $\mathcal{U}_{\mathrm{ACC},N,2}$, the MILP is given by (5.60), where

$$\mathcal{U}_{\mathrm{LC},N} = \{l^\eta \in \{0, 1\} \, : \, \eta \in \mathcal{T}_N \setminus \mathcal{S}_N\}. \tag{5.59}$$

$$\min_{\mathcal{U}_{\mathrm{LC},N},\mathcal{D}_N,\Gamma_N} \sum_{\eta\in\mathcal{T}_N}\sum_{\xi\in\mathcal{K}_N^\eta} \delta^\eta\rho^\xi + \sum_{\eta\in\mathcal{T}_N\setminus\mathcal{S}_N} \beta_l\gamma^\eta \quad \text{s.t.} \tag{5.60a}$$

$$s_1^\eta = s_1^{\mathrm{pre}(\eta)} + v_1^{\mathrm{pre}(\eta)} - v_{\mathrm{m}}^{\mathrm{pre}(\eta)},\ \forall\eta\in\mathcal{T}_N\setminus\{\eta_0\} \tag{5.60b}$$

$$s_2^\eta = s_2^{\mathrm{pre}(\eta)} + v_2^{\mathrm{pre}(\eta)} - v_{\mathrm{m}}^{\mathrm{pre}(\eta)},\ \forall\eta\in\mathcal{T}_N\setminus\{\eta_0\} \tag{5.60c}$$

$$s_{\mathrm{m}}^\eta = s_1^\eta - z_1^{\mathrm{pre}(\eta)} + z_2^{\mathrm{pre}(\eta)},\ \forall\eta\in\mathcal{T}_N\setminus\{\eta_0\} \tag{5.60d}$$

$$v_{\mathrm{m}}^\eta = v_{\mathrm{m}}^{\mathrm{pre}(\eta)} + a_{\mathrm{m},1}^{\mathrm{pre}(\eta)} - y_1^{\mathrm{pre}(\eta)} + y_2^{\mathrm{pre}(\eta)},\ \forall\eta\in\mathcal{T}_N\setminus\{\eta_0\} \tag{5.60e}$$

$$z_i^\eta \le M_i l^\eta,\ i\in\{1,2\},\ \forall\eta\in\mathcal{T}_N\setminus\mathcal{S}_N \tag{5.60f}$$

$$z_i^\eta \ge m_i l^\eta,\ i\in\{1,2\},\ \forall\eta\in\mathcal{T}_N\setminus\mathcal{S}_N \tag{5.60g}$$

$$z_i^{\mathrm{pre}(\eta)} \le s_i^\eta - m_i(1-l^{\mathrm{pre}(\eta)}),\ i\in\{1,2\},\ \forall\eta\in\mathcal{T}_N\setminus\{\eta_0\} \tag{5.60h}$$

$$z_i^{\mathrm{pre}(\eta)} \ge s_i^\eta - M_i(1-l^{\mathrm{pre}(\eta)}),\ i\in\{1,2\},\ \forall\eta\in\mathcal{T}_N\setminus\{\eta_0\} \tag{5.60i}$$

$$y_i^\eta \le a_{\max} l^\eta,\ i\in\{1,2\},\ \forall\eta\in\mathcal{T}_N\setminus\mathcal{S}_N \tag{5.60j}$$

$$y_i^\eta \ge a_{\min} l^\eta,\ i\in\{1,2\},\ \forall\eta\in\mathcal{T}_N\setminus\mathcal{S}_N \tag{5.60k}$$

$$y_i^\eta \le a_{\mathrm{m},i}^\eta - a_{\min}(1-l^{\mathrm{pre}(\eta)}),\ i\in\{1,2\},\ \forall\eta\in\mathcal{T}_N\setminus\mathcal{S}_N \tag{5.60l}$$

$$y_i^\eta \ge a_{\mathrm{m},i}^\eta - a_{\max}(1-l^{\mathrm{pre}(\eta)}),\ i\in\{1,2\},\ \forall\eta\in\mathcal{T}_N\setminus\mathcal{S}_N \tag{5.60m}$$

$$\delta^\eta \ge \delta^{\mathrm{pre}(\eta)},\ \forall\eta\in\mathcal{T}_N\setminus\{\eta_0\} \tag{5.60n}$$

$$\delta^\eta \in \{0,1\}\subset\mathbb{Z},\ \forall\eta\in\mathcal{T}_N \tag{5.60o}$$

$$C_{\mathrm{LC},t^\eta}[s_{\mathrm{m}}^\eta, v_{\mathrm{m}}^\eta]^\top \le b_{\mathrm{LC},t^\eta} + \mathbf{1}M\delta^\eta,\ \forall\eta\in\mathcal{T}_N \tag{5.60p}$$

$$-\gamma^\eta \le l^\eta - l^{\mathrm{pre}(\eta)} \le \gamma^\eta,\ \forall\eta\in\mathcal{T}_N\setminus\{\mathcal{S}_N\cup\{\eta_0\}\} \tag{5.60q}$$

$$l^{\eta_0} \le \gamma^{\eta_0}. \tag{5.60r}$$

Similar to (5.51f)–(5.51i), (5.60j)–(5.60m) are included to use the acceleration that corresponds to the current lane, i.e., use the acceleration given by $\mathcal{U}_{\mathrm{ACC},N,1}$ if the ego car is in car 1's lane ($l = 0$) or use $\mathcal{U}_{\mathrm{ACC},N,2}$ if the ego car is in car 2's lane ($l = 1$). In addition, similar to penalizing excessive acceleration [see (5.51m) and (5.55g)], (5.60q) and (5.60r) are included to penalize excessive lane changes, where $\beta_l \ge 0$ denotes the associated weight in the cost function (5.60a).

The constraints for MILP (5.60) follow from (5.53). Hence, without the constraints on the ego car velocity [which is controlled by ACC-MILP (5.55)], (5.60p) is defined by

$$C_{\mathrm{LC},t} \equiv \begin{bmatrix} -1 & 0.5 \end{bmatrix},\ \ b_{\mathrm{LC},t} \equiv 0. \tag{5.61}$$

If the respective car ahead of the ego car is further away than the prescribed limit on the

time gap for the car following controller (which, according to (5.57), is 3 sec in this case study), the acceleration is undefined. A simple proportional controller is implemented to set the acceleration in this case. The proportional controller regulates the ego car velocity to a prescribed cruise speed v_{cruise} and is as follows

$$p_v(v_{\text{m}}, v_{\text{cruise}}) = \begin{cases} a_{\min}, & \text{if } K_p(v_{\text{cruise}} - v_{\text{m}}) < a_{\min}, \\ a_{\max}, & \text{if } K_p(v_{\text{cruise}} - v_{\text{m}}) > a_{\max}, \\ K_p(v_{\text{cruise}} - v_{\text{m}}), & \text{otherwise}, \end{cases} \tag{5.62}$$

where K_p is set to 0.25 in this case study.

Algorithm 5.4 defines the hybrid SMPC scheme. At each time instant, the current states and disturbances are obtained. Based on these values, the scenario tree $\mathcal{T}_N$ is generated in Step 4. Then the accelerations for each lane $j \in \{1, 2\}$ are computed in Steps 5–16, depending on the respective current time gap $s_j^{\eta_0}/v_{\text{m}}^{\eta_0}$, either by solving ACC-MILP (5.55) or by using the proportional controller in (5.62). Based on the accelerations for each lane, the solution of MILP (5.60) is calculated to make a lane change decision in Step 26 and the acceleration associated with the chosen lane is applied to the system (Steps 27–31). If computing a solution to MILP (5.60) [Steps 18–24] takes longer than specified, no lane change is performed at the current time instant and the acceleration for the current lane ($j = 1$) is applied to the system. The procedure is repeated at the next time instant.

For the following simulations, the parameters are set to $N = 35$ and $t_{\max} = 1$ sec. The weights on acceleration and lane changing are chosen as $\beta_a = 0.01$ and $\beta_l = 1$, respectively, as this setting appears to generate a reasonable level of control inputs (i.e., no excessive acceleration and lane changing) in this specific case study.

The hybrid SMPC strategy (Algorithm 5.4) is applied to the same DCOC driving problem as in Sections 4.5.4 and 5.5.3.2 (same simulation model, parameters, initial condition, etc.). Figure 5.7 shows trajectories for a random simulation when $v_{\text{cruise}} = v_{\min}$. It can be seen that the ego car velocity is increased several times when required by the car following controller. On average, this reduces the time gap to the vehicle in front and reduces the risk of another vehicle cutting in upfront. Compared to the SMPC strategy defined by Algorithm 5.3, the result is a larger average first exit-time, which is $\bar{\tau} = 1104.9$ sec (1000 random simulations) in this case. Note that this value is also significantly larger than the results achieved by the DP-based policies in Section 4.5.4, which, however, are subject to hard constraints on the lane change frequency and can be improved by allowing larger lane change frequencies, see Figure 4.5, or by removing the constraint on lane changes.

Algorithm 5.4 Hybrid SMPC strategy for autonomous driving problem

1: $t \leftarrow 0$
2: $[s_1^{\eta_0}, s_2^{\eta_0}, s_{\mathrm{m}}^{\eta_0}, v_{\mathrm{m}}^{\eta_0}]^\top \leftarrow$ obtain current states $[s_{\mathrm{c}}(t), s_{\mathrm{o}}(t), s_{\mathrm{c}}(t), v_{\mathrm{m}}(t)]^\top$
3: $[v_1^{\eta_0}, v_2^{\eta_0}]^\top \leftarrow$ obtain current disturbances $[v_{\mathrm{c}}(t), v_{\mathrm{o}}(t)]^\top$
4: $\mathcal{T}_N \leftarrow$ output of Algorithm 5.1 with $\rho^{\eta_j^{\mathrm{succ}(\eta_i)}} \leftarrow \rho^{\eta_i} P_{W,\mathrm{SMPC}}(w^j|w^{\eta_i})$ [see (5.44)] in Step 12 and $t^{\eta_0} \leftarrow t$, $x^{\eta_0} \leftarrow [s_1^{\eta_0}, s_2^{\eta_0}, s_{\mathrm{m}}^{\eta_0}, v_{\mathrm{m}}^{\eta_0}]^\top$, and $w^{\eta_0} \leftarrow [v_1^{\eta_0}, v_2^{\eta_0}]^\top$ in Steps 4–6
5: **for** $j \in \{1, 2\}$ **do**
6: **if** $s_j^{\eta_0}/v_{\mathrm{m}}^{\eta_0} > 3$ sec **then**
7: $\mathcal{U}^*_{\mathrm{ACC},N,j} \leftarrow \emptyset$
8: **for** $\eta \in \mathcal{T}_N \setminus \mathcal{S}_N$ **do**
9: $a^\eta_{\mathrm{m},j} \leftarrow p_v(v^\eta_{\mathrm{m}}, v_{\mathrm{cruise}})$, see (5.62)
10: $v^\xi_{\mathrm{m}} \leftarrow v^\eta_{\mathrm{m}} + a^\eta_{\mathrm{m},j}$, for all $\xi \in \mathrm{succ}(\eta) \cap \mathcal{T}_N$
11: $\mathcal{U}^*_{\mathrm{ACC},N,j} \leftarrow \mathcal{U}^*_{\mathrm{ACC},N,j} \cup \{a^\eta_{\mathrm{m},j}\}$
12: **end for**
13: **else**
14: $\mathcal{U}^*_{\mathrm{ACC},N,j} \leftarrow$ solution of ACC-MILP (5.55)
15: **end if**
16: **end for**
17: $t_{\mathrm{comp}} \leftarrow 0$
18: **while** computing solution of MILP (5.60) **do**
19: **if** $t_{\mathrm{comp}} > t_{\max}$ **then**
20: $a_{\mathrm{m}}(t) \leftarrow$ apply root node control $a^{\eta_0}_{\mathrm{m},1} \in \mathcal{U}^*_{\mathrm{ACC},N,1}$ to the system
21: $l_{\mathrm{m}}(t) \leftarrow 0$; go to Step 32
22: **end if**
23: $t_{\mathrm{comp}} \leftarrow$ update t_{comp}
24: **end while**
25: $\mathcal{U}^*_{\mathrm{LC},N} \leftarrow$ solution of MILP (5.60)
26: $l_{\mathrm{m}}(t) \leftarrow$ apply root node control $l^{\eta_0} \in \mathcal{U}^*_{\mathrm{LC},N}$ to the system
27: **if** $l_{\mathrm{m}}(t) = 0$ **then**
28: $a_{\mathrm{m}}(t) \leftarrow$ apply root node control $a^{\eta_0}_{\mathrm{m},1} \in \mathcal{U}^*_{\mathrm{ACC},N,1}$ to the system
29: **else**
30: $a_{\mathrm{m}}(t) \leftarrow$ apply root node control $a^{\eta_0}_{\mathrm{m},2} \in \mathcal{U}^*_{\mathrm{ACC},N,2}$ to the system
31: **end if**
32: $t \leftarrow t + 1$; go to Step 2

Besides better average first exit-times, the reduced average time gap, achieved by the

additional car following controller, increases the number of lane changes as can be seen by comparing the bottom right plots in Figures 5.6 and 5.7 (see also Figure 5.9). The number of lane changes can be reduced (increased) by increasing (decreasing) the parameter β_l, where, for $v_{\text{cruise}} = v_{\min}$ and $\beta_l = 1$, the hybrid SMPC scheme yields an average lane change frequency of 1 lane change per 60.3 sec.

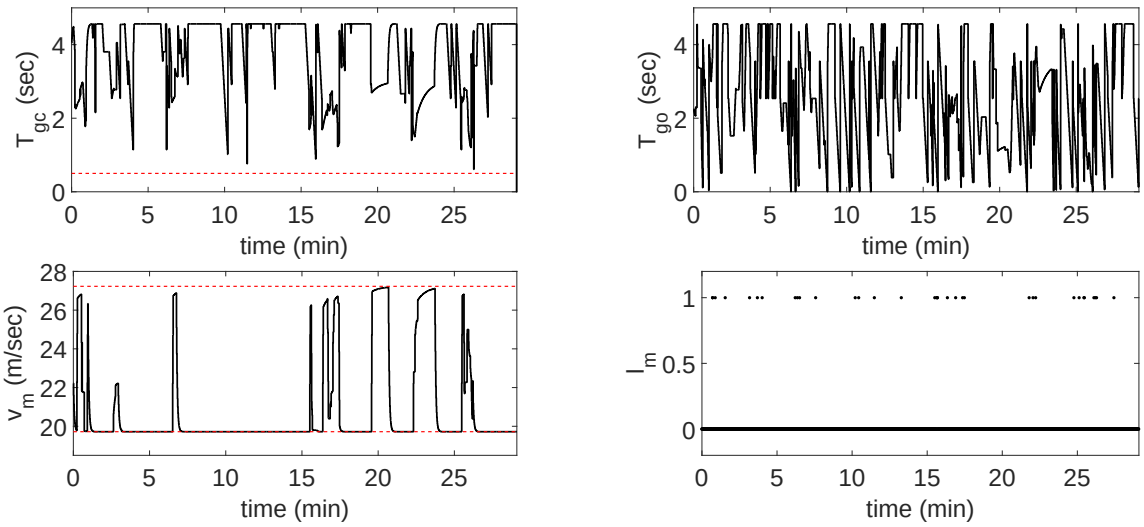

Figure 5.7: Hybrid SMPC with $v_{\text{cruise}} = v_{\min}$ – autonomous driving case study: sample trajectories of relative time gap $T_{\text{g,c}}$ for the ego car's current lane (top left), relative time gap $T_{\text{g,o}}$ for other lane (top right), velocity v_{m} of ego car (bottom left), and lane change indicator l_{m} of ego car (bottom right) over time.

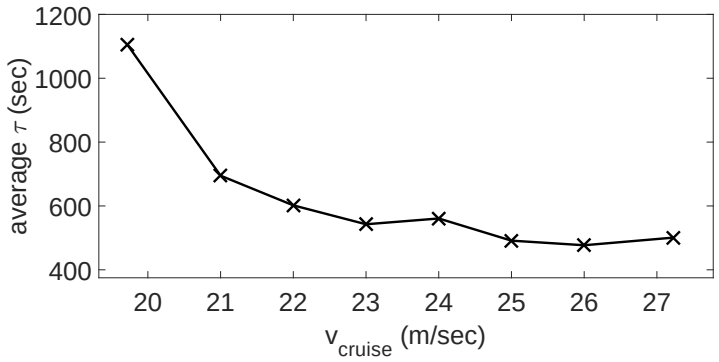

Figure 5.8: Hybrid SMPC – autonomous driving case study: average first exit-time $\bar{\tau}$ (for 1000 random simulations) vs. cruise speed v_{cruise}.

The average first exit-time $\bar{\tau}$ is plotted for different cruise speeds in Figure 5.8. Using $v_{\text{cruise}} = v_{\min}$ promotes defensive driving and yields the largest $\bar{\tau}$, whereas increasing v_{cruise}

reduces $\bar{\tau}$. On the other hand, larger v_{cruise} reduce travel times, improve the traffic flow, and may provide a better travel experience for passengers. Figure 5.9 shows sample trajectories for $v_{\text{cruise}} = 23\,\text{m/sec}$.

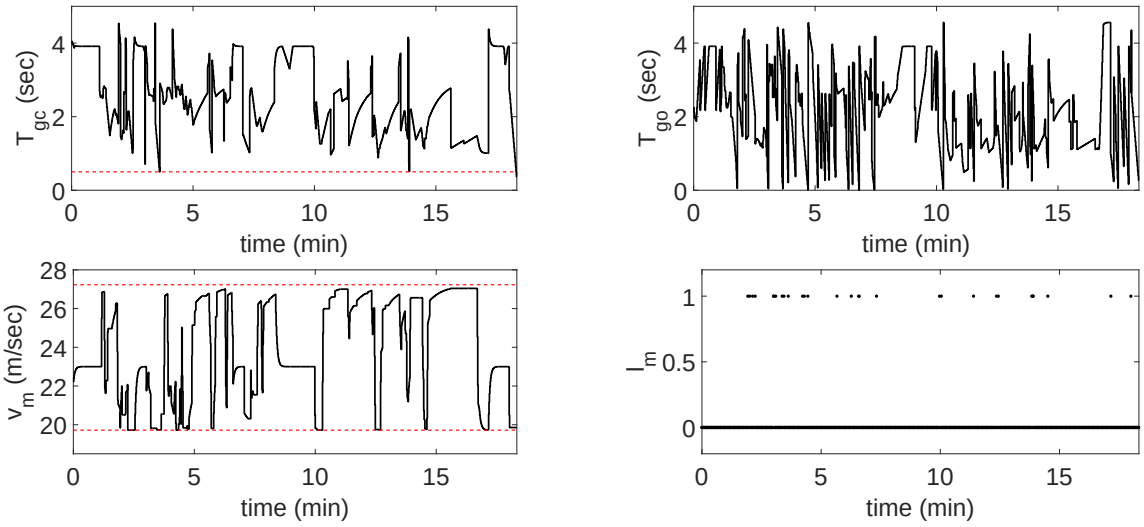

Figure 5.9: Hybrid SMPC with $v_{\text{cruise}} = 23\,\text{m/sec}$ – autonomous driving case study: sample trajectories of relative time gap $T_{\text{g,c}}$ for the ego car's current lane (top left), relative time gap $T_{\text{g,o}}$ for other lane (top right), velocity v_{m} of ego car (bottom left), and lane change indicator l_{m} of ego car (bottom right) over time.

