Robust Model Predictive Control with a Reactive Safety Mode

Thesis by

John M. Carson III

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

California Institute of Technology

Pasadena, California

2008

(Defended April 17, 2008)

"A creative man is motivated by the desire to achieve, not by the desire to beat others."

Ayn Rand

Acknowledgements

My research and dissertation are the culmination of a delightful and fortunate collaboration with my advisor Prof. Richard M. Murray and co-advisors Dr. Douglas G. MacMynowski and Dr. Behçet Açıkmeşe. I express my sincerest thanks to Richard for his guidance and support during my graduate studies at Caltech and throughout my many research interests. His ability to dig into the minute details of theoretical concepts while simultaneously maintaining a broader perspective on research applicability to engineering systems made him both a superb advisor and an inspirational mentor. I am also grateful to Doug for his co-advisement and friendship throughout my various research endeavors. His keen intuition and attention to detail ushered me through countless research conundrums that arose with relating mathematical nuances to engineering reality. My heartfelt appreciation goes to Behçet for selflessly serving as my technical advisor during my thesis research. His incredible mathematical insight and mentorship throughout the development of my thesis research both matured my own mathematical capabilities but also propelled my research well beyond my expectations.

I would also like to acknowledge both Dr. Fred Y. Hadaegh and Dr. David S. Bayard from the NASA/Caltech Jet Propulsion Laboratory for supporting me through a part-time research position at JPL throughout my graduate studies. The exposure to world-class researchers and real-world applications significantly shaped my approach to research and provided me insight into important, contemporary engineering problems.

Most importantly, I want to thank my family. I am enormously grateful for the love, support, and encouragement that my wife Vanessa has given me through my studies at Caltech, and I am beholden and deeply appreciative to my parents John and Janet for their life-long encouragement and unequivocal support of my interests.

Abstract

Control algorithms suitable for online implementation in engineering applications, such as aerospace and mechanical vehicles, often require adherence to physical state and control constraints. Additionally, the chosen algorithms must provide robustness to uncertainty affecting both the system dynamics and the constraints. As further autonomy is built into these systems, the algorithms must be capable of blending multiple operational modes without violating the intrinsic constraints. Further, for real-time applications, the implemented control algorithms must be computationally efficient and reliable. The research in this thesis approaches these application needs by building upon the framework of MPC (Model Predictive Control).

The MPC algorithm makes use of a *nominal* dynamics model to predict and optimize the response of a system under the application of a feedforward control policy, which is computed online in a finite-horizon optimization problem. The MPC algorithm is quite general and can be applied to linear and nonlinear systems and include explicit state and control constraints. The finite-horizon optimization is advantageous given the finite online computational capabilities in practical applications. Further, recursively re-solving the finite-horizon optimization in a compressing- or receding-horizon manner provides a form of closed-loop control that updates the feedforward control policy by setting the nominal state at re-solve to the current *actual* state. However, uncertainty between the nominal model and the actual system dynamics, along with constraint uncertainty can cause feasibility, and hence, robustness issues with the traditional MPC algorithm.

In this thesis, an R-MPC (Robust and re-solvable MPC) algorithm is developed for uncertain nonlinear systems to address uncertainty affecting the dynamics. The R-MPC control policy consists of two components: the feedforward component that is solved online as in traditional MPC; and a separate feedback component that is determined offline, based on a characterization of the uncertainty between the nominal model and actual system.

The addition of the feedback policy generates an invariant tube that ensures the actual system trajectories remain in the proximity of the nominal feedforward trajectory for all time. Further, this tube provides a means to theoretically guarantee continued feasibility and thus re-solvability of the R-MPC algorithm, both of which are required to guarantee asymptotic stability.

To address uncertainty affecting the state constraints, an SR-MPC (Safety-mode augmented R-MPC) algorithm is developed that blends a reactive safety mode with the R-MPC algorithm for uncertain nonlinear systems. The SR-MPC algorithm has two separate operational modes: *standard* mode implements a modified version of the R-MPC algorithm to ensure asymptotic convergence to the origin; *safety* mode, if activated, guarantees containment within an invariant set about a safety reference for all time. The standard mode modifies the R-MPC algorithm with a special constraint to ensure safety-mode availability at any time. The safety-mode control is provided by an offline designed control policy that can be activated at any time during standard mode. The separate, reactive safety mode provides robustness to unexpected state-constraint changes; e.g., other vehicles crossing/stopping in the feasible path, or unexpected ground proximity in landing scenarios.

Explicit design methods are provided for implementation of the R-MPC and SR-MPC algorithms on a class of systems with uncertain nonlinear terms that have norm-bounded derivatives. Further, a discrete SR-MPC algorithm is developed that is more broadly applicable to real engineering systems. The discrete algorithm is formulated as a second-order cone program that can be solved online in a computationally efficient manner by using interior-point algorithms, which provide convergence guarantees in finite time to a prescribed level of accuracy. This discrete SR-MPC algorithm is demonstrated in simulation of a spacecraft descent toward a small asteroid where there is an uncertain gravity model, as well as errors in the expected surface altitude. Further, realistic effects such as control-input uncertainty, sensor noise, and unknown disturbances are included to further demonstrate the applicability of the discrete SR-MPC algorithm in a realistic implementation.

Contents

List of Figures

Chapter 1

Introduction

The proliferation of autonomy and online decision-making capabilities in engineering systems, such as robotic ground or air vehicles, has paralleled the advancements of contemporary computation, sensing, and communication resources. What makes possible the coalescence of these capabilities and resources? Algorithms. Fundamental research and development in algorithms has enabled real-time, online capabilities such as path planning (guidance), sensor fusion, situational awareness, onboard network communication, and many other tasks that push the limits of contemporary engineering systems. The research in this thesis focuses on general control algorithms, which include path planning, that are suitable for online implementation.

Pushing control policy design online provides the benefit of using real-time sensor information and onboard situational awareness to provide improvements to system path planning and operation. Many practical engineering systems are nonlinear, subject to both state and control constraints, and can have several operational modes and control objectives. Nonlinearities arise in the system dynamics, as well as in the physical limitations of real systems (e.g., actuator stroke/throw, maximum torque, temperature bounds, physical barriers). Control algorithms utilize models for these nonlinearities and are inherently flawed by uncertainty in both the actual system dynamics and the constraints. Online re-planning of control policies can mitigate some of this uncertainty by updating the guidance and control policies, and even control objectives, based on actual system information not available to offline designs. These updates can increase efficiency of operation as well as mitigate risks (provide safety) from unanticipated operational changes.

The complexity of online algorithms is subject to the computational limitations of contemporary processors. Online algorithms must be sized appropriately so that available

computational resources can generate any guidance or control policy updates at a rate appropriate to the operational mode and system dynamics. Additionally, online algorithms must be robust to the aforementioned uncertainties in actual systems and be capable of providing feasible control solutions given those uncertainties and the need for safety/risk mitigation.

The development of general control algorithms that can address the needs of practical engineering applications requires a framework that is computationally efficient, handles a diversity of dynamics, incorporates constraints, blends operational modes, and considers control objectives. The framework of model predictive control provides a suitable candidate.

1.1 The Framework of Model Predictive Control

The framework of MPC (Model Predictive Control) computes control inputs through online solution of an FHC (Finite-Horizon optimal Control problem) that can enforce explicit state and control constraints. The FHC utilizes a *nominal* system model to predict the response of the *actual* system dynamics to the control inputs over the finite planning horizon, hence the origin of the MPC algorithm name. In MPC, the computed control is applied to the actual system in a feedforward (i.e., open-loop) manner over a specified time interval (shorter than the FHC planning horizon), followed by a re-solve (re-computation) of the FHC with the initial nominal state at the re-solve time set to the current state of the actual system. Recursively re-solving the FHC in this manner provides a form of closed-loop feedback for the MPC algorithm by incorporating current actual state measurements for computing current feedforward inputs.

The FHC re-solves are typically performed in a receding-horizon manner, which is why MPC is also commonly referred to as RHC (Receding Horizon Control). In RHC, the finite planning horizon is receded forward in time at each FHC re-solve; the planning horizon is often of fixed length in RHC. In another variant of MPC called compressing-horizon control, the planning horizon is shortened at each re-solve by the change in time since the prior FHC solution; thus, the total application time of MPC never exceeds the length of the planning horizon from the first FHC solution.