On average, 17 msec are required in MATLAB to compute the control input (a_{m} and l_{m}) at each time instant t with the hybrid SMPC strategy (i.e., execute Steps 2–31 of Algorithm 5.4). A worst-case computation time of 145 msec is recorded (for 1000 random simulations). Besides the lower number of tree nodes ($N = 35$ vs. $N = 50$), the reason for the shorter computation times of the hybrid SMPC strategy is that dividing the control problem into solving three smaller MILPs (one MILP for obtaining lane change decisions and two MILPs for obtaining the accelerations for each lane), rather than solving one large MILP (one MILP for obtaining both lane change decisions and accelerations), is computationally more efficient since MILP is NP-complete and worst-case computation times grow exponentially with the number of decision variables.

5.6 Summary

In this chapter, an SMPC scheme was developed for solving stochastic DCOC problems with the objective of maximizing the average / expected value of the first exit-time. The developed SMPC strategy is based on a tree structure with a specified number of tree nodes

and the tree generation algorithm has been defined to emphasize the inclusion of the most relevant scenarios. For stochastic linear systems, the SMPC strategy obtains solutions arbitrarily close to the optimal solution in terms of average first exit-time performance by repeatedly solving an MILP over a receding time horizon based on the current state vector and disturbance. The effectiveness of the proposed SMPC strategy was demonstrated in two numerical case studies, including a stochastic adaptive cruise control problem. Moreover, the SMPC strategy was applied to the DCOC driving problem for a two-lane road from Section 4.5 and achieved comparable results to the DP-based policies from Section 4.5.4.

The advantage of DP-based techniques (Chapter 4) is that they are quite general and able to address a broad range of DCOC problems without neglecting any effects of the underlying system. However, they suffer from the curse of dimensionality or, in the ADP case, may require extensive tuning and model training. On the other hand, the SMPC approach is computationally more tractable, while, however, relying on a linear model approximation of the underlying system.

CHAPTER 6

Other Developments for Systems with Disturbances

The work in this chapter has been originally published in [52] and is separate from the DCOC developments in Chapters 2 – 5.

6.1 Motivation and Problem Formulation

An LQ optimal control problem for a class of continuous-time linear systems with augmented disturbance term is considered in this chapter. The class of systems is of the form

$$\dot{x}(t) = Ax(t) + Bu(t) + d(t), \;\; x(0) = x_0, \tag{6.1}$$

where A and B are real matrices, $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $d(t) \in \mathbb{R}^n$ is a time-dependent disturbance term which is known in advance. Given an initial state x_0, the objective is to find a control $u(t)$ over the finite time horizon $[0, T]$ that minimizes the quadratic cost functional

$$J = \frac{1}{2}\int_0^T \left[x^{\mathrm{T}}(t)Qx(t) + u^{\mathrm{T}}(t)Ru(t)\right] \mathrm{d}t + \frac{1}{2}x^{\mathrm{T}}(T)Sx(T) \to \min_{u(t)}, \tag{6.2}$$

with $Q = Q^\top \succeq 0$ and $R = R^\top \succ 0$. For notational convenience, indicating the explicit time dependence of time-dependent variables is omitted when it is clear from the context.

Problem (6.2) is relevant to many real-time optimal control applications, in particular, those where preview is available or needs to be incorporated [107–110]. In the continuous-time formulation, the optimal control problem (6.2) leads to a two point boundary value problem (TPBVP) with mixed boundary conditions. A solution to this problem is based on solving the Riccati differential equation. In addition, an ordinary differential equation (ODE) that accounts for the disturbance has to be solved [111, 112]. In general, there

is no explicit solution to this ODE and numerical and approximate approaches need to be developed. Numerical approaches to solve this ODE are based on integrating backwards in time, which may become computationally impractical, especially when larger time horizons are considered.

An new approach is proposed in this chapter (Section 6.2) that solves problem (6.2) by approximating the disturbance term d as a piecewise-linear function of time and a closed-form approximate solution to the TPBVP is obtained. In addition, an upper bound on the error between the optimal solution and the approximate solution when the piecewise-linear disturbance approximation is used is derived in Section 6.3. Such an approximation yields higher accuracy than piecewise-constant disturbance approximation common in sampled data/discrete-time treatments of the problem.

The presented approach allows for fast computation of an approximation of the optimal control, which facilitates potential onboard/real-time implementation. In particular, this may be useful in applications of MPC with previewed disturbance [113–116] or with disturbance scenarios [117], where an LQ problem with disturbance term similar to (6.2) has to be solved repeatedly over a receding time horizon. Future research to address the inclusion of constraints can further extend the use of the proposed technique for MPC with preview to constrained problems.

As a subsequent numerical case study demonstrates (Section 6.4), the proposed strategy can be effective in spacecraft orbital maneuvering problems to account for higher-order gravity perturbations and air drag. Note that in this case study, the disturbance is computed for the trajectory of the nominal/target orbit. Since, in the case study, the spacecraft is relatively close to the known target orbit, the error is small and the simulation results show that the proposed approach is effective in the context of controlling a nonlinear system, in particular, when recomputing the control over a receding time horizon using MPC techniques to account for unmodeled effects (Section 6.4.3). At the same time, given that the focus of the theoretical analysis is an LQ problem with previewed disturbance, simulation results are also included for the linear model in Section 6.4.2 as they illustrate the conclusions from the analysis in a setting consistent with the assumptions in this chapter.

The developments in this chapter are furthermore motivated by enhancing the implementation of a computational strategy to solve nonlinear optimal control problems [118], where one iterates between using d to approximate a nonlinear term $d^{i+1} = \phi(x^i)$ in the equations of motion evaluated on a current iteration i of the trajectory, and solving the optimal control problem (6.2).

6.2 TPBVP Solution

The necessary conditions for optimality in problem (6.2) are provided by Pontryagin's maximum principle applied to the Hamiltonian H,

$$H = \frac{1}{2}\left[x^{\mathrm{T}}Qx + u^{\mathrm{T}}Ru\right] + \psi^{\top}\left[Ax + Bu + d\right]. \tag{6.3}$$

The optimal control minimizes the Hamiltonian. Hence, $u = -R^{-1}B^{\top}\psi$, where ψ denotes the vector of adjoint variables, which satisfy

$$\dot{\psi} = -\left(\partial H/\partial x\right)^{\top} = -Qx - A^{\top}\psi. \tag{6.4}$$

Moreover, the transversality condition $\psi(T) = Sx(T)$ must be satisfied. Consequently, it follows that

$$\begin{bmatrix} \dot{x} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^{\top} \\ -Q & -A^{\top} \end{bmatrix} \begin{bmatrix} x \\ \psi \end{bmatrix} + \begin{bmatrix} d \\ 0_{n\times 1} \end{bmatrix}. \tag{6.5}$$

By defining $\tilde{x}^{\top} = [x^{\top}, \psi^{\top}]$ and $\tilde{d}^{\top} = [d^{\top}, 0_{1\times n}]$, (6.5) may be written as

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{d}, \tag{6.6}$$

with initial condition $\tilde{x}_0^{\top} = \tilde{x}(0) = \left[x_0^{\top},\ \psi_0^{\top}\right]$, where ψ_0 is unknown. Using the transversality condition for ψ, the terminal state values at time T are $\tilde{x}^{\top}(T) = \left[x^{\top}(T),\ (Sx(T))^{\top}\right]$, where $x(T)$ is unknown. However, $x(T)$ or $\tilde{x}(T)$, respectively, can be computed according to

$$\tilde{x}(T) = \mathrm{e}^{\tilde{A}T}\tilde{x}_0 + \int_0^T \mathrm{e}^{\tilde{A}(T-\tau)}\tilde{d}(\tau)\mathrm{d}\tau. \tag{6.7}$$

The key assumption for deriving a closed-form approximate solution to the TPBVP is that the disturbance d is approximated by a piecewise-linear function of time. In this regard, $(\nu-1)$-equidistant time intervals of length Δt are considered: $t_k = (k-1)\Delta t$, $k = 1, 2, ..., \nu$, where the final time is $t_\nu = T = (\nu-1)\Delta t$ and t_1 is at $t = 0$. In analogy, $\tilde{d}_k = \tilde{d}\,(t_k)$ is defined. Then the piecewise-linear approximation (such an approximation can also be referred to, actually more correct, as piecewise-affine) of the disturbance vector $\tilde{d}(t)$ at time t is given by

$$\tilde{d}_{\mathrm{pwlin}}(t) = \tilde{d}_k + \frac{\tilde{d}_{k+1} - \tilde{d}_k}{\Delta t}\left(t - (k-1)\Delta t\right), \quad \text{for } t \in [t_k, t_{k+1}]. \tag{6.8}$$

Consequently, the approximation of (6.7) using (6.8) is

$$\tilde{x}_{\text{pwlin}}(T) = \mathrm{e}^{\tilde{A}T}\tilde{x}_0 + \sum_{k=1}^{\nu-1}\left\{\int_{(k-1)\Delta t}^{k\Delta t} \mathrm{e}^{\tilde{A}(T-\tau)}\left[\tilde{d}_k + \frac{\tilde{d}_{k+1}-\tilde{d}_k}{\Delta t}\left(\tau-(k-1)\Delta t\right)\right]\mathrm{d}\tau\right\}. \tag{6.9}$$

6.2.1 Case $\tilde{A}$ is invertible

When $\tilde{A}$ is non-singular, integration by parts of (6.9) and further simplification yields

$$\begin{aligned}\tilde{x}_{\text{pwlin}}(T) = \mathrm{e}^{\tilde{A}T}\tilde{x}_0 + \tilde{A}^{-1}\sum_{k=1}^{\nu-1}\Bigg\{&\mathrm{e}^{\tilde{A}(\nu-k)\Delta t}\tilde{d}_k - \mathrm{e}^{\tilde{A}(\nu-k-1)\Delta t}\tilde{d}_{k+1}\\ &+ \tilde{A}^{-1}\left(\mathrm{e}^{\tilde{A}(\nu-k)\Delta t} - \mathrm{e}^{\tilde{A}(\nu-k-1)\Delta t}\right)\left(\frac{\tilde{d}_{k+1}-\tilde{d}_k}{\Delta t}\right)\Bigg\}.\end{aligned} \tag{6.10}$$

The constant matrices $K_k \in \mathbb{R}^{2n\times 2n}$ are defined as follows

$$K_k = \mathrm{e}^{\tilde{A}(k-1)\Delta t}, \quad k = 1, 2, ..., \nu, \tag{6.11}$$

and (6.10) is rewritten as

$$\begin{aligned}\tilde{x}_{\text{pwlin}}(T) =& K_\nu\tilde{x}_0 + \tilde{A}^{-1}\left(K_\nu\tilde{d}_1 - K_1\tilde{d}_\nu\right) +\\ &\left(\tilde{A}^{-1}\right)^2\sum_{k=1}^{\nu-1}\left\{\left(K_{\nu+1-k} - K_{\nu-k}\right)\left(\frac{\tilde{d}_{k+1}-\tilde{d}_k}{\Delta t}\right)\right\},\end{aligned} \tag{6.12}$$

which provides $2n$ equations to solve for the $2n$ unknowns contained in ψ_0 and $x(T)$. In order to solve the system of linear equations, the sum of the second and third term in (6.12) is denoted by $q^\top = \left[q_1^\top,\, q_2^\top\right]$,

$$q = \tilde{A}^{-1}\left(K_\nu\tilde{d}_1 - K_1\tilde{d}_\nu\right) + \left(\tilde{A}^{-1}\right)^2\sum_{k=1}^{\nu-1}\left\{\left(K_{\nu+1-k} - K_{\nu-k}\right)\left(\frac{\tilde{d}_{k+1}-\tilde{d}_k}{\Delta t}\right)\right\}, \tag{6.13}$$

where $q_1 \in \mathbb{R}^n$ and $q_2 \in \mathbb{R}^n$. It follows that

$$\tilde{x}_{\text{pwlin}}(T) = \begin{bmatrix} x_{\text{pwlin}}(T) \\ Sx_{\text{pwlin}}(T)\end{bmatrix} = K_\nu\begin{bmatrix} x_0 \\ \psi_{0,\text{pwlin}}\end{bmatrix} + q, \tag{6.14}$$

where $x_{\text{pwlin}}(T)$ and $\psi_{0,\text{pwlin}}$ are the approximations of $x(T)$ and ψ_0 using the piecewise-

linear disturbance term. By noting that $K_\nu = \begin{bmatrix} K_{\nu,11} & K_{\nu,12} \\ K_{\nu,21} & K_{\nu,22} \end{bmatrix}$ and $K_{\nu,ij} \in \mathbb{R}^{n\times n}$, the solution to the TPBVP is given by

$$\begin{bmatrix} x_{\text{pwlin}}(T) \\ \psi_{0,\text{pwlin}} \end{bmatrix} = \begin{bmatrix} I_{n\times n} - K_{\nu,12}C_{\text{inv}}S & K_{\nu,12}C_{\text{inv}} \\ -C_{\text{inv}}S & C_{\text{inv}} \end{bmatrix} \begin{bmatrix} K_{\nu,11}x_0 + q_1 \\ K_{\nu,21}x_0 + q_2 \end{bmatrix}, \tag{6.15}$$

where $C_{\text{inv}} = (SK_{\nu,12} - K_{\nu,22})^{-1}$. Now, following the same steps, the solution to the state equation can be derived, which, at the discrete time steps t_k, reads

$$\begin{aligned} \tilde{x}_{\text{pwlin}}(t_k) =& K_k\tilde{x}_{0,\text{pwlin}} + \tilde{A}^{-1}\left(K_k\tilde{d}_1 - K_1\tilde{d}_k\right) + \\ & \left(\tilde{A}^{-1}\right)^2 \sum_{i=1}^{k-1}\left\{ (K_{k+1-i} - K_{k-i})\left(\frac{\tilde{d}_{i+1} - \tilde{d}_i}{\Delta t}\right)\right\}, \end{aligned} \tag{6.16}$$

and the approximate optimal control at the k-th time instant is

$$u_{\text{pwlin}}(t_k) = -R^{-1}B^\top\left[0_{n\times n}, I_{n\times n}\right]\tilde{x}_{\text{pwlin}}(t_k). \tag{6.17}$$

6.2.2 Case $\tilde{A}$ is not invertible

When $\tilde{A}$ is not invertible it has $p \in \{0, 1, ..., 2n-1\}$ nonzero eigenvalues and there exists an invertible matrix $M \in \mathbb{C}^{2n\times 2n}$ such that $\tilde{A}$ can be decomposed as

$$\tilde{A} = M\begin{bmatrix} J_1 & 0_{p\times(2n-p)} \\ 0_{(2n-p)\times p} & J_2 \end{bmatrix} M^{-1}, \tag{6.18}$$

where $J_1 \in \mathbb{C}^{p\times p}$ is invertible and $J_2 \in \mathbb{R}^{(2n-p)\times(2n-p)}$ is not invertible, see Chapter 6.2 in [119]. Therefore, the integral of the matrix exponential may be written as

$$\int_{t_n}^{t_{n+1}} \mathrm{e}^{\tilde{A}(t_k-\tau)}\mathrm{d}\tau = M\begin{bmatrix} -J_1^{-1}\left(\mathrm{e}^{J_1(t_k-t_{n+1})} - \mathrm{e}^{J_1(t_k-t_n)}\right) & 0_{p\times(2n-p)} \\ 0_{(2n-p)\times p} & \int_{t_n}^{t_{n+1}} \mathrm{e}^{J_2(t_k-\tau)}\mathrm{d}\tau \end{bmatrix} M^{-1}. \tag{6.19}$$

The integral of $\mathrm{e}^{J_2(t_k-\tau)}$ with respect to τ depends on the number (algebraic multiplicity) of zero eigenvalues of $\tilde{A}$ as well as on the dimension of the nullspace of $\tilde{A}$. A procedure that distinguishes between the possible cases may be implemented for computation purposes, see [119]. In order to solve the integral of the matrix exponential in (6.9),

the following indefinite integrals or antiderivatives are defined,

$$\int \mathrm{e}^{\tilde{A}(t_k-\tau)}\mathrm{d}\tau = F_k(\tau) + C, \quad \int F_k(\tau)\mathrm{d}\tau = G_k(\tau) + C, \tag{6.20}$$

where $C \in \mathbb{R}^{2n\times 2n}$ is a constant matrix and $F_k(\tau)$ and $G_k(\tau)$ are the respective antiderivatives for a given k and $\tau \in [0, T]$. Note that, in general, the antiderivative $Z(x)$ of a function $z(x)$, $x \in I$, satisfies $\mathrm{d}Z(x)/\mathrm{d}x = z(x)$ for $x \in I$, where $\int z(x)\mathrm{d}x = Z(x) + C$ and $C = \mathrm{const}$. With F_k and G_k, integration by parts of (6.9) yields

$$\begin{aligned}\tilde{x}_{\mathrm{pwlin}}(T) =& \mathrm{e}^{\tilde{A}T}\tilde{x}_0 + F_\nu(T)\tilde{d}_\nu - F_\nu(0)\tilde{d}_1 + \\ & \sum_{k=1}^{\nu-1}\left\{[G_\nu(k\Delta t) - G_\nu((k-1)\Delta t)]\left[\frac{\tilde{d}_{k+1}-\tilde{d}_k}{\Delta t}\right]\right\}.\end{aligned} \tag{6.21}$$

Based on (6.20), the following constant matrices are defined

$$\tilde{F}_{i,k} = F_k((i-1)\Delta t), \quad \tilde{G}_{i,k} = G_k((i-1)\Delta t), \tag{6.22}$$

with $i = 1, 2, ..., \nu$ and $k = 1, 2, ..., \nu$. Using (6.11) and (6.22), (6.21) may be written as

$$\tilde{x}_{\mathrm{pwlin}}(T) = K_\nu\tilde{x}_0 + \tilde{F}_{\nu,\nu}\tilde{d}_\nu - \tilde{F}_{1,\nu}\tilde{d}_1 + \sum_{k=1}^{\nu-1}\left\{\left[\tilde{G}_{k+1,\nu} - \tilde{G}_{k,\nu}\right]\left[\frac{\tilde{d}_{k+1}-\tilde{d}_k}{\Delta t}\right]\right\}. \tag{6.23}$$

In analogy to (6.13), the constant vector $q^\top = \left[q_1^\top,\, q_2^\top\right]$ is defined in order to solve the TPBVP,

$$q = \tilde{F}_{\nu,\nu}\tilde{d}_\nu - \tilde{F}_{1,\nu}\tilde{d}_1 + \sum_{k=1}^{\nu-1}\left\{\left[\tilde{G}_{k+1,\nu} - \tilde{G}_{k,\nu}\right]\left[\frac{\tilde{d}_{k+1}-\tilde{d}_k}{\Delta t}\right]\right\}. \tag{6.24}$$

Now the initial adjoint variables $\psi_{0,\mathrm{pwlin}}$ and the final state vector $x_{\mathrm{pwlin}}(T)$ can be obtained according to (6.15). Similarly, the solution to the state equations can be derived at the discrete time steps t_k,

$$\tilde{x}_{\mathrm{pwlin}}(t_k) = K_k\tilde{x}_{0,\mathrm{pwlin}} + \tilde{F}_{k,k}\tilde{d}_k - \tilde{F}_{1,k}\tilde{d}_1 + \sum_{i=1}^{k-1}\left\{\left(\tilde{G}_{i+1,k} - \tilde{G}_{i,k}\right)\left(\frac{\tilde{d}_{i+1}-\tilde{d}_i}{\Delta t}\right)\right\}, \tag{6.25}$$

and the approximate optimal control is computed according to (6.17).

6.3 Error Estimation

An upper bound for the error between the optimal solution $\tilde{x}(t)$ and its approximation $\tilde{x}_{\text{pwlin}}(t)$ is derived in this section. The following assumptions are made for the derivation.