The framework for MPC is quite general and can be applied to linear and nonlinear systems. The MPC algorithm is beneficial in applications where online computation of

control inputs is necessary; for instance, in applications with evolving control objectives or applications where state or control constraints might change based on current operating conditions (i.e., location in state space). Additionally, the finite planning horizon in the FHC is beneficial in many practical applications given the limitations of online computation. However, uncertainty between the nominal model and the actual system dynamics, along with constraint uncertainty can cause feasibility issues during FHC re-solves and thus robustness problems in the practical application of MPC. The research contributions in this thesis address these issues.

1.2 A Summary of the MPC Literature

The generality of MPC comes from several decades of research and development of the framework. The literature on MPC is quite extensive, and the review herein is by no means exhaustive. Many excellent references are omitted for brevity. Fortunately, survey papers on MPC highlight many of the contributions; see for example Mayne *et al.* [31] and Garcia and Morari [17]. The intent of this review is to familiarize the reader with some of the past contributions to the theoretical framework for MPC to both clarify the concept of MPC and to put into context the theoretical framework presented in Chapter 2.

The origin of MPC was in engineering applications from the chemical and process control industry where system dynamics are nonlinear and subject to both state and control constraints; the papers by Qin and Badgwell [38] and Richalet [42] provide some examples of these applications. At inception, no theoretical frameworks were available to suggest MPC was a stable control method. However, tuning of the control policies for the slow dynamics of these applications demonstrated empirically that the MPC framework was indeed stable.

Initial proofs of stability for the MPC algorithm focused on unconstrained continuous-time systems, except for the terminal state fixed at the origin. The dynamics were assumed linear and perfectly known (i.e., the nominal model used in the FHC was identical to the actual dynamics being controlled). One of the early stability results for MPC was formulated by Thomas [49] for linear time-invariant systems with an FHC consisting of a quadratic cost on the control input, along with the fixed terminal state constraint. A fixed finite-time horizon allowed for an a priori computation of a single, optimal gain that provided closed-loop asymptotic stability in application of MPC to these types of systems.

This early work was extended by Kwon and Pearson [26] to linear time-varying systems with the FHC incorporating a quadratic cost on both the state and control. The algorithm required backward integrating a time-varying Riccati equation over a finite interval, from the horizon length back to the current time, to generate controller gains for each MPC update. Extensions of this work by Kwon *et al.* [25] provided a method to update gains in a computationally-efficient manner that avoided re-integration of the Riccati equation.

Stability of MPC was extended to continuous-time, autonomous nonlinear systems by Chen and Shaw [12]. This research was also likely the earliest contribution to utilize the FHC cost function as a Lyapunov function [21] to prove stability of the MPC algorithm. Unrelated research by Mayne and Michalska [30] for continuous-time, autonomous nonlinear systems also utilized a Lyapunov approach to prove stability, along with an in-depth analysis of the continuity of the MPC control policy based on the necessary conditions for optimality from Pontryagin's Minimum Principal [22]. The research by Rawlings and Muske [41] established a stable MPC algorithm for discrete-time, linear time-invariant systems that could be solved online as a finite-dimension quadratic program, including both state and control constraints. Research by Keerthi and Gilbert [20] established the stability of MPC for discrete-time, time-varying nonlinear systems and also incorporated both discrete state and control constraints.

The early MPC stability results constrained the terminal state to the origin, which can cause computational difficultly in the FHC optimization with MPC applied to nonlinear systems. This issue was addressed with the dual-mode receding-horizon MPC developed by Michalska and Mayne [33]; this method has become a standard approach in many MPC algorithms. The method proposed a relaxation to the terminal equality constraint, enforcing instead an inequality constraint in the FHC that created a terminal set about the origin. The dual-mode approach is required to ensure stability of the algorithm with the relaxation. Outside the terminal set, control comes from the receding-horizon MPC solutions. Inside the terminal set, control is switched to a local linear feedback controller that stabilizes the nonlinear system in a neighborhood of the origin (the local linearized system must be stabilizable). Extensions of this work by Chen and Allgöwer [13] created a quasi-infinite horizon MPC where the FHC cost function adds a terminal cost, or cost-to-go, component that acts like the discarded portion of an infinite-horizon cost function. This terminal cost further acts as a CLF (Control Lyapunov Function [46, 47, 16]) for the nonlinear system

within the terminal set. This latter contribution is part of the baseline MPC algorithm defined in Chapter 2 of this thesis.