Assumption 6.1. The function $\tilde{d}(t)$ is twice continuously differentiable for $t \in [0, T]$.

Assumption 6.2. The matrices S and $\tilde{A}$ are such that $C_T \exp\left(\tilde{A}T\right) = [K_1, K_2]$, where $K_2 \in \mathbb{R}^{n\times n}$ has rank n and $C_T = [S,\ -I_{n\times n}]$.

Numerical experiments suggest that Assumption 6.2 holds. This may be because the diagonal elements of $\exp\left(\tilde{A}T\right)$ are nonzero in most applications (see infinite power series representation for matrix exponentials [119] where the first term is the identity matrix) and S is usually diagonal. In the following, $\|\cdot\| = \|\cdot\|_p$ denotes the p-norm or Hölder norm [119] of a vector or matrix and $\exp(\cdot) = \mathrm{e}^{(\cdot)}$.

Theorem 6.1. *Suppose Assumptions 6.1 and 6.2 hold. Then, for $t \in [0, T]$,*

$$\|\tilde{x}(t) - \tilde{x}_{\text{pwlin}}(t)\| \leq \exp\left(\left\|\tilde{A}\right\| t\right)\left[\max_{a\in[0,T]}\left\|\ddot{\tilde{d}}(a)\right\| 4t\Delta t^2/27 + M(\Delta t^2)\right],$$

where,

$$M(\Delta t^2) = \|C_T\|\left\|\begin{bmatrix} C_T \exp\left(\tilde{A}T\right) \\ C_0 \end{bmatrix}^{-1}\right\| \left\|\exp\left(\tilde{A}T\right)\right\| \max_{a\in[0,T]}\left\|\ddot{\tilde{d}}(a)\right\| 4T\Delta t^2/27,$$

with $C_0 = [I_{n\times n},\ 0_{n\times n}]$ and $C_T = [S,\ -I_{n\times n}]$.

Proof. First, a bound for $\left\|\tilde{d}(t) - \tilde{d}_{\text{pwlin}}(t)\right\|$ is derived. In contrast to the piecewise-linear $\tilde{d}_{\text{pwlin}}(t)$, the continuous $\tilde{d}(t)$ is differentiable at each of the discrete time points t_k, $k = 1, 2, ..., \nu$. Using Taylor's Theorem, $\tilde{d}(t_k)$ and $\tilde{d}(t_{k+1})$ are expressed as

$$\begin{aligned} \tilde{d}(t_k) &= \tilde{d}(t) + (t_k - t)\dot{\tilde{d}}(t) + \frac{1}{2}(t_k - t)^2\ddot{\tilde{d}}(a_1), \\ \tilde{d}(t_{k+1}) &= \tilde{d}(t) + (t_{k+1} - t)\dot{\tilde{d}}(t) + \frac{1}{2}(t_{k+1} - t)^2\ddot{\tilde{d}}(a_2), \end{aligned} \tag{6.26}$$

where $t \in [t_k, t_{k+1}]$, $a_1 \in [t_k, t]$, and $a_2 \in [t, t_{k+1}]$. By noting that $\tilde{d}(t_k) = \tilde{d}_k$ and $\tilde{d}(t_{k+1}) = \tilde{d}_{k+1}$, the expression for $\tilde{d}_{\text{pwlin}}(t)$ in (6.8) can be stated as

$$\tilde{d}_{\text{pwlin}}(t) = \frac{t_{k+1} - t}{\Delta t}\tilde{d}(t_k) - \frac{t_k - t}{\Delta t}\tilde{d}(t_{k+1}). \tag{6.27}$$

Using (6.26) and $\Delta t = t_{k+1} - t_k$, (6.27) becomes

$$\tilde{d}_{\text{pwlin}}(t) = \tilde{d}(t) + \frac{\ddot{\tilde{d}}(a_1)}{2\Delta t}(t_{k+1} - t)(t_k - t)^2 + \frac{\ddot{\tilde{d}}(a_2)}{2\Delta t}(t - t_k)(t_{k+1} - t)^2. \tag{6.28}$$

It follows from (6.28) that, for $t \in [t_k, t_{k+1}]$, the error between $\tilde{d}(t)$ and $\tilde{d}_{\text{pwlin}}(t)$ is bounded by

$$\begin{aligned}
\left\|\tilde{d}_{\text{pwlin}}(t) - \tilde{d}(t)\right\| &\leq \left\| \frac{\ddot{\tilde{d}}(a_1)}{2\Delta t}(t_{k+1} - t)(t_k - t)^2 + \frac{\ddot{\tilde{d}}(a_2)}{2\Delta t}(t - t_k)(t_{k+1} - t)^2 \right\| \\
&\leq \frac{\left\|\ddot{\tilde{d}}(a_1)\right\|}{2\Delta t}(t_{k+1} - t)(t_k - t)^2 + \frac{\left\|\ddot{\tilde{d}}(a_2)\right\|}{2\Delta t}(t - t_k)(t_{k+1} - t)^2 \\
&\leq \frac{2\Delta t^2}{27}\left(\left\|\ddot{\tilde{d}}(a_1)\right\| + \left\|\ddot{\tilde{d}}(a_2)\right\|\right) \\
&\leq \frac{4\Delta t^2}{27} \max_{a \in [t_k, t_{k+1}]} \left\|\ddot{\tilde{d}}(a)\right\|,
\end{aligned} \tag{6.29}$$

since $(t_{k+1} - t)(t_k - t)^2 \leq 4\Delta t^3/27$ and $(t - t_k)(t_{k+1} - t)^2 \leq 4\Delta t^3/27$. The error between the solutions to the state equation (6.6) is denoted by $e(t) = \tilde{x}(t) - \tilde{x}_{\text{pwlin}}(t)$. Using (6.6), the time derivative of the error is $\dot{e}(t) = \tilde{d}(t) - \tilde{d}_{\text{pwlin}}(t) + \tilde{A}e(t)$. Integrating this expression yields

$$e(t) = \int_0^t \left[\tilde{d}(\tau) - \tilde{d}_{\text{pwlin}}(\tau) + \tilde{A}e(\tau)\right] \mathrm{d}\tau + e(0). \tag{6.30}$$

It follows from (6.30) and the triangle inequality that

$$\|e(t)\| \leq \int_0^t \left\|\tilde{d}(\tau) - \tilde{d}_{\text{pwlin}}(\tau)\right\| \mathrm{d}\tau + \int_0^t \left\|\tilde{A}\right\| \|e(\tau)\| \, \mathrm{d}\tau + \|e(0)\|. \tag{6.31}$$

Using the Gronwall-Bellman inequality [120], (6.31) becomes

$$\|e(t)\| \leq \left[\int_0^t \left\|\tilde{d}(\tau) - \tilde{d}_{\text{pwlin}}(\tau)\right\| \mathrm{d}\tau + \|e(0)\|\right] \exp\left(\left\|\tilde{A}\right\| t\right). \tag{6.32}$$

Next, an error bound for $\|e(0)\|$ is derived. Using the transversality condition for the adjoint variables as well as the fact that the initial error between the states is zero, the following equations are obtained

$$C_T e(T) = 0_{n \times 1}, \tag{6.33}$$

$$C_0 e(0) = 0_{n\times 1}, \tag{6.34}$$

where $C_T = [S,\ -I_{n\times n}]$ and $C_0 = [I_{n\times n},\ 0_{n\times n}]$. Using (6.30), (6.33) may be expressed as

$$C_T \exp\left(\tilde{A}T\right) e(0) + C_T \int_0^T \exp\left(\tilde{A}(T-\tau)\right)\left(\tilde{d}(\tau) - \tilde{d}_{\text{pwlin}}(\tau)\right) \mathrm{d}\tau = 0_{n\times 1}. \tag{6.35}$$

Combining (6.34) and (6.35) yields

$$e(0) = \left[\begin{array}{c} C_T \exp\left(\tilde{A}T\right) \\ C_0 \end{array}\right]^{-1} \left[\begin{array}{c} -C_T \int_0^T \exp\left(\tilde{A}(T-\tau)\right)\left(\tilde{d}(\tau) - \tilde{d}_{\text{pwlin}}(\tau)\right) \mathrm{d}\tau \\ 0_{n\times 1} \end{array}\right], \tag{6.36}$$

where the inverse exists by Assumption 6.2. Finally, (6.36) implies that

$$\begin{aligned} \|e(0)\| \leq \|C_T\| & \left\| \left[\begin{array}{c} C_T \exp\left(\tilde{A}T\right) \\ C_0 \end{array}\right]^{-1} \right\| \\ & \times \sum_{k=1}^{T/\Delta t} \left\{ \int_{t_k}^{t_{k+1}} \left\| \exp\left(\tilde{A}(T-\tau)\right)\left(\tilde{d}(\tau) - \tilde{d}_{\text{pwlin}}(\tau)\right) \right\| \mathrm{d}\tau \right\}. \end{aligned} \tag{6.37}$$

Using (6.29) and (6.37), it follows that

$$\begin{aligned} \|e(0)\| &\leq \|C_T\| \left\| \left[\begin{array}{c} C_T \exp\left(\tilde{A}T\right) \\ C_0 \end{array}\right]^{-1} \right\| \max_{\tau\in[0,T]} \left\| \exp\left(\tilde{A}(T-\tau)\right) \right\| \frac{T}{\Delta t} \frac{4\Delta t^3}{27} \max_{a\in[0,T]} \left\| \dddot{\tilde{d}}(a) \right\| \\ &\leq \|C_T\| \left\| \left[\begin{array}{c} C_T \exp\left(\tilde{A}T\right) \\ C_0 \end{array}\right]^{-1} \right\| \left\| \exp\left(\tilde{A}T\right) \right\| \frac{4T\Delta t^2}{27} \max_{a\in[0,T]} \left\| \dddot{\tilde{d}}(a) \right\|. \end{aligned} \tag{6.38}$$

For notational convenience the error bound for $e(0)$ is denoted by M, which is a function of Δt^2,

$$M(\Delta t^2) = \|C_T\| \left\| \left[\begin{array}{c} C_T \exp\left(\tilde{A}T\right) \\ C_0 \end{array}\right]^{-1} \right\| \left\| \exp\left(\tilde{A}T\right) \right\| \frac{4T\Delta t^2}{27} \max_{a\in[0,T]} \left\| \dddot{\tilde{d}}(a) \right\|. \tag{6.39}$$

With (6.38) and (6.39), the error bound for $e(t)$ in (6.32) may be stated as

$$\|e(t)\| \leq \left[t\frac{4\Delta t^2}{27} \max_{a\in[0,T]} \left\| \ddot{\tilde{d}}(a) \right\| + M(\Delta t^2) \right] \exp\left(\left\| \tilde{A} \right\| t \right). \tag{6.40}$$

□

6.4 Numerical Case Study: Spacecraft Orbital Maneuver

The proposed method is applied to a spacecraft orbital maneuvering problem. The control problem, including the nonlinear and linearized spacecraft model, is described in Section 6.4.1. Section 6.4.2 presents the open-loop solution based on the linear model and numerically quantifies the error incurred by the piecewise-linear approximation of the disturbance. In Section 6.4.3, an MPC implementation of the linear-model-based controller is proposed and closed-loop simulations on the nonlinear model are presented. All computations in this section are performed in MATLAB 2015a on a laptop with an i5-6300 processor.

6.4.1 Control Problem

The following nonlinear equations of motion are considered,

$$\ddot{r} = -\frac{\mu}{\|r\|_2^3} r - \frac{1}{2BC}\rho \|\dot{r}\|_2 \dot{r} + f_{\mathrm{g}} + u, \tag{6.41}$$

where r is the position vector of the spacecraft relative to the center of the attracting body, u denotes the vector of control input accelerations, and μ is the gravitational parameter associated with the two-body problem. The second term in (6.41) represents the perturbation due to atmospheric drag, where BC is the spacecraft's ballistic coefficient and ρ is the density of the atmosphere which is computed using the NRLMSISE-00 model [68]. In this model, the effect of a moving atmosphere is neglected. The third term f_{g} in (6.41) is a nonlinear function of r, representing the J_2 and J_3 perturbations, see [64], which, in addition to atmospheric drag, are the major perturbations in LEO. In general, the developed approach allows to consider any kind of disturbances and additional perturbations can be readily included.

An optimal control problem with the cost functional (6.2) is considered, where the quadratic penalty on the control u reflects propellant consumption for a variable specific impulse (VSI) thruster [121]. The linear model is obtained by linearizing the nonlinear model in (6.41) around a circular target/desired orbit and is given by the CW equa-

tions [65]. The CW equations describe the motion of the spacecraft in Hill's frame, where the x-axis is along the radial direction and the z-axis is orthogonal to the orbital plane of the nominal orbit (pointing in the direction of the nominal orbit's angular momentum vector). The y-axis completes the right hand frame. The state vector in Hill's frame is $x^\top = [r_x, r_y, r_z, v_x, v_y, v_z]$, describing the position and velocity of the spacecraft relative to the target orbit. The control vector is $u^\top = [u_x, u_y, u_z]$, where u_x, u_y, and u_z are accelerations in the respective directions of Hill's frame. The time-varying disturbance term as viewed from Hill's frame is $d^\top = [0, 0, 0, d_x, d_y, d_z]$. It is obtained by calculating the respective disturbances, i.e., $-0.5\rho_0 \left\|\dot{r}_0\right\|_2 \dot{r}_0/BC + f_{\mathrm{g},0}$, for the known nominal orbit, and then transforming the vectors from the ECI frame to Hill's frame.

A generic spacecraft of mass $m = 250$ kg is assumed. For the linear model, atmospheric drag is only taken into account along the nominal orbital track direction (y-direction), where a relevant surface area of $A = 5\ m^2$ and a drag coefficient of 2.5 is assumed, yielding a ballistic coefficient of $BC = 20\ \mathrm{kg/m^2}$. The weights for the cost function are chosen as $Q = 0_{6\times6}$, $R = \mathrm{diag}(10, 10, 10)$, and $S = \mathrm{diag}(10^6, 10, 10^6, 10^6, 10^6, 10^6)$, emphasizing achieving desired final state (except for the final position, $r_y(T)$, on the target orbit) with minimum propellant consumption with a VSI low-thrust engine. The emphasis on minimum fuel consumption may be increased by either lowering the diagonal elements of S or increasing the diagonal elements of R, which may, however, increase the error in the final state. Note that the approach does not take into account hard constraints on the state and control input. However, the matrices Q, R, and S provide tuning parameters by which maximum state and control deviators can be affected. Moreover, in some problems, nearly feasible solutions are acceptable, especially when no strictly feasible solution exists.

Two different cases with different target orbits and initial conditions are considered. In each case, the maneuver time T is set to the orbital period of the nominal orbit. The first case assumes a target orbit of 250 km altitude with inclination $i = 30$ deg, right ascension of the ascending node $\mathrm{RAAN} = 50$ deg, and orbital period $T = 1.49$ hours, where the initial condition is given by $r_x(0) = r_y(0) = 20$ km and $r_z(0) = v_x(0) = v_y(0) = v_z(0) = 0$. The second case assumes a 1000 km target orbit with $i = -50$ deg, $\mathrm{RAAN} = 0$, and orbital period $T = 1.75$ hours, where $r_x(0) = -100$ km, $r_y(0) = 60$ km, $r_z(0) = 40$ km, $v_x(0) = -80$ m/sec, $v_y(0) = 10$ m/sec, and $v_z(0) = 40$ m/sec.

6.4.2 Linear Model Results

The results for the linear model (6.1) are analyzed here. Figure 6.1 (top) shows the time history of the relative control input error $u_{\mathrm{rel}}(t) = \|u(t) - u_{\mathrm{pwlin}}(t)\| \,/\, \|u(t)\|$ for the two

test cases using a sampling time of $\Delta t = t_{k+1} - t_k = 100$ sec, where u is the optimal solution when the actual d rather than its approximation is used. Moreover, the average error,

$$e_{\text{avrg}} = \sum_{k=1}^{\nu} \|\tilde{x}(t_k) - \tilde{x}_{\text{pwlin}}(t_k)\| / \nu, \tag{6.42}$$

of the augmented state vector is plotted against Δt in the bottom of Figure 6.1. Note that throughout this section the 2-norm and units of kilometers and seconds are used to compute cost values and norms. The relative control input error incurred by the piecewise-linear approximation of d with $\Delta t = 100$ sec is less than 0.1 percent for most of the maneuver time and never exceeds 0.8 percent according to Figure 6.1. Furthermore, in line with Theorem 6.1, the average error e_{avrg} can be bounded by a quadratic function of Δt.

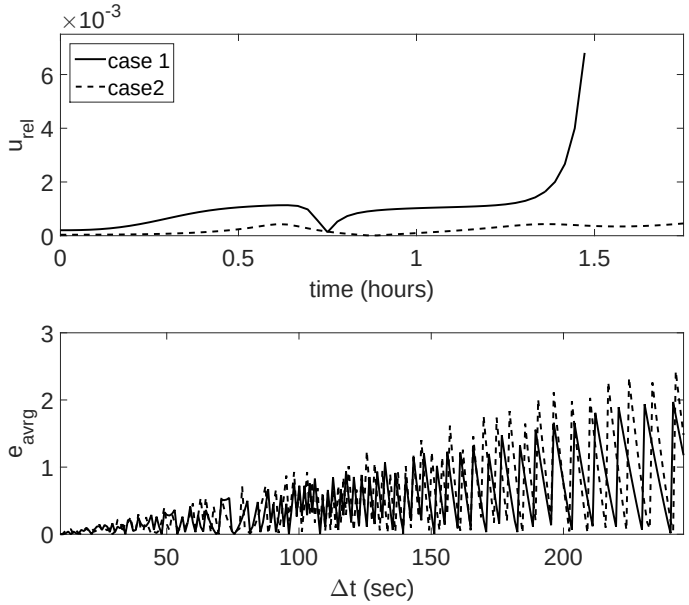

Figure 6.1: LQ optimal control problem of spacecraft orbital maneuvering: linear model results. Top: relative control input error $u_{\text{rel}}(t) = \|u(t) - u_{\text{pwlin}}(t)\| \,/\, \|u(t)\|$ for $\Delta t = 100$ sec. Bottom: average error e_{avrg} according to (6.42) vs. sampling time Δt.

Figure 6.2 shows the required computation times for the proposed method for different sampling times Δt. The total computation time is the sum of the required time to build the matrices in (6.11) and (6.20) (top plot in Figure 6.2), the time to solve the TPBVP, i.e., obtain $\psi_{0,\text{pwlin}}$ (middle plot), and the time to compute the state and control sequences for all t_k (bottom plot). In general, the computation times are decreasing exponentially with increasing Δt and the major part of the total computation time is due to building the matrices (top plot in Figure 6.2). For practical applications, this needs to be done only

once and can be performed offline. Solving the TPBVP and obtaining the state and control sequences is performed substantially faster on the order of milliseconds.

In contrast, when the actual d rather than its approximation is used, the TPBVP solution is obtained numerically using $\mathrm{ode45}$ and fsolve in MATLAB. While the computation time is affected by the initial guess of ψ_0, for default solver settings and an initial guess of $\psi_0 = 0_{6\times 1}$, the computation time is about 7.8 sec for test case 1 and 9.3 sec for test case 2. For poor initial guesses of ψ_0, computation times are longer.