Since the computation of the MPC control policy relies on a nominal system model, the robustness of MPC to system uncertainties is non-trivial to establish. For an overview of several robust MPC contributions, refer to the survey paper by Jalali and Nadimi [19]. One approach to robustness is sufficiently reducing the time intervals for re-solving the FHC. The research by Michalska and Mayne [33] derived explicit upper bounds, which can be very conservative, for the re-solve time intervals. A game theoretic approach to robust MPC by Chen *et al.* [14] combines MPC with $\mathcal{H}_\infty$ control to formulate an open-loop min-max MPC method that applies to a class of continuous-time uncertain nonlinear systems with input constraints and exogenous disturbances. Research by Magni *et al.* [28] uses a similar game theoretic and min-max MPC approach for generating closed-loop $\mathcal{H}_\infty$ control laws for unconstrained, uncertain nonlinear systems. A drawback of these min-max methods is a significant online computational demand for re-solves.

For discrete-time, linear time-invariant systems, several robust MPC methods have been developed. Methods by Kothare *et al.* [23] design the control policy as a sequence of state-feedback control laws that at each discrete time step minimize a worst-case infinite-horizon objective function (as in min-max MPC) that includes state and control constraints. The optimization requires online solution of LMIs (Linear Matrix Inequalities), which can be computationally expensive for large-dimension problems. Separate research by Scokaert and Mayne [44] also used a feedback min-max MPC approach for developing robust control policies for discrete-time linear time-invariant systems subject to bounded disturbances and state and control constraints. An LMI-based robust MPC method by Smith [45] combines the benefits of feedforward- and feedback-based approaches to robust MPC. A simplification of the online MPC re-solves was provided in the research by Bemporad *et al.* [6, 7] for systems with linear state and linear control constraints. For this class of systems, the input control policy is shown to be a piecewise-affine and continuous function of the state, based on a polyhedral partition of the feasible state space.

Another concept for robust MPC involves generating invariant tubes about the feed-forward guidance policy to maintain the actual system in the proximity of the predicted nominal model response. A tube-based approach for linear discrete-time systems with bounded disturbances is described by Mayne and Langson [29] and Mayne, Seron, and

Raković [32]. Time-varying or parameter-uncertain linear discrete-time systems have also been considered by Langson *et al.* [27], and computational improvements and optimization of the invariant-tube cross section are presented by Raković and Mayne [40]. Extensions by Raković *et al.* [39] apply the tube-based methods to nonlinear discrete-time systems with matched nonlinearities, including some special classes of piecewise affine systems.

A common assumption made in asymptotic stability proofs for MPC is that the FHC terminal set contains the desired target state, which is usually assumed as the origin. Many practical applications have computational limitations or limited knowledge of nearby state constraints, which result in an FHC planning horizon that does not include the desired target state. Thus, the FHC terminal set does not contain the target state for the initial feasible solution. Subsequent FHC re-solves shift the terminal set toward the desired target state, but there is no mathematical guarantee that the FHC will remain feasible. To address the potential for an infeasible re-solve, research by Schouwenaars *et al.* [43] for discrete, linear time-invariant systems imposes a constraint in the FHC that ensures each feasible solution ends in a terminal safety set, which is a control-invariant set. If a subsequent re-solve is infeasible, the prior feasible solution allows entry into the terminal safety set. This method is enhanced in the RSBK (Robust Safe But Knowledgeable) algorithm by Kuwata *et al.* [24]. These algorithms assume perfect state-constraint knowledge during the current planning horizon, with the safety mode added to address feasibility issues of FHC re-solves for a shifted terminal set.

1.3 Thesis Contributions and Organization

The contributions in this thesis are motivated by vehicle control applications requiring efficient, online guidance and control algorithms capable of robustly maneuvering a system toward a desired state while simultaneously providing safety from uncertainty in state-constraint knowledge (e.g., safety from other vehicles unexpectedly blocking the feasible path or unexpected ground proximity during landing).

A baseline MPC algorithm is presented in Chapter 2 based on traditional methods outlined in the extensive MPC literature. This baseline algorithm is valid for general nonlinear systems and enforces both state and control constraints in the FHC. Further, a terminal set is enforced rather than a fixed terminal-state constraint. The contributions in this thesis

build upon this baseline MPC algorithm.