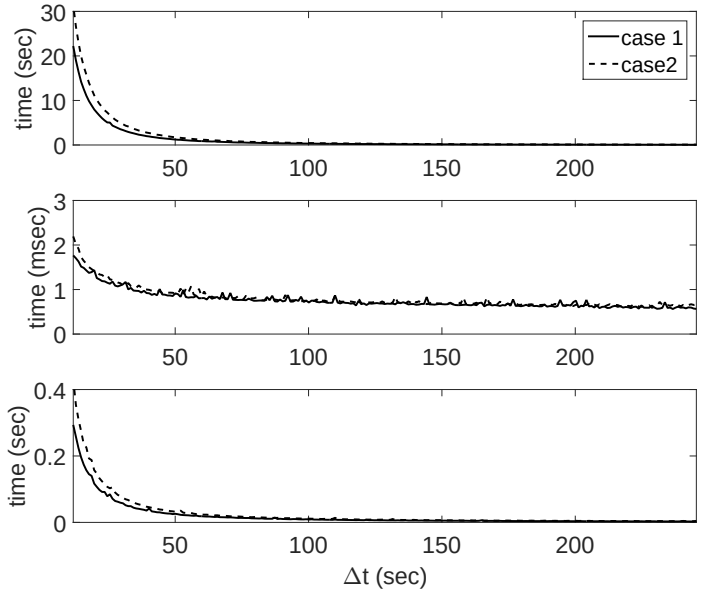

Figure 6.2: LQ optimal control problem of spacecraft orbital maneuvering: computation times vs. Δt. Top: time to build the matrices in (6.11) and (6.20). Middle: time to solve the TPBVP. Bottom: time to compute $\tilde{x}_{\mathrm{pwlin}}(t)$ and $u_{\mathrm{pwlin}}(t)$ for all t_k.

6.4.3 Nonlinear Model Results

An MPC implementation of the proposed approach is used to control the nonlinear spacecraft model. The sampling time is set to $\Delta t = 100$ sec and, based on the current state, the solution to the TPBVP is computed at every sampling instant t_k, $k = 1, 2, ..., \nu - 1$, for a receding and shrinking time horizon $T - t_k$. The controls $u_{\mathrm{pwlin}}(t_k)$ and $u_{\mathrm{pwlin}}(t_{k+1})$ are computed according to (6.17) and the control $u_{\mathrm{interp}}(t) = u_{\mathrm{pwlin}}(t_k) + (u_{\mathrm{pwlin}}(t_{k+1}) - u_{\mathrm{pwlin}}(t_k))(t - t_k)/\Delta t$ is applied to the nonlinear spacecraft model during the sampling interval $t \in [t_k, t_{k+1})$. This receding horizon implementation provides a form of feedback to compensate for unmodeled effects not present in the linear model. While the respective matrices in (6.11) and (6.20) are built offline before the maneuver (computation times

for $\Delta t = 100$ sec: ≈ 0.3 sec for case 1 and ≈ 0.4 sec for case 2, see top plot in Figure 6.2), the computation times for recomputing the control are negligible. For recomputing the control according to the proposed MPC scheme, a worst-case computation time over all sampling instants of 1.5 msec is recorded for case 1 and 1.8 msec for case 2, where the average computation time over all sampling instants is about 1 msec for both cases. Hence, the MPC implementation appears to be suitable for real-time applications. Note that it is not necessary to recompute the disturbance term $d(t)$ since the nominal trajectory given by the target orbit does not change.

Figures 6.3 – 6.5 show the control input accelerations as well as the spacecraft position and velocity relative to the target orbit for cases 1 and 2 using the proposed MPC implementation. In both cases, the controller is able to drive the spacecraft to the desired target orbit.

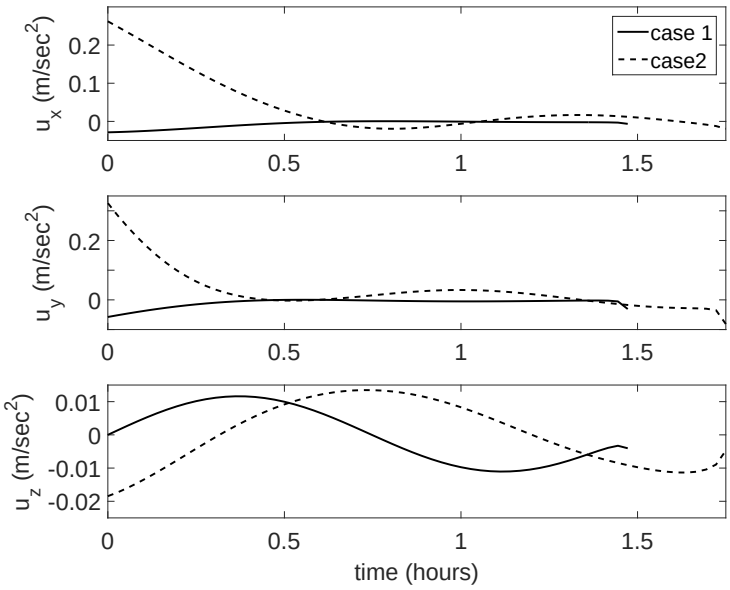

Figure 6.3: LQ optimal control problem of spacecraft orbital maneuvering: nonlinear model results with MPC scheme: control input acceleration in Hill's frame vs. time.

The cost values for the trajectories in Figures 6.3 – 6.5 are listed in Tables 6.1 and 6.2, which also include the final states. In addition to the MPC scheme based on the piecewise-linear approximation of d ($\tilde{d}_{\text{pwlin}}$), Tables 6.1 and 6.2 include the results when using the MPC scheme with either a piecewise-constant approximation of d, i.e., $\tilde{d}_{\text{pwconst}}(t) = [d^\top(t_k), 0_{1\times n}]^\top$ for $t \in [t_k, t_{k+1})$, or without taking into account the disturbance when computing the control ($\tilde{d} \equiv 0_{2n\times 1}$).

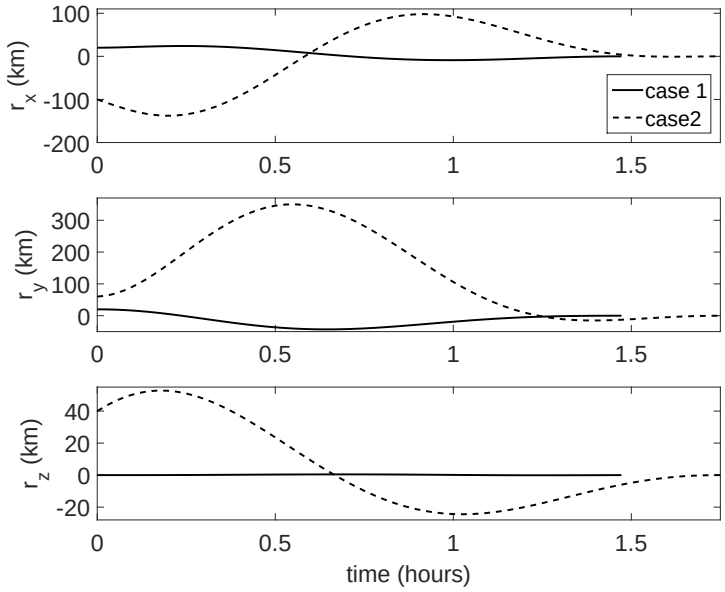

Figure 6.4: LQ optimal control problem of spacecraft orbital maneuvering: nonlinear model results with MPC scheme: spacecraft position in Hill's frame vs. time.

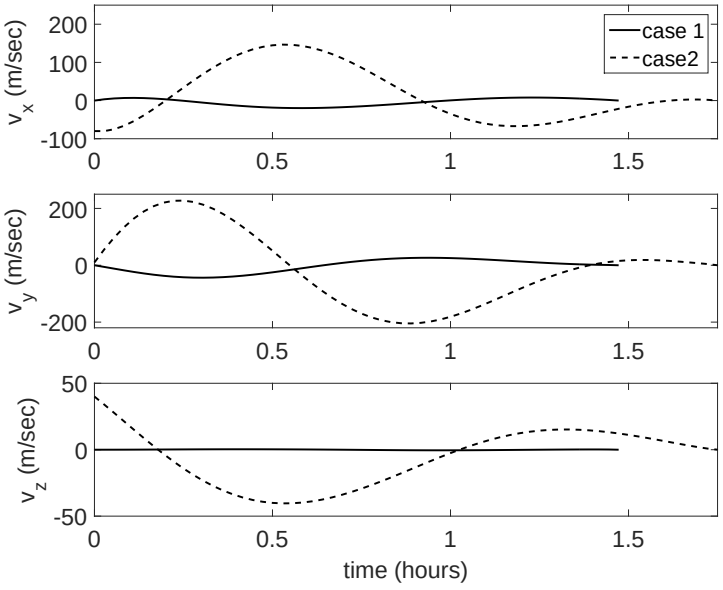

Figure 6.5: LQ optimal control problem of spacecraft orbital maneuvering: nonlinear model results with MPC scheme: spacecraft velocity in Hill's frame vs. time.

It is evident that taking into account the disturbance for computing the control input improves the performance as the controller with $\tilde{d} \equiv 0_{2n\times 1}$ performs poorly compared to the controllers based on $\tilde{d}_{\text{pwlin}}$ and $\tilde{d}_{\text{pwconst}}$. Moreover, the piecewise-linear approximation

of d improves the performance compared to a piecewise-constant approximation, where the advantage of using $\tilde{d}_{\text{pwlin}}$ increases with increasing Δt. Note that the weight for $r_y(T)$ (final position on the target orbit) is smaller compared to the terminal weights on the other states, which explains the deviations compared to the other states in Tables 6.1 and 6.2.

	J	$r_x(T)$, m	$r_y(T)$, m	$r_z(T)$, m	$v_x(T)$, $\frac{\text{m}}{\text{sec}}$	$v_y(T)$, $\frac{\text{m}}{\text{sec}}$	$v_z(T)$, $\frac{\text{m}}{\text{sec}}$
$\tilde{d}_{\text{pwlin}}$	0.16	-0.24	-29.9	-0.5	-3×10^{-3}	-3×10^{-3}	-9×10^{-3}
$\tilde{d}_{\text{pwconst}}$	4.26	-0.87	-31.2	-2.8	-23×10^{-3}	-40×10^{-3}	-77×10^{-3}
$\tilde{d} \equiv 0_{2n\times 1}$	2246	-66.5	-20.5	8.5	-1.3	0.23	0.15

Table 6.1: LQ optimal control problem of spacecraft orbital maneuvering: nonlinear model results with MPC scheme, case 1: cost values and final states for different controllers.

	J	$r_x(T)$, m	$r_y(T)$, m	$r_z(T)$, m	$v_x(T)$, $\frac{\text{m}}{\text{sec}}$	$v_y(T)$, $\frac{\text{m}}{\text{sec}}$	$v_z(T)$, $\frac{\text{m}}{\text{sec}}$
$\tilde{d}_{\text{pwlin}}$	0.42	0.6	-53.6	0.67	-6×10^{-3}	-9×10^{-3}	10×10^{-3}
$\tilde{d}_{\text{pwconst}}$	2.28	0.21	-55.3	2.1	-18×10^{-3}	-60×10^{-3}	54×10^{-3}
$\tilde{d} \equiv 0_{2n\times 1}$	921.1	-42.8	-47	-2.5	-0.88	0.14	-38×10^{-3}

Table 6.2: LQ optimal control problem of spacecraft orbital maneuvering: nonlinear model results with MPC scheme, case 2: cost values and final states for different controllers.

6.5 Summary

Separate from the DCOC developments in this disseratation, an LQ optimal control problem for linear systems with previewed time-varying disturbance term $d(t)$ was considered in this chapter. A closed-form solution was derived based on Pontryagin's maximum principle by approximating $d(t)$ by a piecewise-linear function of time using equidistant time intervals. It was shown that the error due to the piecewise-linear approximation can be bounded by a quadratic function of the length of the time intervals. The closed-form solution can readily be implemented in computer code and allows for fast computations in real-time. Besides, the proposed approach can be used to warm start nonlinear optimal control solvers that require a good initial guess for convergence and can, in addition, handle state and control constraints. The approach was applied to spacecraft orbital maneuvering where two numerical test cases with different initial conditions and target orbits were treated. In both cases, the spacecraft was successfully driven to the prescribed target orbit using an MPC implementation of the developed approach.

CHAPTER 7

Conclusions and Future Directions

7.1 Conclusions

This dissertation focused on theoretical and methodological advances of drift counteraction optimal control (DCOC), in which the goal is to design control algorithms that maximize the time or total yield before a given system violates prescribed constraints. Unlike conventional control problems, there is no set-point or reference trajectory in DCOC, only a set of constraints on the system's process variables that need to be satisfied for as long as possible. For both deterministic and stochastic settings of DCOC, theoretical results were extended by this dissertation in several ways. In particular, properties of the first exit-time and of the objective function of DCOC problems were studied, a solution to the DCOC problem was characterized, and conditions for the existence of a solution were derived.

A variety of new algorithms that obtain solutions or good-quality approximations of solutions to DCOC problems were proposed. Using dynamic programming (DP) techniques, an enhanced version of value iteration (VI) was developed for both deterministic and stochastic DCOC problems, where the value function is updated in proportion to the error in the Bellman equation. The convergence behavior of the enhanced VI algorithm was studied and it was shown that a solution can be obtained significantly faster than with conventional VI in a numerical setting. Another DP-based algorithm, referred to as base-trajectory VI, was also introduced based on the observation that in some problems, such as in space applications, the optimal control action (such as zero thrust) is known for many states. Base-trajectory VI was shown to be more accurate than conventional VI in a numerical setting and able to generate better performing control policies. In addition, approximate dynamic programming (ADP) methods were proposed for both the deterministic and stochastic case in order to mitigate the curse of dimensionality.

The DP-based theoretical characterization of the solution can be used for a broad range of nonlinear system models and quite general DCOC problem formulations. However,

computationally, the DP-based methods are usually limited to lower-dimensional problems due to the curse of dimensionality or, when using ADP methods, extensive tuning and model training may be required to achieve good-quality results. This motivated the development of model predictive control (MPC) approaches to DCOC as accomplished in this dissertation. The proposed MPC approaches assume a linear model and obtain the control input at each time instant by repeatedly solving either a mixed-integer linear program (MILP) or a standard linear program (LP) over a receding time horizon based on the current state of the system. Since the respective LP and MILP were shown to be always feasible under appropriate assumptions, this is computationally more robust compared to DP-based techniques. The robustness of the MPC approaches was further increased by specifying a computation time limit for solving MILPs, where, if the time limit is reached, the solution of the related LP is used instead.

The proposed MPC approach for the deterministic case provides a drift counteraction solution based on linear models which, in fact, under suitable assumptions and in contrast to usual MPC formulations, exactly coincides with the optimal control if there is no model mismatch. Moreover, the MPC approach was also shown to be effective in DCOC of nonlinear systems, obtaining solutions close to the respective optimum in terms of performance. For stochastic systems, the proposed stochastic MPC (SMPC) scheme uses a tree structure to encode the most likely system behavior. As the tree grows (i.e., the number of tree nodes increases), the SMPC solution approaches the solution of the stochastic DCOC problem, assuming the system is linear with additive random disturbances modeled by a Markov chain. Similar to the deterministic case, the SMPC strategy applied to linearized models is able to provide good-quality approximations of solutions to DCOC problems in the case of more general stochastic nonlinear systems.

In addition to the MPC developments, a mixed-integer nonlinear program (MINLP) was presented that obtains open-loop solutions to deterministic DCOC problems and good-quality approximations of a solution are provided by a similar nonlinear program without integer variables.

New practical applications of DCOC were identified with a focus on spacecraft control and driving policies for autonomous vehicles. For spacecraft, satellite station keeping can be formulated as a DCOC problem with the objective of finding a thrust strategy that, given fuel limitations, counteracts drift imposed by orbital perturbations in order to maximize the time that prescribed position constraints are satisfied. Hence, DCOC provides a systematic and direct approach for extending the lifetime of a satellite before either its orbit decays or it runs out of fuel. Furthermore, DCOC can be exploited for spacecraft attitude control (specifically, underactutated spacecraft attitude control), where tight pointing constraints as

well as constraints on momentum exchange devices or fuel need to be satisfied for as long as possible under disturbance torques from, for example, solar radiation pressure or atmospheric drag. In terms of driving policies for automated or autonomous cars, an adaptive cruise control (ACC) problem can be formulated as a DCOC problem. In the ACC case, the objective is to control the acceleration of the follower vehicle such that the distance to the lead vehicle stays within a prescribed range for as long as possible (on average), where the lead vehicle velocity is modeled as a random disturbance. The DCOC car following problem can be extended by also allowing lane changes. In this case, the objective is to generate a driving policy that maximizes the average time that none of the surrounding cars and other traffic participants such as cyclists, pedestrians, etc. (treated as random disturbances) enter a prescribed safe zone around the ego car. Several numerical case studies of such application-oriented DCOC problems and other DCOC problems were successfully treated in this dissertation using the developed framework.

7.2 Future Directions

Many topics for future research remain. Some of them are summarized in what follows.

Extend the framework of DCOC-based driving policies for autonomous cars
The DCOC-based driving policies for autonomous cars (see Sections 4.5 and 5.5.3) need to be tested in different and more sophisticated traffic simulations and, ultimately, in real traffic. In real traffic applications, a recovery controller is furthermore required to resolve cases of constraint violation, i.e., when a car enters the prescribed safe zone around the ego car. Direct comparisons to other driving approaches are necessary, where additional metrics (other than the average first exit-time $\bar{\tau}$) may be required to measure the performance of a driving policy. Moreover, additional traffic scenarios may be considered such as roads with more than two lanes (see Remark 4.2), highway merging and exiting (for example, by prioritizing specific lanes), or intersections.

Derive simpler strategies from optimal strategies for DCOC applications
For practical applications of DCOC, it may be possible to derive simpler (e.g., rule-based) control strategies from optimal control strategies to efficiently provide good-quality suboptimal results. Machine learning techniques can also be used to approximate the solutions to model predictive control problems.

Approaches to DCOC based on necessary conditions and indirect methods

Necessary optimality conditions for DCOC along the lines of Pontryagin's maximum principle deserve further study. The challenge is to acknowledge that, after leaving the set of state constraints G, the state trajectory may reenter G indefinitely many times. One possible approach to address this issue is to describe the system by a hybrid model with two modes. The system dynamics are represented by the first mode, whereas the second mode models a switch to zero dynamics (i.e., $\dot{x} = 0$ or $x_{t+1} = x_t$), which is activated when the state trajectory hits the boundary of G. Such a model may be treated using similar steps as involved in deriving hybrid variants of the maximum principle. In some cases, by an appropriate selection of the set G, it may be possible to guarantee that the trajectory never reenters G after the first exit. In such a case, significant simplifications to the problem and necessary conditions can be made by discarding state constraints prior to the first exit-time and only requiring a terminal constraint that the state is on the boundary of G at the terminal time. In addition, fast indirect numerical schemes need to be developed to solve the resulting boundary value problem.

Set-theoretic treatment of DCOC problems

The solution to DCOC problems may also be pursued using set theoretic methods. For example, for deterministic DCOC problems with the objective of maximizing the first exit-time, knowledge of the sets $\mathcal{K}_m$ in (2.28) allows to construct an optimal control policy according to $\pi^*(x) \in \{u \in U : f(x,u) \in \mathcal{K}_{m-1}\}$ for all $x \in \mathcal{K}_m$ and $m \in \{1, 2, ..., \tau(x, \pi^*) - 1\}$. The sets $\mathcal{K}_m$ may be obtained with a VI-type algorithm using DP techniques similar to the developments in [122].

Robust DCOC

Another approach for handling uncertainty is to assume that its values are set-bounded. Robust DCOC strategies can potentially be obtained by including set-bounded disturbances and/or model parameters into the mathematical programs developed in this dissertation.

Other applications of DCOC

While the focus of this dissertation was on DCOC applications to autonomous cars and spacecraft, DCOC techniques can be applied to many other engineering applications, some of which are discussed in Section 1.2. In addition, an important field of DCOC applications is system/component life-extending control. For example, DCOC may be used to derive control strategies that extend the life of batteries in electric cars. New approaches may be required to handle the relatively long time horizons that may arise in such problems. Other

classes of applications may include control of flexible aircraft where constraints are imposed on deflections, structural loads, and aircraft shape during aircraft maneuvers or when responding to gusts and the aircraft is controlled to maximize time to constraint violation. The application of DCOC to such problems to obtain maneuver and gust load alleviation schemes will require dealing with high order models and multiple aircraft surfaces; it will likely have to be approached within the MPC framework based on linear models.

APPENDIX A

Rotational Dynamics of a Rigid Body with Time-Varying Mass/Inertia Properties

The notations for the derivation in this section are adopted from [123]. The position of a point x relative to a point y is described by the physical vector $\vec{r}_{x/y}$. The physical vector $\vec{r}_{x/y}$ resolved in the frame $\mathrm{F_A}$ is denoted by $\vec{r}_{x/y}|_{\mathrm{A}}$. The time derivative of $\vec{r}_{x/y}$ with respect to the frame $\mathrm{F_A}$ is denoted by $\overset{\mathrm{A}\bullet}{\vec{r}}_{x/y}$. The velocity of a point x relative to a point y with respect to the frame $\mathrm{F_A}$ is $\vec{v}_{x/y/\mathrm{A}} = \overset{\mathrm{A}\bullet}{\vec{r}}_{x/y}$. Likewise, the acceleration of a point x relative to a point y with respect to the frame $\mathrm{F_A}$ is $\vec{a}_{x/y/\mathrm{A}} = \overset{\mathrm{A}\bullet\bullet}{\vec{r}}_{x/y}$.

Let $\mathcal{B}$ be a rigid body and w and z are points. Then the following relation holds between the moment on $\mathcal{B}$ relative to z and the moment on $\mathcal{B}$ relative to w:

$$\vec{M}_{\mathcal{B}/z} = \vec{M}_{\mathcal{B}/w} - \vec{r}_{z/w} \times \vec{f}_{\mathcal{B}}, \tag{A.1}$$

where $\vec{f}_{\mathcal{B}}$ denotes the total force acting on $\mathcal{B}$. Now w is assumed to be an unforced particle (a particle that has no force applied on it [123]). Then

$$\vec{M}_{\mathcal{B}/w} = \overset{\mathrm{A}\bullet}{\vec{H}}_{\mathcal{B}/w/\mathrm{A}}, \tag{A.2}$$

where $\vec{H}_{\mathcal{B}/w/\mathrm{A}}$ denotes the angular momentum of $\mathcal{B}$ relative to w with respect to the frame $\mathrm{F_A}$. Substituting (A.1) into (A.2) yields

$$\vec{M}_{\mathcal{B}/z} = \overset{\mathrm{A}\bullet}{\vec{H}}_{\mathcal{B}/w/\mathrm{A}} - \vec{r}_{z/w} \times \vec{f}_{\mathcal{B}}. \tag{A.3}$$

By denoting the center of mass of the body $\mathcal{B}$ by c, it is straightforward to show that

$$\vec{H}_{\mathcal{B}/w/\mathrm{A}} = \vec{H}_{\mathcal{B}/z/\mathrm{A}} + \vec{r}_{c/z} \times m_{\mathcal{B}} \vec{v}_{z/w/\mathrm{A}} + \vec{r}_{z/w} \times m_{\mathcal{B}} \vec{v}_{c/w/\mathrm{A}}, \tag{A.4}$$

where $m_{\mathcal{B}}$ is the time-varying total mass of the body $\mathcal{B}$. The time derivative of (A.4) with respect to frame $\mathrm{F_A}$ yields

$$\begin{aligned}\overset{\mathrm{A}\bullet}{\vec{H}}_{\mathcal{B}/w/\mathrm{A}} = &\overset{\mathrm{A}\bullet}{\vec{H}}_{\mathcal{B}/z/\mathrm{A}} + \dot{m}_{\mathcal{B}}\left(\vec{r}_{c/z}\times\vec{v}_{z/w/\mathrm{A}} + \vec{r}_{z/w}\times\vec{v}_{c/w/\mathrm{A}}\right)\\ &+ m_{\mathcal{B}}\left(\vec{v}_{c/z/\mathrm{A}}\times\vec{v}_{z/w/\mathrm{A}} + \vec{v}_{z/w/\mathrm{A}}\times\vec{v}_{c/w/\mathrm{A}} + \vec{r}_{z/w}\times\vec{a}_{c/w/\mathrm{A}} + \vec{r}_{c/z}\times\vec{a}_{z/w/\mathrm{A}}\right),\end{aligned} \tag{A.5}$$

with $\dot{m}_{\mathcal{B}} = \mathrm{d}m_{\mathcal{B}}/\mathrm{d}t$. The velocity cross products in the second term on the right-hand side in (A.5) can be simplified as follows

$$\begin{aligned}\vec{v}_{c/z/\mathrm{A}}\times\vec{v}_{z/w/\mathrm{A}} + \vec{v}_{z/w/\mathrm{A}}\times\vec{v}_{c/w/\mathrm{A}} &= \vec{v}_{c/z/\mathrm{A}}\times\vec{v}_{z/w/\mathrm{A}} + \vec{v}_{w/c/\mathrm{A}}\times\vec{v}_{z/w/\mathrm{A}}\\ &= \left(\vec{v}_{c/z/\mathrm{A}} + \vec{v}_{w/c/\mathrm{A}}\right)\times\vec{v}_{z/w/\mathrm{A}}\\ &= \left(\vec{v}_{c/w/\mathrm{A}} + \vec{v}_{w/z/\mathrm{A}} + \vec{v}_{w/c/\mathrm{A}}\right)\times\vec{v}_{z/w/\mathrm{A}}\\ &= -\vec{v}_{z/w/\mathrm{A}}\times\vec{v}_{z/w/\mathrm{A}} = 0.\end{aligned} \tag{A.6}$$

Therefore, (A.5) becomes

$$\begin{aligned}\overset{\mathrm{A}\bullet}{\vec{H}}_{\mathcal{B}/w/\mathrm{A}} = &\overset{\mathrm{A}\bullet}{\vec{H}}_{\mathcal{B}/z/\mathrm{A}} + m_{\mathcal{B}}\left(\vec{r}_{z/w}\times\vec{a}_{c/w/\mathrm{A}} + \vec{r}_{c/z}\times\vec{a}_{z/w/\mathrm{A}}\right)\\ &+ \dot{m}_{\mathcal{B}}\left(\vec{r}_{c/z}\times\vec{v}_{z/w/\mathrm{A}} + \vec{r}_{z/w}\times\vec{v}_{c/w/\mathrm{A}}\right).\end{aligned} \tag{A.7}$$

Substituting (A.7) into (A.3) yields

$$\begin{aligned}\vec{M}_{\mathcal{B}/z} = &\overset{\mathrm{A}\bullet}{\vec{H}}_{\mathcal{B}/z/\mathrm{A}} + m_{\mathcal{B}}\vec{r}_{c/z}\times\vec{a}_{z/w/\mathrm{A}} + \dot{m}_{\mathcal{B}}\vec{r}_{c/z}\times\vec{v}_{z/w/\mathrm{A}}\\ &+ \vec{r}_{z/w}\times\left(m_{\mathcal{B}}\vec{a}_{c/w/\mathrm{A}} + \dot{m}_{\mathcal{B}}\vec{v}_{c/w/\mathrm{A}} - \vec{f}_{\mathcal{B}}\right).\end{aligned} \tag{A.8}$$

Since c is the center of mass of the body $\mathcal{B}$, the translational momentum of $\mathcal{B}$ relative to w with respect to frame $\mathrm{F_A}$ is $\vec{p}_{\mathcal{B}/w/\mathrm{A}} = m_{\mathcal{B}}\vec{v}_{c/w/\mathrm{A}}$ [123]. Thus, $\overset{\mathrm{A}\bullet}{\vec{p}}_{\mathcal{B}/w/\mathrm{A}} = m_{\mathcal{B}}\vec{a}_{c/w/\mathrm{A}} + \dot{m}_{\mathcal{B}}\vec{v}_{c/w/\mathrm{A}}$. Furthermore, $\overset{\mathrm{A}\bullet}{\vec{p}}_{\mathcal{B}/w/\mathrm{A}} = \vec{f}_{\mathcal{B}}$ since $\mathrm{F_A}$ is an inertial frame and w is an unforced particle. Consequently, the last term on the right-hand side in (A.8) is zero and (A.8) becomes

$$\vec{M}_{\mathcal{B}/z} = \overset{\mathrm{A}\bullet}{\vec{H}}_{\mathcal{B}/z/\mathrm{A}} + \vec{r}_{c/z}\times\left(m_{\mathcal{B}}\vec{a}_{z/w/\mathrm{A}} + \dot{m}_{\mathcal{B}}\vec{v}_{z/w/\mathrm{A}}\right). \tag{A.9}$$

Using the transport theorem and introducing the body-fixed frame $\mathrm{F_B}$ with mutually per-

pendicular frame vectors $(\hat{b}_1, \hat{b}_2, \hat{b}_3)$, (A.9) becomes

$$\begin{aligned}\vec{M}_{\mathcal{B}/z} = \overset{\mathrm{B}\bullet}{\vec{H}}_{\mathcal{B}/z/\mathrm{A}} + \vec{\omega}_{\mathrm{B/A}} \times \vec{H}_{\mathcal{B}/z/\mathrm{A}} + \vec{r}_{c/z} \times \Bigg[m_{\mathcal{B}} \Big(\vec{a}_{z/c/\mathrm{B}} + 2\vec{\omega}_{\mathrm{B/A}} \times \vec{v}_{z/c/\mathrm{B}} \\ + \vec{\alpha}_{\mathrm{B/A}} \times \vec{r}_{z/c} + \vec{\omega}_{\mathrm{B/A}} \times (\vec{\omega}_{\mathrm{B/A}} \times \vec{r}_{z/c}) + \vec{a}_{c/w/\mathrm{B}} \Big) \\ + \dot{m}_{\mathcal{B}} \left(\vec{v}_{z/c/\mathrm{B}} + \vec{\omega}_{\mathrm{B/A}} \times \vec{r}_{z/c} + \vec{v}_{c/w/\mathrm{A}} \right) \Bigg],\end{aligned} \tag{A.10}$$

where $\vec{\omega}_{\mathrm{B/A}}$ and $\vec{\alpha}_{\mathrm{B/A}} = \overset{\mathrm{A}\bullet}{\vec{\omega}}_{\mathrm{B/A}} = \overset{\mathrm{B}\bullet}{\vec{\omega}}_{\mathrm{B/A}}$ are the physical angular velocity and angular acceleration vectors, respectively, of frame $\mathrm{F_B}$ relative to frame $\mathrm{F_A}$. Again using the fact that $m_{\mathcal{B}}\vec{a}_{c/w/\mathrm{A}} + \dot{m}_{\mathcal{B}}\vec{v}_{c/w/\mathrm{A}} = \vec{f}_{\mathcal{B}}$, (A.10) may be written as follows

$$\begin{aligned}\vec{M}_{\mathcal{B}/z} = \overset{\mathrm{B}\bullet}{\vec{H}}_{\mathcal{B}/z/\mathrm{A}} + \vec{\omega}_{\mathrm{B/A}} \times \vec{H}_{\mathcal{B}/z/\mathrm{A}} + \vec{r}_{c/z} \times \Bigg[m_{\mathcal{B}} \Big(\vec{a}_{z/c/\mathrm{B}} + 2\vec{\omega}_{\mathrm{B/A}} \times \vec{v}_{z/c/\mathrm{B}} \\ + \vec{\alpha}_{\mathrm{B/A}} \times \vec{r}_{z/c} + \vec{\omega}_{\mathrm{B/A}} \times (\vec{\omega}_{\mathrm{B/A}} \times \vec{r}_{z/c}) \Big) \\ + \dot{m}_{\mathcal{B}} \left(\vec{v}_{z/c/\mathrm{B}} + \vec{\omega}_{\mathrm{B/A}} \times \vec{r}_{z/c} \right) + \vec{f}_{\mathcal{B}} \Bigg].\end{aligned} \tag{A.11}$$

$\vec{H}_{\mathcal{B}/z/\mathrm{A}}$ may be expressed using the physical inertia matrix $\vec{I}_{\mathcal{B}/z}$ of the body $\mathcal{B}$ relative to the point z: $\vec{H}_{\mathcal{B}/z/\mathrm{A}} = \vec{I}_{\mathcal{B}/z}\vec{\omega}_{\mathrm{B/A}}$. Therefore, (A.11) becomes

$$\begin{aligned}\vec{M}_{\mathcal{B}/z} = \overset{\mathrm{B}\bullet}{\vec{I}}_{\mathcal{B}/z}\, \vec{\omega}_{\mathrm{B/A}} + \vec{I}_{\mathcal{B}/z}\vec{\alpha}_{\mathrm{B/A}} + \vec{\omega}_{\mathrm{B/A}} \times \left(\vec{I}_{\mathcal{B}/z}\vec{\omega}_{\mathrm{B/A}} \right) - \vec{r}_{c/z} \times \Bigg[m_{\mathcal{B}} \Big(\vec{a}_{c/z/\mathrm{B}} \\ + 2\vec{\omega}_{\mathrm{B/A}} \times \vec{v}_{c/z/\mathrm{B}} + \vec{\alpha}_{\mathrm{B/A}} \times \vec{r}_{c/z} + \vec{\omega}_{\mathrm{B/A}} \times (\vec{\omega}_{\mathrm{B/A}} \times \vec{r}_{c/z}) \Big) \\ + \dot{m}_{\mathcal{B}} \left(\vec{v}_{c/z/\mathrm{B}} + \vec{\omega}_{\mathrm{B/A}} \times \vec{r}_{c/z} \right) - \vec{f}_{\mathcal{B}} \Bigg].\end{aligned} \tag{A.12}$$

This equation describes the general angular motion of a rigid body $\mathcal{B}$ with center of mass c and time-varying mass and inertia properties. (A.12) is now resolved in the body-fixed frame $\mathrm{F_B}$ for the example axisymmetric spacecraft in Figure 2.14, where $\mathrm{F_B}$ is assumed to be the principal frame. The inertia matrix of the axisymmetric spacecraft $\mathcal{B}$ relative to the

body-fixed point z is expressed in frame $\mathrm{F_B}$ as

$$J_z = \vec{I}_{\mathcal{B}/z}|_{\mathrm{B}} = \begin{bmatrix} J_{\mathrm{T}} & 0 & 0 \\ 0 & J_{\mathrm{T}} & 0 \\ 0 & 0 & J_{\mathrm{R}} \end{bmatrix}. \tag{A.13}$$

The angular velocity vector is resolved in $\mathrm{F_B}$ as $\vec{\omega}_{\mathrm{B/A}}|_{\mathrm{B}} = [\omega_1, \omega_2, \omega_3]^{\mathrm{T}}$, where ω_1, ω_2, and ω_3 are the angular velocity vector projections on the principal axes of $\mathcal{B}$. Likewise, $\vec{\alpha}_{\mathrm{B/A}}|_{\mathrm{B}} = [\dot{\omega}_1, \dot{\omega}_2, \dot{\omega}_3]^{\mathrm{T}}$, $\vec{M}_{\mathcal{B}/z}|_{\mathrm{B}} = [M_1, M_2, M_3]^{\mathrm{T}}$, $\vec{f}_{\mathcal{B}}|_{\mathrm{B}} = [f_1, f_2, f_3]^{\mathrm{T}}$, $\vec{a}_{c/z/\mathrm{B}}|_{\mathrm{B}} = [a_1, a_2, a_3]^{\mathrm{T}}$, $\vec{v}_{c/z/\mathrm{B}}|_{\mathrm{B}} = [v_1, v_2, v_3]^{\mathrm{T}}$, and $\vec{r}_{c/z}|_{\mathrm{B}} = [r_1, r_2, r_3]^{\mathrm{T}}$. It is assumed that the spacecraft mass changes with a constant rate,

$$m_{\mathcal{B}}(t) = m_{\mathcal{B},0} - (\dot{m}_{\mathrm{ox}} + \dot{m}_{\mathrm{f}})t, \tag{A.14}$$

where $m_{\mathcal{B},0}$, $\dot{m}_{\mathrm{ox}}$, and $\dot{m}_{\mathrm{f}}$ are constant scalars. Note that $\dot{m}_{\mathrm{ox}}$ and $\dot{m}_{\mathrm{f}}$ are the mass flow rates of the oxidizer and fuel, respectively. In analogy to (A.14), the time-varying masses of the fuel and oxidizer are

$$m_{\mathrm{f}}(t) = m_{\mathrm{f},0} - \dot{m}_{\mathrm{f}}t, \tag{A.15}$$

$$m_{\mathrm{ox}}(t) = m_{\mathrm{ox},0} - \dot{m}_{\mathrm{ox}}t, \tag{A.16}$$

where $m_{\mathrm{f},0}$ and $m_{\mathrm{ox},0}$ are the initial fuel and oxidizer masses. The time-varying lengths of the remaining fuel l_{f} and oxidizer l_{ox} (see Figure 2.14) are

$$l_{\mathrm{f}}(t) = l_{\mathrm{f},0} - \dot{l}_{\mathrm{f}}t, \tag{A.17}$$

$$l_{\mathrm{ox}}(t) = l_{\mathrm{ox},0} - \dot{l}_{\mathrm{ox}}t, \tag{A.18}$$

where $\dot{l}_{\mathrm{f}} = \dot{m}_{\mathrm{f}}/(\pi r_{\mathrm{E}}^2 \rho_{\mathrm{f}})$ and $\dot{l}_{\mathrm{ox}} = \dot{m}_{\mathrm{ox}}/(\pi r_{\mathrm{E}}^2 \rho_{\mathrm{ox}})$ are constant scalars. Here, ρ_{f} and ρ_{ox} are the fuel and oxidizer density, respectively. The parameter r_{E} is the radius of the engine and tank section of the spacecraft as shown in Figure 2.14. It is assumed that the center of mass is always on the symmetry axis ($\hat{b}_3$-axis as shown in Figure 2.14). Therefore, the first and second component of $\vec{r}_{z/c}|_{\mathrm{B}}$, $\vec{v}_{z/c/\mathrm{B}}|_{\mathrm{B}}$, and $\vec{a}_{z/c/\mathrm{B}}|_{\mathrm{B}}$ are zero: $r_1 = r_2 = 0$, $v_1 = v_2 = 0$, and $a_1 = a_2 = 0$. Using (A.14) – (A.18), the distance between the center of mass c and point z is given by

$$r_3(t) = \frac{C_{r_3} + m_{\mathrm{ox}}(t)\left(l_{\mathrm{e}} + \frac{l_{\mathrm{ox}}(t)}{2}\right) + m_{\mathrm{f}}(t)\left(l_{\mathrm{e}} + l_{\mathrm{ox},0} + \frac{l_{\mathrm{f}}(t)}{2}\right)}{m_{\mathcal{B}}(t)}, \tag{A.19}$$

where $C_{r_3} = m_\mathrm{e} l_\mathrm{e}/2 + m_\mathrm{p}(l_\mathrm{e} + l_{\mathrm{ox},0} + l_{\mathrm{f},0} + l_\mathrm{p}/2)$ is a constant scalar. m_e and m_p denote the masses of the engine and the payload, respectively. The velocity of the center of mass c relative to point z with respect to frame $\mathrm{F_B}$ is given by $v_3 = \dot{r}_3$,