A robust MPC algorithm is developed in Chapter 3 for continuous-time uncertain nonlinear systems. This R-MPC (Robust and re-solvable MPC) algorithm generates online a nominal feedforward control policy based on a modified FHC. An additional offline-designed feedback policy is added in R-MPC to generate an invariant tube that ensures the actual states remain in the proximity of the nominal feedforward trajectory. The tube provides an explicit characterization of the R-MPC robustness, which can accommodate uncertainties and disturbances in the actual dynamics up to the level considered in the feedback policy design. An important contribution of the R-MPC algorithm is the modified FHC, which puts a relaxation on the initial nominal state for re-solves. The initial nominal state is allowed to be within the invariant tube rather than fixed to the actual state, as in the baseline MPC; this relaxation guarantees continued FHC feasibility during re-solves and leads to robust asymptotic stability of the R-MPC algorithm. The R-MPC algorithm places no requirements on the re-solve rate, which is useful in online applications with computational limitations. Additionally, specific design methods are provided for a class of continuous-time systems with uncertain nonlinear terms that have norm-bounded derivatives. The R-MPC algorithm development was joint work performed with Behçet Açıkmeşe [1].

The R-MPC algorithm is modified and augmented with a separate, reactive safety mode in Chapter 4 to handle uncertainty in the state constraints during the planning horizon. The SR-MPC (Safety-mode augmented R-MPC) algorithm has two operational modes: *standard* mode and *safety* mode. The standard mode implements a modified R-MPC algorithm that is executed as long as changes in state constraints do not violate the FHC feasibility or system safety. The safety mode consists of an offline-designed control policy that, if needed, can maintain the system in an invariant safety set. The R-MPC modification enforces an additional FHC constraint to ensure safety-mode availability at any time along the standard-mode planning horizon. This is a significant contribution that also blends the two operational modes, guaranteeing a mode switch from standard to safety mode will not violate prescribed state or control constraints. From the safety mode, a higher-level decision algorithm (which is not a part of this thesis research) would need to establish a new initial feasible solution, if one exists, or a new control objective to restart standard mode. The SR-MPC algorithm is applicable to systems with static state constraints that might change after initial feasibility is established for the standard mode. For example, a vehicle can be

maintained in safety mode if an object or other vehicle crosses/stops in the feasible path (outside the invariant safety set), or a spacecraft can be held in safety mode if sensors detect unexpected proximity (low altitude) relative to the ground. Specific design methods are provided for the class of continuous-time systems with uncertain nonlinear terms that have norm-bounded derivatives.

Chapter 5 develops a discrete implementation of the SR-MPC algorithm targeted to applications such as mechanical or aerospace vehicles. The significant contributions are the development of discrete versions of the continuous-time algorithms from Chapter 3 and 4. Additionally, the discrete algorithm is formulated as a second-order cone program [9] that ensures constraint satisfaction and can be solved online in a computationally efficient manner by using interior-point algorithms, which provide convergence guarantees to within a specified accuracy in a finite number of steps [36, 48, 50]. The discrete SR-MPC algorithm maintains the ability to activate the safety mode at any discrete time along the planning horizon, providing safety to uncertain state constraints along the entire planning horizon. This is a contribution that adds to similar methods that incorporate safety only at the end of the planning horizon and assume perfect state-constraint knowledge along the planning horizon. The feedback policy design and portions of the discrete FHC development were joint work with Behçet Açıkmeşe [10].

The discrete SR-MPC algorithm is applied to a detailed engineering example in Chapter 6 involving an autonomous spacecraft descending toward the surface of an asteroid. Uncertainty in the asteroid gravity field and surface topology provide an ideal scenario for evaluating the framework of the discrete SR-MPC algorithm Additionally, realistic effects such as control-input uncertainty, sensor noise, and unknown disturbances, which are not included in the theoretical formulation of SR-MPC, are incorporated into the simulations to evaluate the performance of the discrete algorithm in a less-academic example.

A concluding chapter summarizes the algorithms developed in this thesis. Additionally, limitations and potential extensions are discussed for each of the algorithms.

Chapter 2

Model Predictive Control

In traditional receding- or compressing-horizon implementations of MPC (Model Predictive Control), control inputs are computed online by solving an FHC (Finite-Horizon optimal Control problem) over a finite time horizon, subject to state and control constraints, and with the current state of the system as the initial state. The control is then applied to the system in a feedforward (i.e., open-loop) manner over a specified time interval, followed by an update to the current state and a re-solve (re-computation) of the FHC over a compressed or receded time horizon. The re-solve provides an updated feedforward input, which is then applied to the system and the cycle repeats.