$$v_3(t) = \frac{C_{v_3} - m_\mathrm{ox}(t)\frac{\dot{l}_\mathrm{ox}}{2} - \dot{m}_\mathrm{ox}\frac{l_\mathrm{ox}(t)}{2} - m_\mathrm{f}(t)\frac{\dot{l}_\mathrm{f}}{2} - \dot{m}_\mathrm{f}\frac{l_\mathrm{f}(t)}{2} + r_3(t)(\dot{m}_\mathrm{ox} + \dot{m}_\mathrm{f})}{m_\mathcal{B}(t)}, \tag{A.20}$$

where $C_{v_3} = -\dot{m}_\mathrm{ox} l_\mathrm{e} - \dot{m}_\mathrm{f}(l_\mathrm{e} + l_{\mathrm{ox},0})$ is a constant scalar. Likewise, the acceleration of the center of mass c relative to point z with respect to frame $\mathrm{F_B}$ is given by $a_3 = \dot{v}_3 = \ddot{r}_3$, which yields

$$a_3(t) = \frac{C_{a_3} + 2v_3(t)(\dot{m}_\mathrm{ox} + \dot{m}_\mathrm{f})}{m_\mathcal{B}(t)}, \tag{A.21}$$

where $C_{a_3} = \dot{m}_\mathrm{ox}\dot{l}_\mathrm{ox} + \dot{m}_\mathrm{f}\dot{l}_\mathrm{f}$ is a constant scalar. The time-dependent components of the inertia matrix in (A.13) follow from the parallel axis theorem. The principal moment of inertia about the $\hat{b}_3$ axis relative to point z is given by

$$J_\mathrm{R}(t) = \frac{1}{2}\left(C_{J_\mathrm{R}} + r_\mathrm{E}^2\left[m_\mathrm{ox}(t) + m_\mathrm{f}(t)\right]\right), \tag{A.22}$$

where $C_{J_\mathrm{R}} = r_\mathrm{E}^2 m_\mathrm{e} + r_\mathrm{p}^2 m_\mathrm{p}$ is a constant scalar. The parameter r_p is the radius of the payload section. The principal moment of inertia about the $\hat{b}_1$ or $\hat{b}_2$ axis is

$$\begin{aligned} J_\mathrm{T}(t) =& C_{J_\mathrm{T}} + m_\mathrm{ox}(t)\left(\frac{r^2}{4} + \frac{l_\mathrm{ox}^2(t)}{12} + \left[l_\mathrm{e} + \frac{l_\mathrm{ox}(t)}{2}\right]^2\right) \\ &+ m_\mathrm{f}(t)\left(\frac{r^2}{4} + \frac{l_\mathrm{f}^2(t)}{12} + \left[l_\mathrm{e} + l_{\mathrm{ox},0} + \frac{l_\mathrm{f}(t)}{2}\right]^2\right), \end{aligned} \tag{A.23}$$

where $C_{J_\mathrm{T}} = m_\mathrm{e}\left(\frac{r_\mathrm{E}^2}{4} + \frac{l_\mathrm{e}^2}{3}\right) + m_\mathrm{p}\left(\frac{r_\mathrm{p}^2}{4} + \frac{l_\mathrm{p}^2}{12} + \left[l_\mathrm{e} + l_{\mathrm{ox},0} + l_{\mathrm{f},0} + \frac{l_\mathrm{p}}{2}\right]^2\right)$ is a constant scalar. The time derivatives with respect to frame $\mathrm{F_B}$ of the principal moment of inertia are given by

$$\dot{J}_\mathrm{R} = -\frac{r_\mathrm{E}^2}{2}\left(\dot{m}_\mathrm{ox} + \dot{m}_\mathrm{f}\right), \tag{A.24}$$

$$\begin{aligned} \dot{J}_\mathrm{T}(t) =& C_{\dot{J}_\mathrm{T}} - \left(l_\mathrm{e} + \frac{l_\mathrm{ox}(t)}{3}\right)\left(\dot{m}_\mathrm{ox} l_\mathrm{ox}(t) + m_\mathrm{ox}(t)\dot{l}_\mathrm{ox}\right) \\ &- \left(l_\mathrm{e} + l_{\mathrm{ox},0} + \frac{l_\mathrm{f}(t)}{3}\right)\left(\dot{m}_\mathrm{ox} l_\mathrm{f}(t) + m_\mathrm{f}(t)\dot{l}_\mathrm{f}\right) \\ &- \frac{m_\mathrm{ox}(t) l_\mathrm{ox}(t)\dot{l}_\mathrm{ox} + m_\mathrm{f}(t) l_\mathrm{f}(t)\dot{l}_\mathrm{f}}{3}, \end{aligned} \tag{A.25}$$

where $C_{\dot{J}_\mathrm{T}} = -\dot{m}_\mathrm{ox}\left(r_\mathrm{E}^2/4 + l_\mathrm{e}^2\right) - \dot{m}_\mathrm{f}\left(r_\mathrm{E}^2/4 + (l_\mathrm{e} + l_{\mathrm{ox},0})^2\right)$ is a constant scalar. Note that $\dot{J}_\mathrm{R}$ in (A.24) is a constant scalar. The equations of motion for the example axisymmetric spacecraft with time-varying mass/inertia properties follow from (A.12), yielding

$$\begin{aligned}\dot{\omega}_1 =& \frac{1}{J_\mathrm{T}(t) - r_3^2(t)m_\mathcal{B}(t)}\Big[M_1 + \Big(2r_3(t)v_3(t)m_\mathcal{B}(t) + r_3^2(t)\dot{m}_\mathcal{B} - \dot{J}_\mathrm{T}(t)\Big)\,\omega_1 \\ &+ \big(J_\mathrm{T}(t) - J_\mathrm{R}(t) - r_3^2(t)m_\mathcal{B}(t)\big)\,\omega_2\omega_3 - r_3(t)f_2\Big],\end{aligned} \tag{A.26}$$

$$\begin{aligned}\dot{\omega}_2 =& \frac{1}{J_\mathrm{T}(t) - r_3^2(t)m_\mathcal{B}(t)}\Big[M_2 + \Big(2r_3(t)v_3(t)m_\mathcal{B}(t) + r_3^2(t)\dot{m}_\mathcal{B} - \dot{J}_\mathrm{T}(t)\Big)\,\omega_2 \\ &+ \big(J_\mathrm{R}(t) - J_\mathrm{T}(t) + r_3^2(t)m_\mathcal{B}(t)\big)\,\omega_1\omega_3 + r_3(t)f_1\Big],\end{aligned} \tag{A.27}$$

$$\dot{\omega}_3 = \frac{M_3 - \omega_3\dot{J}_\mathrm{R}}{J_\mathrm{R}(t)}, \tag{A.28}$$

where $\dot{m}_\mathcal{B} = \dot{m}_\mathrm{ox} + \dot{m}_\mathrm{f}$ is a constant scalar. Note that the time dependence of each $m_\mathcal{B}$, r_3, v_3, J_R, J_T, and $\dot{J}_\mathrm{T}$ is explicitly stated in (A.26) – (A.28). It is self-evident that the state variables ω_1, ω_2, and ω_3 as well as $\dot{\omega}_1$, $\dot{\omega}_2$, and $\dot{\omega}_3$ are also time-dependent. The same may be true for the components of the moment M_1, M_2, and M_3 as well as for the components of the total force f_1, f_2, and f_3.

In the following, $M_3 = 0$ is assumed. Thus, the solution for ω_3 is readily obtained since (A.28) is decoupled from (A.26) and (A.27), i.e.,

$$\omega_3(t) = \frac{\omega_{3,0}J_{\mathrm{R},0}}{J_{\mathrm{R},0} + \dot{J}_\mathrm{R}t}, \tag{A.29}$$

where $\dot{J}_\mathrm{R}$ is given by (A.24), $J_{\mathrm{R},0} = J_\mathrm{R}(0) = \left(r_\mathrm{E}^2(m_{\mathrm{ox},0} + m_{\mathrm{f},0}) + C_{J_\mathrm{R}}\right)/2$ is the initial principal angular momentum about the symmetry axis, and $\omega_{3,0}$ is the initial angular velocity about the symmetry axis. The number of differential equations describing the system may be reduced by substituting (A.29) into (A.26) and (A.27).

APPENDIX B

Proof of Theorem 3.2

For the proof of Theorem 3.2, $\{x_t\}$, with $x_0 \in G_0$, denotes a state trajectory corresponding to a control sequence $\{u_t\}$ and the dynamics $x_{t+1} = f_t(x_t, u_t)$.

Proof. For the first part of the proof, show that a solution to MINLP (3.16) is also a solution to (3.5). By Assumption 3.1, there exists at least one control sequence $\{u_t\} \in U_{\text{seq}}$ with corresponding $\{x_t\}$ and first exit-time $\tau(x_0, \{u_t\}) \geq \tau_{\text{lb}}$. Hence, because M is sufficiently large according to Assumption 3.2, $\delta_t \equiv 1$ is feasible. Since the number of possible δ_t sequences is finite and a feasible solution exists for at least one of them, a solution to MINLP (3.16) exists. Suppose that $(\{u_t^{\text{NP}}\}, \{\delta_t^{\text{NP}}\})$ is a solution to (3.16), i.e.,

$$\sum_{t=\tau_{\text{lb}}}^{N} \delta_t^{\text{NP}} \leq \sum_{t=\tau_{\text{lb}}}^{N} \delta_t', \tag{B.1}$$

for all $(\{u_t'\}, \{\delta_t'\})$ that satisfy the constraints in (3.16). Moreover, for a given $\{u_t'\} \in U_{\text{seq}}$, let $\{\bar{\delta}_t'\}$ be such that $\bar{\delta}_t' = 0$ iff $t < \tau(x_0, \{u_t'\})$, which is always feasible with respect to (3.16) due to M being sufficiently large (Assumption 3.2). Consequently, because N is sufficiently large according to Assumption 3.2,

$$\tau(x_0, \{u_t'\}) = \tau_{\text{lb}} + \sum_{t=\tau_{\text{lb}}}^{N} (1 - \bar{\delta}_t') = N + 1 - \sum_{t=\tau_{\text{lb}}}^{N} \bar{\delta}_t'. \tag{B.2}$$

Hence, by (B.1) and (B.2),

$$\begin{aligned} \tau(x_0, \{u_t^{\text{NP}}\}) &= \min\{t : \delta_t^{\text{NP}} = 1\} \\ &= \tau_{\text{lb}} + \sum_{t=\tau_{\text{lb}}}^{N} (1 - \delta_t^{\text{NP}}) = N + 1 - \sum_{t=\tau_{\text{lb}}}^{N} \delta_t^{\text{NP}} \geq N + 1 - \sum_{t=\tau_{\text{lb}}}^{N} \bar{\delta}_t' = \tau(x_0, \{u_t'\}), \end{aligned} \tag{B.3}$$

for all $\{u_t'\} \in U_{\text{seq}}$. It follows that $\{u_t^{\text{NP}}\}$ is a solution to (3.5).

For the second part of the proof, show that a solution to (3.5), which exists by Assumption 3.1, is also a solution to MINLP (3.16). Suppose $\{u_t^*\}$ is a solution to (3.5). Thus,

$$\tau(x_0, \{u_t^*\}) \geq \tau(x_0, \{u_t'\}), \tag{B.4}$$

for all $\{u_t'\} \in U_{\text{seq}}$. Then (3.2) and (3.4), the constraints in (3.16), and $N \geq \tau(x_0, \{u_t\})$ for all $\{u_t\} \in U_{\text{seq}}$ (Assumption 3.2) imply that $\delta_t^* = 1$ for $t \in \{\tau(x_0, \{u_t^*\}), ..., N\}$ and $\delta_t' = 1$ for $t \in \{\tau(x_0, \{u_t'\}), ..., N\}$, where $\{\delta_t^*\}$ is the solution to (3.16) with $\{u_t\} = \{u_t^*\}$ fixed and $\{\delta_t'\}$ is the solution to (3.16) with $\{u_t\} = \{u_t'\}$ fixed. Assuming that the lower bound in (3.16) satisfies $\tau_{\text{lb}} \leq \tau(x_0, \{u_t'\})$, it follows that $\delta_t^* = 0$ for $\tau_{\text{lb}} \leq t < \tau(x_0, \{u_t^*\})$ and $\delta_t' = 0$ for $\tau_{\text{lb}} \leq t < \tau(x_0, \{u_t'\})$. This and (B.4) imply that

$$\begin{aligned} \sum_{t=\tau_{\text{lb}}}^{N} \delta_t^* &= \sum_{t=\tau_{\text{lb}}}^{\tau(x_0,\{u_t^*\})-1} \delta_t^* + \sum_{t=\tau(x_0,\{u_t^*\})}^{N} \delta_t^* \\ &= N + 1 - \tau(x_0, \{u_t^*\}) \leq N + 1 - \tau(x_0, \{u_t'\}) = \sum_{t=\tau_{\text{lb}}}^{N} \delta_t', \end{aligned} \tag{B.5}$$

for all $(\{u_t'\}, \{\delta_t'\})$ that satisfy the constraints of MINLP (3.16). Thus, $(\{u_t^*\}, \{\delta_t^*\})$ is a solution of MINLP (3.16). □

APPENDIX C

Nonlinear Model for GEO Satellite Station Keeping Problem

The derivation of the nonlinear spacecraft model for the GEO station keeping problem in Section 3.4.2.1 is described here. With frame $\mathcal{I}$ as the ECI frame and frame $\mathcal{H}$ as the Hill's frame, a vector $\vec{r}$, resolved in frame $\mathcal{I}$, is transformed into frame $\mathcal{H}$ according to $\vec{r}|_{\mathcal{H}} = O_{\mathcal{H}/\mathcal{I}}\vec{r}|_{\mathcal{I}}$, where $O_{\mathcal{H}/\mathcal{I}}$ is the respective orientation matrix. Likewise, $\vec{r}|_{\mathcal{I}} = O_{\mathcal{H}/\mathcal{I}}^{\top}\vec{r}|_{\mathcal{H}} = O_{\mathcal{I}/\mathcal{H}}\vec{r}|_{\mathcal{H}}$. In the following, $\bar{r} = \vec{r}|_{\mathcal{H}}$ is used to denote a vector resolved in Hill's frame. Furthermore, the time derivative of a vector $\vec{r}$ with respect to a frame $\mathcal{F}$ is denoted by $\overset{\mathcal{F}\bullet}{\vec{r}}$. The spacecraft position vector relative to Earth's center is denoted by $\vec{r}_{\mathrm{SC/E}}$ and the velocity and acceleration vectors with respect to frame $\mathcal{I}$ are $\vec{v}_{\mathrm{SC/E}/\mathcal{I}} = \overset{\mathcal{I}\bullet}{\vec{r}}_{\mathrm{SC/E}}$ and $\vec{a}_{\mathrm{SC/E}/\mathcal{I}} = \overset{\mathcal{I}\bullet\bullet}{\vec{r}}_{\mathrm{SC/E}}$, respectively. Thus, employing the two-body problem in continuous-time [65], it follows that

$$\vec{a}_{\mathrm{SC/E}/\mathcal{I}} = -\frac{\mu_{\mathrm{E}}}{\left\|\vec{r}_{\mathrm{SC/E}}\right\|_2^3}\vec{r}_{\mathrm{SC/E}} + \frac{\vec{F}}{m_{\mathrm{SC}}} + \vec{d}_{\mathrm{p}}, \tag{C.1}$$

where μ_{E} is Earth's gravitational parameter, $\vec{F}$ denotes the thrust vector, m_{SC} is the spacecraft mass, and $\vec{d}_{\mathrm{p}}$ is a vector containing perturbing accelerations. Instead of (C.1), the spacecraft motion is described relative to a GEO reference orbit, i.e., $\vec{r}_{\mathrm{SC/GEO}}$. Hence, an expression for the relative acceleration vector with respect to Hill's frame, $\vec{a}_{\mathrm{SC/GEO}/\mathcal{H}}$, needs to be derived. It is $\vec{a}_{\mathrm{SC/GEO}/\mathcal{H}} = \vec{a}_{\mathrm{SC/E}/\mathcal{H}} - \vec{a}_{\mathrm{GEO/E}/\mathcal{H}}$, where

$$\bar{a}_{\mathrm{GEO/E}/\mathcal{H}} = -\frac{\mu_{\mathrm{E}}}{r_0^3}\begin{bmatrix} r_0 & 0 & 0 \end{bmatrix}^{\top}, \tag{C.2}$$

with r_0 as the constant distance between the GEO reference orbit and Earth's center. On the other hand,

$$\vec{a}_{\mathrm{SC/E/\mathcal{H}}} = \vec{a}_{\mathrm{SC/E/\mathcal{I}}} + 2\vec{\omega}_{\mathcal{I}/\mathcal{H}} \times \vec{v}_{\mathrm{SC/E/\mathcal{I}}} + \vec{\omega}_{\mathcal{I}/\mathcal{H}} \times (\vec{\omega}_{\mathcal{I}/\mathcal{H}} \times \vec{r}_{\mathrm{SC/E}}), \tag{C.3}$$

where $\vec{\omega}_{\mathcal{I}/\mathcal{H}}$ denotes the angular velocity vector of frame $\mathcal{I}$ relative to frame $\mathcal{H}$. Given the constant angular rate of the GEO reference orbit,

$$n_0 = \sqrt{\mu_{\mathrm{E}}/r_0^3}, \tag{C.4}$$

$\vec{\omega}_{\mathcal{I}/\mathcal{H}}$ is resolved in Hill's frame as $\bar{\omega}_{\mathcal{I}/\mathcal{H}} = [0, 0, -n_0]^\top$. Furthermore,

$$\bar{r}_{\mathrm{SC/GEO}} = [r_1, r_2, r_3]^\top,$$

$$\bar{v}_{\mathrm{SC/GEO/\mathcal{H}}} = [v_1, v_2, v_3]^\top,$$

$$\bar{v}_{\mathrm{GEO/E/\mathcal{I}}} = [0, v_0, 0]^\top,$$

with $v_0 = \sqrt{\mu_{\mathrm{E}}/r_0}$. Using (C.1), (C.3) is resolved in Hill's frame as follows

$$\begin{aligned}\bar{a}_{\mathrm{SC/E/\mathcal{H}}} = &- \frac{\mu_{\mathrm{E}}}{\sqrt{(r_1+r_0)^2 + r_2^2 + r_3^2}^3} \begin{bmatrix} r_1 + r_0 \\ r_2 \\ r_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -2n_0 \end{bmatrix} \times \begin{bmatrix} v_1 - n_0 r_2 \\ v_2 + v_0 + n_0 r_1 \\ v_3 \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ 0 \\ -n_0 \end{bmatrix} \times \begin{bmatrix} n_0 r_2 \\ -n_0(r_1 + r_0) \\ 0 \end{bmatrix} + \frac{1}{m_{\mathrm{SC}}} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} + \begin{bmatrix} d_{\mathrm{p},1} \\ d_{\mathrm{p},2} \\ d_{\mathrm{p},3} \end{bmatrix},\end{aligned} \tag{C.5}$$

where $\bar{F} = \vec{F}|_{\mathcal{H}} = [F_1, F_2, F_3]^\top$ and $\bar{d}_{\mathrm{p}} = \vec{d}_{\mathrm{p}}|_{\mathcal{H}} = [d_{\mathrm{p},1}, d_{\mathrm{p},2}, d_{\mathrm{p},3}]^\top$. Combining (C.2), (C.4), and (C.5) yields

$$\begin{aligned}\bar{a}_{\mathrm{SC/GEO/\mathcal{H}}} = &- \frac{\mu_{\mathrm{E}}}{\sqrt{(r_1+r_0)^2 + r_2^2 + r_3^2}^3} \begin{bmatrix} r_1 + r_0 \\ r_2 \\ r_3 \end{bmatrix} + \begin{bmatrix} 2n_0 v_2 + n_0^2 r_1 + \frac{\mu_{\mathrm{E}}}{r_0^2} \\ -2n_0 v_1 + n_0^2 r_2 \\ 0 \end{bmatrix} \\ &+ \frac{1}{m_{\mathrm{SC}}} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} + \begin{bmatrix} d_{\mathrm{p},1} \\ d_{\mathrm{p},2} \\ d_{\mathrm{p},3} \end{bmatrix}.\end{aligned} \tag{C.6}$$

For the disturbance term $\vec{d}_{\mathrm{p}}$, perturbing accelerations due to the gravity of the Moon and Sun, SRP, and J2 are taken into account (additional perturbations can readily be included). Thus,

$$\vec{d}_{\mathrm{p}} = \vec{d}_{\mathrm{p,M}} + \vec{d}_{\mathrm{p,S}} + \vec{d}_{\mathrm{p,srp}} + \vec{d}_{\mathrm{p,J2}}. \tag{C.7}$$

The perturbations due to the gravity of Moon and Sun may be obtained according to the three-body problem (spacecraft/Earth/Moon or spacecraft/Earth/Sun, respectively), yielding

$$\vec{d}_{\mathrm{p,M}} = \mu_{\mathrm{M}} \left(\frac{\vec{r}_{\mathrm{M/SC}}}{\left\| \vec{r}_{\mathrm{M/SC}} \right\|_2^3} - \frac{\vec{r}_{\mathrm{M/E}}}{\left\| \vec{r}_{\mathrm{M/E}} \right\|_2^3} \right), \tag{C.8}$$

$$\vec{d}_{\mathrm{p,S}} = \mu_{\mathrm{S}} \left(\frac{\vec{r}_{\mathrm{S/SC}}}{\left\| \vec{r}_{\mathrm{S/SC}} \right\|_2^3} - \frac{\vec{r}_{\mathrm{S/E}}}{\left\| \vec{r}_{\mathrm{S/E}} \right\|_2^3} \right), \tag{C.9}$$

where $\vec{r}_{\mathrm{M/SC}}$ and $\vec{r}_{\mathrm{S/SC}}$ are the position vectors of the Moon and Sun relative to the spacecraft, respectively, and the position vectors of the Moon and Sun relative to Earth are $\vec{r}_{\mathrm{M/E}}$ and $\vec{r}_{\mathrm{S/E}}$, respectively. The gravitational parameters of the Moon and Sun are denoted by μ_{M} and μ_{S}, respectively. The SRP and J2 perturbations are given by [64, 80]

$$\vec{d}_{\mathrm{p,srp}} = -C_{\mathrm{srp}} \frac{S_{\mathrm{SC}}(1 + c_{\mathrm{refl}})}{2m_{\mathrm{SC}}} \frac{\vec{r}_{\mathrm{S/SC}}}{\left\| \vec{r}_{\mathrm{S/SC}} \right\|_2}, \tag{C.10}$$

$$\vec{d}_{\mathrm{p,J2}} = \frac{3\mu_E J_2 r_{\mathrm{E}}^2}{2 \left\| \vec{r}_{\mathrm{SC/E}} \right\|_2^5} \left[\left(5 \frac{\vec{r}_{\mathrm{SC/E}} \cdot \hat{k}_{\mathcal{I}}}{\left\| \vec{r}_{\mathrm{SC/E}} \right\|_2^2} - 1 \right) \vec{r}_{\mathrm{SC/E}} - 2(\vec{r}_{\mathrm{SC/E}} \cdot \hat{k}_{\mathcal{I}}) \hat{k}_{\mathcal{I}} \right], \tag{C.11}$$

where C_{srp} and c_{refl} are constants, S_{SC} is the spacecraft's solar-facing area, r_{E} is Earth's equatorial radius, $\hat{k}_{\mathcal{I}}$ is the 3-axis unit vector of frame $\mathcal{I}$, and $J_2 = 1.08264 \times 10^{-3}$. Now the discrete-time model is obtained from the continuous-time model given by (C.6) and (C.7)–(C.11) using Euler's forward method, yielding

$$\begin{bmatrix} \bar{r}_{\mathrm{SC/GEO},t+1} \\ \bar{v}_{\mathrm{SC/GEO}/\mathcal{H},t+1} \end{bmatrix} = \begin{bmatrix} \bar{r}_{\mathrm{SC/GEO},t} \\ \bar{v}_{\mathrm{SC/GEO}/\mathcal{H},t} \end{bmatrix} + \Delta t \begin{bmatrix} \bar{v}_{\mathrm{SC/GEO}/\mathcal{H},t} \\ \bar{a}_{\mathrm{SC/GEO}/\mathcal{H},t} \end{bmatrix}, \tag{C.12}$$

where Δt is the sampling time and (C.12) may be expressed as in (3.23) and (3.24).

APPENDIX D

Nonlinear Model for Spacecraft Attitude Control Problem

The nonlinear spacecraft model for the attitude control problem in Section 3.4.2.1 is adopted from [79]. As in Appendix C, for the derivation of the model, a physical vector is denoted by $\vec{r}$ and a physical unit vector is denoted by $\hat{r}$. The mathematical vector $\vec{r}|_{\mathcal{F}}$ is obtained by resolving $\vec{r}$ in a given frame $\mathcal{F}$. Two frames are considered, an inertial reference frame denoted by $\mathcal{I}$ and the spacecraft body-fixed frame $\mathcal{B}$, which is assumed to be a principal frame. The notation $\bar{r} = \vec{r}|_{\mathcal{B}}$ is used to denote $\vec{r}$ resolved in frame $\mathcal{B}$. The skew-symmetric matrix associated with $\bar{r} = [r_1, r_2, r_3]^\top$ is defined as

$$S[\bar{r}] = \begin{bmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{bmatrix}. \tag{D.1}$$

The orientation of frame $\mathcal{B}$ relative to frame $\mathcal{I}$ is described by the 3-2-1 Euler angles ψ (yaw), θ (pitch), and ϕ (roll). The continuous-time kinematic equations are given by [124]

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{\cos(\theta)} \begin{bmatrix} \cos(\theta) & \sin(\phi)\sin(\theta) & \cos(\phi)\sin(\theta) \\ 0 & \cos(\phi)\cos(\theta) & -\sin(\phi)\cos(\theta) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \bar{\omega}_{\mathcal{B}/\mathcal{I}}, \tag{D.2}$$

where $\vec{\omega}_{\mathcal{B}/\mathcal{I}}$ is the angular velocity of frame $\mathcal{B}$ relative to $\mathcal{I}$ and $\bar{\omega}_{\mathcal{B}/\mathcal{I}} = [\omega_1, \omega_2, \omega_3]^\top$.

The spacecraft is equipped with p RWs, where $\bar{g}_i$ denotes the unit vector of the ith RW spin axis resolved in the $\mathcal{B}$ frame. The spin rate of the ith RW is ν_i and $\bar{\nu} = [\nu_1, \nu_2, ..., \nu_p]^\top$. Moreover, let

$$W = [\bar{g}_1, \bar{g}_2, ..., \bar{g}_p]. \tag{D.3}$$

All RWs are assumed to be identical and thin (moments of inertia about axes transversal

to spin axis are approximately zero). The moment of inertia about the RW spin axis is denoted by J_{w} and the moment of inertia matrix of the spacecraft bus resolved in the $\mathcal{B}$ frame is given by $J = \mathrm{diag}(J_1, J_2, J_3)$. The locked inertia is defined as

$$\bar{J} = J + J_{\mathrm{w}} W W^{\top}. \tag{D.4}$$

The continuous-time rotational dynamics of the spacecraft described in the $\mathcal{B}$ frame are given by [79]

$$\bar{J}\dot{\bar{\omega}}_{\mathcal{B}/\mathcal{I}} + S[\bar{\omega}_{\mathcal{B}/\mathcal{I}}](\bar{J}\bar{\omega}_{\mathcal{B}/\mathcal{I}} + J_{\mathrm{w}} W \bar{\nu}) + J_{\mathrm{w}} W \dot{\bar{\nu}} = \bar{\tau}_{\mathrm{srp}}, \tag{D.5}$$

where $\dot{\bar{\omega}}_{\mathcal{B}/\mathcal{I}} = \overset{\mathcal{B}\bullet}{\bar{\omega}}_{\mathcal{B}/\mathcal{I}}$ and $\dot{\bar{\nu}} = \overset{\mathcal{B}\bullet}{\bar{\nu}}$ are the time derivatives with respect to frame $\mathcal{B}$ (and resolved in frame $\mathcal{B}$) of the spacecraft and RW angular velocity vectors, respectively. Note that $\overset{\mathcal{B}\bullet}{\bar{\omega}}_{\mathcal{B}/\mathcal{I}} = \overset{\mathcal{I}\bullet}{\bar{\omega}}_{\mathcal{B}/\mathcal{I}}$ and $\overset{\mathcal{B}\bullet}{\bar{\nu}} \approx \overset{\mathcal{I}\bullet}{\bar{\nu}}$ since the RWs are assumed to rotate orders of magnitude faster than the spacecraft bus, i.e., $\|\bar{\nu}\|_1 \gg \left\|\bar{\omega}_{\mathcal{B}/\mathcal{I}}\right\|_1$.

The symbol $\bar{\tau}_{\mathrm{srp}}$ in (D.5) denotes an external torque due to SRP, which is modeled based on the assumption of a cuboid spacecraft with six flat panels. With C_{diff} as the diffusion coefficient, which is assumed to be the same for all panels, $\beta = (4/9)C_{\mathrm{diff}}$ is defined. Moreover, according to [125],

$$\kappa = \frac{\Phi_{\mathrm{S}}}{c(r_{\mathrm{SC/S}}/r_{\mathrm{E/S}})^2}, \tag{D.6}$$

where c is the speed of light, Φ_{S} is the solar flux acting on the spacecraft, $r_{\mathrm{E/S}} = 1\,\mathrm{AU}$ is the nominal distance between Earth and Sun, and $r_{\mathrm{SC/S}}$ is the distance between the spacecraft and Sun, assuming $r_{\mathrm{SC/S}} = 0.99\,\mathrm{AU}$ for this problem. Under the assumption that the SRP acts identically across all points on the jth panel, the SRP acting on panel j may be expressed as follows [79, 125]

$$\vec{P}_j = -\kappa(\hat{q}_j \cdot \hat{q}_{\mathrm{S}})(\hat{q}_j + \beta \hat{q}_{\mathrm{S}}), \tag{D.7}$$

where $\hat{q}_j$ is the normal to the surface of the jth panel (pointing outward from the spacecraft) and $\hat{q}_{\mathrm{S}}$ points from the spacecraft towards the Sun. It follows that the SRP torque due to the jth panel resolved in frame $\mathcal{B}$ is given by

$$\bar{\tau}_{\mathrm{srp},j} = S[\bar{r}_{j/O} - \bar{r}_{C/O}] A_j \bar{P}_j, \tag{D.8}$$

where $\bar{r}_{C/O} = [l_x, l_y, l_z]^{\top}$ denotes the position vector of the spacecraft's center of mass, C, relative to the geometric center, O, of the cuboid. The position vector of the geometric

center of the jth panel relative to O is given by $\bar{r}_{j/O}$, where $j \in \{x+, x-, y+, y-, z+, z-\}$. Thus,

$$\bar{r}_{x+/O} = -\bar{r}_{x-/O} = [L_x/2, 0, 0]^\top,$$

$$\bar{r}_{y+/O} = -\bar{r}_{y-/O} = [0, L_y/2, 0]^\top,$$

$$\bar{r}_{z+/O} = -\bar{r}_{z-/O} = [0, 0, L_z/2]^\top,$$

where the surface areas of the panels are given by $A_{x+} = A_{x-} = L_y L_z$, $A_{y+} = A_{y-} = L_x L_z$, and $A_{z+} = A_{z-} = L_x L_y$. The total SRP torque is the sum of all panel contributions,

$$\bar{\tau}_{\mathrm{srp}} = \sum_{j=1}^{6} \bar{\tau}_{\mathrm{srp},j} I_j, \tag{D.9}$$

where $I_j = 1$ if $\hat{q}_j \cdot \hat{q}_{\mathrm{S}} > 0$, i.e., the jth panel is facing the Sun, and $I_j = 0$ otherwise.

The discrete-time nonlinear model in (3.42) is obtained from the continuous-time nonlinear model given by (D.2)–(D.9) using Euler's forward method.

APPENDIX E

Proof of Lemma 4.1

Proof. Define $h_w^x(u) = \bar{V}(x, u, w)$. It needs to be shown that, for every $\varepsilon > 0$, there exists $\delta > 0$ such that $v, u \in U$ and $\|v - u\| < \delta$ imply $h_w^x(v) < h_w^x(u) + \varepsilon$ for all $x \in G$ and $w \in W$, see Definition 2.1. Since $f(x, u, w)$ is continuous with respect to $u \in U$ for all $x \in G$ and $w \in W$, for every $\varepsilon_1 > 0$, there exists $\delta_1 > 0$ such that

$$\|v - u\| < \delta_1 \Rightarrow \|f(x, v, w) - f(x, u, w)\| < \varepsilon_1, \tag{E.1}$$

for all $x \in G$ [54]. Moreover, since V is USC with respect to $x \in G$ for all $w \in W$, for any $\varepsilon_2 > 0$, there exists $\delta_2 > 0$ such that, for all $x \in G$ and $w \in W$,

$$\begin{aligned} &\|y - f(x, u, w)\| < \delta_2 \\ &\Rightarrow \sum_{w^i \in W} \left[V(y, w^i) P_W(w^i|w)\right] < \sum_{w^i \in W} \left[V(f(x, u, w), w^i) P_W(w^i|w)\right] + \varepsilon_2. \end{aligned} \tag{E.2}$$

For $\varepsilon > 0$, using (E.2) with $y = f(x, v, w)$, there exists $\delta_2 > 0$ such that

$$\|f(x, v, w) - f(x, u, w)\| < \delta_2 \Rightarrow h_w^x(v) < h_w^x(u) + \varepsilon, \tag{E.3}$$

according to (4.18). By taking $\varepsilon_1 = \delta_2$ and $\delta_1 = \delta > 0$ in (E.1), $\|v - u\| < \delta$ implies $h_w^x(v) < h_w^x(u) + \varepsilon$.

□

BIBLIOGRAPHY

[1] Lions, P. L., "On the Hamilton-Jacobi-Bellman equations," *Acta Applicandae Mathematica*, Vol. 1, No. 1, 1983, pp. 17–41.

[2] Crandall, M. G. and Lions, P.-L., "Viscosity solutions of Hamilton-Jacobi equations," *Transactions of the American Mathematical Society*, Vol. 277, No. 1, 1983, pp. 1–42.

[3] Crandall, M. G., Evans, L. C., and Lions, P.-L., "Some properties of viscosity solutions of Hamilton-Jacobi equations," *Transactions of the American Mathematical Society*, Vol. 282, No. 2, 1984, pp. 487–502.

[4] Fleming, W. H. and Soner, H. M., *Controlled Markov processes and viscosity solutions*, Vol. 25, Springer Science & Business Media, 2006.

[5] Barles, G. and Rouy, E., "A strong comparison result for the Bellman equation arising in stochastic exit time control problems and its applications," *Communications in Partial Differential Equations*, Vol. 23, No. 11-12, 1998, pp. 1995–2033.

[6] Bayraktar, E., Song, Q., and Yang, J., "On the continuity of stochastic exit time control problems," *Stochastic Analysis and Applications*, Vol. 29, No. 1, 2010, pp. 48–60.

[7] Touzi, N., *Optimal stochastic control, stochastic target problems, and backward SDE*, Vol. 29, Springer Science & Business Media, 2012.

[8] Tang, S. and Zhang, F., "Path-dependent optimal stochastic control and viscosity solution of associated Bellman equations," *arXiv preprint arXiv:1210.2078*, 2012.

[9] Bokanowski, O., Picarelli, A., and Zidani, H., "Dynamic programming and error estimates for stochastic control problems with maximum cost," *Applied Mathematics & Optimization*, Vol. 71, No. 1, 2015, pp. 125–163.

[10] Buckdahn, R. and Nie, T., "Generalized Hamilton-Jacobi-Bellman equations with Dirichlet boundary condition and stochastic exit time optimal control problem," *SIAM Journal on Control and Optimization*, Vol. 54, No. 2, 2016, pp. 602–631.

[11] Bardi, M. and Capuzzo-Dolcetta, I., *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, Springer Science & Business Media, 2008.

[12] Soner, H. M., "Optimal control with state-space constraint I," *SIAM Journal on Control and Optimization*, Vol. 24, No. 3, 1986, pp. 552–561.

[13] Barles, G. and Perthame, B., "Discontinuous solutions of deterministic optimal stopping time problems," *ESAIM: Mathematical Modelling and Numerical Analysis*, Vol. 21, No. 4, 1987, pp. 557–579.

[14] Barles, G. and Perthame, B., "Exit time problems in optimal control and vanishing viscosity method," *SIAM Journal on Control and Optimization*, Vol. 26, No. 5, 1988, pp. 1133–1148.

[15] Barles, G., "Discontinuous viscosity solutions of first-order Hamilton-Jacobi equations: a guided visit," *Nonlinear Analysis: Theory, Methods & Applications*, Vol. 20, No. 9, 1993, pp. 1123–1134.

[16] Blanc, A.-P., "Deterministic exit time control problems with discontinuous exit costs," *SIAM journal on control and optimization*, Vol. 35, No. 2, 1997, pp. 399–434.

[17] Malisoff, M., "Viscosity solutions of the Bellman equation for exit time optimal control problems with non-Lipschitz dynamics," *ESAIM: Control, Optimisation and Calculus of Variations*, Vol. 6, 2001, pp. 415–441.

[18] Malisoff, M., "Viscosity solutions of the Bellman equation for exit time optimal control problems with vanishing Lagrangians," *SIAM Journal on Control and Optimization*, Vol. 40, No. 5, 2002, pp. 1358–1383.

[19] Malisoff, M., "Further results on the Bellman equation for optimal control problems with exit times and nonnegative Lagrangians," *Systems & control letters*, Vol. 50, No. 1, 2003, pp. 65–79.

[20] Sinestrari, C., "Semiconcavity of the value function for exit time problems with nonsmooth target," *Commun. Pure Appl. Anal*, Vol. 3, No. 4, 2004, pp. 757–774.

[21] Motta, M. and Sartori, C., "The value function of an asymptotic exit-time optimal control problem," *Nonlinear Differential Equations and Applications NoDEA*, Vol. 22, No. 1, 2015, pp. 21–44.

[22] Cannarsa, P., Pignotti, C., and Sinestrari, C., "Semiconcavity for optimal control problems with exit time," *Discrete and Continuous Dynamical Systems*, Vol. 6, No. 4, 2000, pp. 975–997.

[23] Rungger, M. and Stursberg, O., "Continuity of the value function for exit time optimal control problems of hybrid systems," *Decision and Control (CDC), 2010 49th IEEE Conference on*, 2010, pp. 4210–4215.

[24] Kolmanovsky, I. and Maizenberg, T. L., "Optimal containment control for a class of stochastic systems perturbed by Poisson and Wiener processes," *American Control Conference, 2002. Proceedings of the 2002*, Vol. 1, 2002, pp. 322–327.

[25] Clark, J. and Vinter, R., "Stochastic exit time problems arising in process control," *Stochastics An International Journal of Probability and Stochastic Processes*, Vol. 84, No. 5-6, 2012, pp. 667–681.

[26] Kushner, H. J., *Probability methods for approximations in stochastic control and for elliptic equations*, Vol. 129, Academic Press, 1977.

[27] Kushner, H. and Dupuis, P. G., *Numerical methods for stochastic control problems in continuous time*, Vol. 24, Springer Science & Business Media, 2013.

[28] Rungger, M. and Stursberg, O., "A numerical method for hybrid optimal control based on dynamic programming," *Nonlinear Analysis: Hybrid Systems*, Vol. 5, No. 2, 2011, pp. 254–274.

[29] Bertsekas, D. P., *Dynamic Programming and Optimal Control, Vol. I*, Athena Scientific, Belmont, MA, 2005.

[30] Barles, G. and Souganidis, P. E., "Convergence of approximation schemes for fully nonlinear second order equations," *Asymptotic analysis*, Vol. 4, No. 3, 1991, pp. 271–283.

[31] Beylkin, G. and Mohlenkamp, M. J., "Algorithms for numerical analysis in high dimensions," *SIAM Journal on Scientific Computing*, Vol. 26, No. 6, 2005, pp. 2133–2159.

[32] Horowitz, M. B., Damle, A., and Burdick, J. W., "Linear Hamilton Jacobi Bellman equations in high dimensions," *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, 2014, pp. 5880–5887.

[33] Gorodetsky, A. A., Karaman, S., and Marzouk, Y. M., "Efficient high-dimensional stochastic optimal motion control using tensor-train decomposition," *Robotics: Science and Systems*, 2015.

[34] Raffard, R. L., Hu, J., and Tomlin, C., "Adjoint-based optimal control of the expected exit time for stochastic hybrid systems," *Lecture Notes in Computer Science*, Vol. 3414, 2005, pp. 557–572.

[35] Kolmanovsky, I. V., Lezhnev, L., and Maizenberg, T. L., "Discrete-time drift counteraction stochastic optimal control: theory and application-motivated examples," *Automatica*, Vol. 44, No. 1, 2008, pp. 177–184.

[36] Kolmanovsky, I. V., Sun, J., and Sivashankar, S. N., "An integrated software environment for powertrain feasibility assessment using optimization and optimal control," *Asian Journal of Control*, Vol. 8, No. 3, 2006, pp. 199–209.

[37] Kolmanovsky, I. V. and Filev, D. P., "Stochastic optimal control of systems with soft constraints and opportunities for automotive applications," *IEEE Control Applications (CCA) & Intelligent Control (ISIC)*, 2009, pp. 1265–1270.

[38] Balasubramanian, K. and Kolmanovsky, I. V., "Range maximization of a direct methanol fuel cell powered mini air vehicle using stochastic drift counteraction optimal control," *Proceedings of the 2012 American Control Conference (ACC)*, Montreal, Canada, June 2012.

[39] Kolmanovsky, I. V. and Menezes, A. A., "A stochastic drift counteraction optimal control approach to glider flight management," *Proceedings of the 2011 American Control Conference*, 2011, pp. 1009–1014.

[40] Menezes, A. A., Shah, D. D., and Kolmanovsky, I. V., "An evaluation of stochastic model-dependent and model-independent glider flight management," *IEEE Transactions on Control Systems Technology*, Vol. PP, No. 99, 2017, pp. 1–17.

[41] Zidek, R. A. E. and Kolmanovsky, I. V., "Drift counteraction optimal control for deterministic systems and enhancing convergence of value iteration," *Automatica*, Vol. 83, 2017, pp. 108–115.

[42] Zidek, R. A. E., Kolmanovsky, I. V., and Bemporad, A., "Spacecraft drift counteraction optimal control: open-loop and receding horizon solutions," *Journal of Guidance, Control, and Dynamics*, 2017, under review.

[43] Zidek, R. A. E. and Kolmanovsky, I. V., "Deterministic drift counteraction optimal control and its application to satellite life extension," *54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 3397–3402.

[44] Zidek, R. A. E. and Kolmanovsky, I. V., "Deterministic drift counteraction optimal control for attitude control of spacecraft with time-varying mass," *AIAA Guidance, Navigation, and Control Conference*, 2016, p. 0369.

[45] Zidek, R. A. E. and Kolmanovsky, I. V., "Geostationary satellite station keeping using drift counteraction optimal control," *26th AAS/AIAA Space Flight Mechanics Meeting*, 2016, pp. 989–1000.

[46] Zidek, R. A. E. and Kolmanovsky, I. V., "Stochastic drift counteraction optimal control and enhancing convergence of value iteration," *55th IEEE Conference on Decision and Control (CDC)*, 2016, pp. 1119–1124.

[47] Zidek, R. A. E., Petersen, C. D., Bemporad, A., and Kolmanovsky, I. V., "Receding horizon drift counteraction and its application to spacecraft attitude control," *27th AAS/AIAA Space flight mechanics meeting*, 2017, pp. AAS 17–465.

[48] Zidek, R. A. E., Bemporad, A., and Kolmanovsky, I. V., "Optimal and receding horizon drift counteraction control: linear programming approaches," *American Control Conference (ACC)*, 2017, pp. 2636–2641.

[49] Zidek, R. A. E. and Kolmanovsky, I. V., "A new algorithm for a class of deterministic drift counteraction optimal control problems," *American Control Conference (ACC)*, 2017, pp. 623–629.

[50] Zidek, R. A. E. and Kolmanovsky, I. V., "Optimal driving policies for autonomous vehicles based on stochastic drift counteraction," *20th IFAC World Congress*, 2017, pp. 292–298.

[51] Zidek, R. A. E., Kolmanovsky, I. V., and Bemporad, A., "Stochastic MPC approach to drift counteraction," *American Control Conference (ACC)*, 2018, under review.

[52] Zidek, R. A. E. and Kolmanovsky, I. V., "Approximate closed-form solution to a linear quadratic optimal control problem with disturbance," *Journal of Guidance, Control, and Dynamics*, Vol. 40, 2017, pp. 477–483.

[53] Yeh, J., *Real analysis: theory of measure and integration*, Vol. 2, World Scientific, Singapore, 2006.

[54] Rudin, W., *Principles of mathematical analysis*, Vol. 3, McGraw-Hill, New York, 1964.

[55] Sundaram, R. K., *A first course in optimization theory*, Cambridge University Press, Cambridge, UK, 1996.

[56] Dontchev, A. L. and Zolezzi, T., *Well-posed optimization problems*, Springer, 1993.

[57] Bertsekas, D. P. and Tsitsiklis, J. N., "Neuro-dynamic programming: an overview," *Decision and Control (CDC), Proceedings of the 34th IEEE Conference on*, Vol. 1, 1995, pp. 560–564.

[58] Werbos, P. J., "Reinforcement learning and approximate dynamic programming (RLADP) – foundations, common misconceptions, and the challenges ahead," *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, 2012, pp. 1–30.

[59] Heydari, A., "Revisiting approximate dynamic programming and its convergence," *IEEE Transactions on Cybernetics*, Vol. 44, No. 12, 2014, pp. 2733–2743.

[60] Wei, Q., Liu, D., and Lin, H., "Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems," *IEEE Transactions on Cybernetics*, Vol. 46, No. 3, 2016, pp. 840–853.

[61] Matheron, G., "Principles of geostatistics," *Economic geology*, Vol. 58, No. 8, 1963, pp. 1246–1266.

[62] Deisenroth, M. P., Rasmussen, C. E., and Peters, J., "Gaussian process dynamic programming," *Neurocomputing*, Vol. 72, No. 7, 2009, pp. 1508–1524.

[63] Bellman, R., "The theory of dynamic programming," Tech. rep., DTIC Document, 1954.

[64] Bate, R. B., Mueller, D. D., and White, J. E., *Fundamentals of astrodynamics*, Dover Publications, New York, NY, 1971.

[65] Wie, B., *Space vehicle dynamics and control*, American Institute of Aeronautics and Astronautics, Reston, VA, 2008.

[66] Schmuland, D. T., Masse, R. K., and Sota, C. G., "Hydrazine propulsion module for CubeSats," *25th Annual AIAA/USU Conference on Small Satellites*, Logan, UT, 2011.

[67] Schmuland, D. T., Carpenter, C., and Masse, R. K., "Mission applications of the MRS-142 CubeSat high-impulse adaptable monopropellant propulsion system (CHAMPS)," *48th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, Atlanta, GA, 2012.

[68] Picone, J. M., Hedin, A. E., Drob, D. P., and Aikin, A. C., "NRLMSISE-00 empirical model of the atmosphere: statistical comparisons and scientific issues," *Journal of Geophysical Research: Space Physics*, Vol. 107, No. A12, 2002.

[69] Lophaven, S. N., Nielsen, H. B., and Søndergaard, J., "DACE-A Matlab Kriging toolbox, version 2.0," Tech. rep., 2002.

[70] Soop, E. M., *Handbook of geostationary orbits*, Vol. 3, Springer Science & Business Media, 1994.

[71] Pocha, J. J., *An introduction to mission design for geostationary satellites*, Vol. 1, Springer Science & Business Media, 2012.

[72] Tsiotras, P. and Longuski, J. M., "A new parameterization of the attitude kinematics," *Journal of the Astronautical Sciences*, Vol. 43, No. 3, 2008, pp. 243–262.

[73] Huzel, D. K. and Huang, D. H., "Design of liquid propellant rocket engines," Tech. Rep. NASA SP-125, Rocketdyne Division, North American Aviation, Inc., Washington, D.C., 1967.

[74] Sutton, G. P., *History of Liquid Propellant Rocket Engines*, American Institute of Aeronautics and Astronautics, Inc., Reston, VA, 2006.

[75] Bemporad, A. and Morari, M., "Control of systems integrating logic, dynamics, and constraints," *Automatica*, Vol. 35, No. 3, 1999, pp. 407–427.

[76] Bertsimas, D. and Tsitsiklis, J. N., *Introduction to linear optimization*, Vol. 6, Athena Scientific Belmont, MA, 1997.

[77] Richards, A. and How, J., "Mixed-integer programming for control," *Proceedings of the 2005, American Control Conference, 2005.*, 2005, pp. 2676–2683.

[78] Bemporad, A., "Hybrid toolbox - user's guide," 2004, http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox.

[79] Petersen, C. D., Leve, F., Flynn, M., and Kolmanovsky, I., "Recovering linear controllability of an underactuated spacecraft by exploiting solar radiation pressure," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 4, 2015, pp. 826–837.

[80] Weiss, A., Kalabić, U., and Di Cairano, S., "Model predictive control for simultaneous station keeping and momentum management of low-thrust satellites," *2015 American Control Conference (ACC)*, 2015, pp. 2305–2310.

[81] Romero, P. and Gambi, J. M., "Optimal control in the east/west station-keeping manoeuvres for geostationary satellites," *Aerospace Science and Technology*, Vol. 8, No. 8, 2004, pp. 729–734.

[82] Romero, P., Gambi, J. M., and Patiño, E., "Stationkeeping manoeuvres for geostationary satellites using feedback control techniques," *Aerospace Science and Technology*, Vol. 11, No. 2, 2007, pp. 229–237.

[83] Lee, B.-S., Hwang, Y., Kim, H.-Y., and Park, S., "East-west station-keeping maneuver strategy for COMS satellite using iterative process," *Advances in Space Research*, Vol. 47, No. 1, 2011, pp. 149–159.

[84] de Bruijn, F. J., Theil, S., Choukroun, D., and Gill, E., "Geostationary satellite station-keeping using convex optimization," *Journal of Guidance, Control, and Dynamics*, , No. null, 2015, pp. 605–616.

[85] Crouch, P. E., "Spacecraft attitude control and stabilization," *IEEE Transactions on Automatic Control*, Vol. AC-29, No. 4, 1984, pp. 321–331.

[86] Tsiotras, P. and Luo, J., "Control of underactuated spacecraft with bounded inputs," *Automatica*, Vol. 36, No. 8, 2000, pp. 1153–1169.

[87] Zavoli, A., De Matteis, G., Giulietti, F., and Avanzini, G., "Single-axis pointing of an underactuated spacecraft equipped with two reaction wheels," *Journal of Guidance, Control, and Dynamics*, 2017.

[88] Larson, K. A., McCalmont, K. M., Peterson, C. A., and Ross, S. E., "Kepler mission operations response to wheel anomalies," *SpaceOps 2014 Conference*, 2014, p. 1882.

[89] Van Cleve, J. E., Howell, S. B., Smith, J. C., Clarke, B. D., Thompson, S. E., Bryson, S. T., Lund, M. N., Handberg, R., and Chaplin, W. J., "That's how we roll: the NASA K2 mission science products and their performance metrics," *Publications of the Astronomical Society of the Pacific*, Vol. 128, No. 965, 2016, pp. 075002.

[90] Martinez-Sanchez, M. and Pollard, J. E., "Spacecraft electric propulsion – an overview," *Journal of Propulsion and Power*, Vol. 14, No. 5, 1998, pp. 688–699.

[91] Goebel, D., Martinez-Lavin, M., Bond, T., and King, A., "Performance of XIPS electric propulsion in on-orbit station keeping of the Boeing 702 spacecraft," *38th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 2002, p. 4348.

[92] Earl, M. G. and D'andrea, R., "Iterative MILP methods for vehicle-control problems," *IEEE Transactions on Robotics*, Vol. 21, No. 6, 2005, pp. 1158–1167.

[93] Puterman, M. L., *Markov decision processes: discrete stochastic dynamic programming*, John Wiley & Sons, 2014.

[94] Thomson, B. S., Bruckner, J. B., and Bruckner, A. M., *Elementary real analysis*, www.classicalrealanalysis.com, 2008.

[95] Hatipoglu, C., Ozguner, U., and Redmill, K. A., "Automated lane change controller design," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 4, No. 1, 2003, pp. 13–22.

[96] Guo, L., Ge, P.-S., Yue, M., and Zhao, Y.-B., "Lane changing trajectory planning and tracking controller design for intelligent vehicle running on curved road," *Mathematical Problems in Engineering*, Vol. 2014, 2014.

[97] Hu, C., Jing, H., Wang, R., Yan, F., and Chadli, M., "Robust H_∞ output-feedback control for path following of autonomous ground vehicles," *Mechanical Systems and Signal Processing*, Vol. 70, 2016, pp. 414–427.

[98] Worrall, R., Bullen, A., and Gur, Y., "An elementary stochastic model of lane-changing on a multilane highway," *Highway Research Record*, , No. 308, 1970.

[99] Gipps, P. G., "A model for the structure of lane-changing decisions," *Transportation Research Part B: Methodological*, Vol. 20, No. 5, 1986, pp. 403–414.

[100] Wu, J., Brackstone, M., and McDonald, M., "Fuzzy sets and systems for a motorway microscopic simulation model," *Fuzzy Sets and Systems*, Vol. 116, No. 1, 2000, pp. 65–76.

[101] Toledo, T., Koutsopoulos, H. N., and Ben-Akiva, M., "Integrated driving behavior modeling," *Transportation Research Part C: Emerging Technologies*, Vol. 15, No. 2, 2007, pp. 96–112.

[102] Schubert, R., Schulze, K., and Wanielik, G., "Situation assessment for automatic lane-change maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 11, No. 3, 2010, pp. 607–616.

[103] Wang, M., Hoogendoorn, S. P., Daamen, W., van Arem, B., and Happee, R., "Game theoretic approach for predictive lane-changing and car-following control," *Transportation Research Part C: Emerging Technologies*, Vol. 58, 2015, pp. 73–92.

[104] Hertz, J., Krogh, A., and Palmer, R. G., *Introduction to the theory of neural computation*, Vol. 1, Basic Books, 1991.

[105] Oyler, D. W., Yildiz, Y., Girard, A. R., Li, N. I., and Kolmanovsky, I. V., "A game theoretical model of traffic with multiple interacting drivers for use in autonomous vehicle development," *American Control Conference (ACC), 2016*, 2016, pp. 1705–1710.

[106] Bernardini, D. and Bemporad, A., "Stabilizing model predictive control of stochastic constrained linear systems," *IEEE Transactions on Automatic Control*, Vol. 57, No. 6, 2012, pp. 1468–1480.

[107] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., and Hirukawa, H., "Biped walking pattern generation by using preview control of zero-moment point," *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, Vol. 2, 2003, pp. 1620–1626.

[108] Takaba, K., "A tutorial on preview control systems," *SICE 2003 Annual Conference*, Vol. 2, 2003, pp. 1388–1393.

[109] De Bruyne, S., Van der Auweraer, H., Anthonis, J., Desmet, W., and Swevers, J., "Preview control of a constrained hydraulic active suspension system," *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, 2012, pp. 4400–4405.

[110] Li, Z., Kolmanovsky, I., Atkins, E., Lu, J., Filev, D., and Michelini, J., "Cloud aided semi-active suspension control," *Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2014 IEEE Symposium on*, 2014, pp. 76–83.

[111] Moran, A., Mikami, Y., and Hayase, M., "Analysis and design of H_∞ preview tracking control systems," *4th International Workshop on Advanced Motion Control (AMC)*, Vol. 2, 1996, pp. 482–487.

[112] Mianzo, L. and Peng, H., "A unified framework for LQ and H_∞ preview control algorithms," *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 1998, pp. 2816–2821.

[113] Mehra, R. K., Amin, J. N., Hedrick, K. J., Osorio, C., and Gopalasamy, S., "Active suspension using preview information and model predictive control," *Control Applications. Proceedings of the IEEE International Conference on*, 1997, pp. 860–865.

[114] Cole, D., Pick, A., and Odhams, A., "Predictive and linear quadratic methods for potential application to modelling driver steering control," *Vehicle System Dynamics*, Vol. 44, No. 3, 2006, pp. 259–284.

[115] Laks, J., Pao, L. Y., Simley, E., Wright, A., Kelley, N., and Jonkman, B., "Model predictive control using preview measurements from lidar," *49th AIAA aerospace sciences meeting*, 2011, pp. 2011–0813.

[116] Spencer, M. D., Stol, K. A., Unsworth, C. P., Cater, J. E., and Norris, S. E., "Model predictive control of a wind turbine using short-term wind field predictions," *Wind Energy*, Vol. 16, No. 3, 2013, pp. 417–434.

[117] Calafiore, G. C. and Fagiano, L., "Robust model predictive control via scenario optimization," *IEEE Transactions on Automatic Control*, Vol. 58, No. 1, 2013, pp. 219–224.

[118] Zidek, R. A. E. and Kolmanovsky, I. V., "Approximate optimal control of nonlinear systems with quadratic performance criteria," *2015 American Control Conference (ACC)*, 2015, pp. 5587–5592.

[119] Bernstein, D. S., *Matrix mathematics*, Princeton University Press, Princeton, NJ, 2009.

[120] Khalil, H. K., *Nonlinear systems*, Prentice Hall, 3rd ed., 2002.

[121] Longuski, J. M., Guzman, J. J., and Prussing, J. E., *Optimal control with aerospace applications*, Springer, 2014.

[122] Blanchini, F. and Miani, S., *Set-theoretic methods in control*, Springer, 2008.

[123] Bernstein, D. S., *Geometry, kinematics, statics, and dynamics*, Princeton University Press, Princeton and Oxford, 2012.

[124] Hughes, P. C., *Spacecraft attitude dynamics*, Courier Corporation, 2004.

[125] Linares, R., Jah, M. K., Crassidis, J. L., Leve, F. A., and Kelecy, T., "Astrometric and photometric data fusion for inactive space object mass and area estimation," *Acta Astronautica*, Vol. 99, 2014, pp. 1–15.