The intent of this chapter is to define the baseline MPC method that is used for comparison with the contributions in this thesis. The MPC formulation presented herein is based upon those by Chen and Allgöwer [13], Mayne *et al.* [31], Jadbabaie [18] and Primbs [37] where a CLF (Control Lyapunov Function [46, 47, 16]) is imposed on the terminal state as part of the cost function. The method additionally uses a terminal state constraint, as well as trajectory state and control constraints. Refer back to Section 1.1 of Chapter 1 for a review of the other relevant literature and contributions to MPC. Proofs for the lemmas and theorem in this chapter are provided in Appendix A in lieu of external references because future chapters build upon and extend them.

2.1 System Description and Control Objective

Consider the following nonlinear system as the *nominal* system for application of MPC:

$$\dot{z} = F(z, u_o, t), \tag{2.1}$$

with nominal state $z \in \mathbb{R}^n$, control input $u_o \in \mathbb{R}^m$, and $F(\cdot) \in \mathbb{R}^n$ is perfectly known (i.e., there are no parametric uncertainties or unknown disturbances). Without a loss of generality, the origin is considered an equilibrium point for system (2.1), $F(0,0,t) = 0$; any non-zero equilibrium point $F(\bar{z}, \bar{u}_o, t) = 0$ can be shifted to the origin.

The control objective is to obtain control input u_o that, when applied to nominal system (2.1), renders the origin ($z = 0$) asymptotically stable, with a region of attraction $\mathcal{R}_n \subseteq \mathbf{Z}_n$, such that

$$z(t) \in \mathbf{Z}_n \text{ and } u_o(t) \in \mathbf{U}_o, \ \forall\, t \geq t_0, \tag{2.2}$$

when $z(t_0) \in \mathcal{R}_n$. Sets $\mathbf{Z}_n \subseteq \mathbb{R}^n$ and $\mathbf{U}_o \subseteq \mathbb{R}^m$ define nominal state and control constraints, respectively: $\mathbf{Z}_n$ is connected and contains the origin in its interior; $\mathbf{U}_o$ is compact and contains the origin in its interior. The set $\mathcal{R}_n$ will be defined based on the architecture of the MPC algorithm.

2.2 Architecture of MPC Algorithm

The following FHC is typical of the type of constrained optimization solved in an MPC framework. The FHC finds a control input u_o that minimizes an objective function over a finite time horizon ($T \geq 0$), subject to the dynamics of nominal system (2.1), the imposed state and control constraints, and a terminal state constraint.

FHC (for MPC)

Find $J^* = \min_{u_o} J(z, u_o;\ t_i, T, z(t_i))$ where

$$J(z, u_o;\ t_i, T, z(t_i)) = \int_{t_i}^{t_i+T} h(z(\tau), u_o(\tau))d\tau + V(z(t_i + T))$$

subject to

$$\left.\begin{aligned} &\dot{z} = F(z, u_o, t), \\ &z(t) \in \mathbf{Z}_n, \\ &u_o(t) \in \mathbf{U}_o, \end{aligned}\right\} \forall\, t \in [t_i,\, t_i + T]$$

$$z(t_i + T) \in \mathbf{\Omega}_o,$$

where $z(t_i)$ is the nominal system (2.1) state at initial time t_i.

The set $\mathcal{R}_n$ will be defined based on feasibility of the FHC:

$$\mathcal{R}_n = \{\xi \in \mathbf{Z}_n : \text{ FHC is feasible with } z(t_i) = \xi\}. \tag{2.3}$$

The following conditions on the FHC and the nominal system (2.1) are instrumental for proving asymptotic stability (e.g., [13, 18]) of the MPC algorithm:

Condition 2.1. Function $h(\cdot)$ is positive definite [21], satisfying

$$h(z, u_o) \geq a||z||^p + b||u_o||^r, \quad \forall z, u_o, \tag{2.4}$$

with $p \geq 1$, $r \geq 0$, a and b both positive constants, and $h(0,0) = 0$. ◇

Condition 2.2. Function $V(z)$ is positive definite ($V(z) > 0, \forall z \neq 0, \text{with } V(0) = 0$), and there exists a control law $u_o = \mathcal{L}(z)$ such that V defines a Control Lyapunov Function for (2.1) satisfying

$$\nabla V(z) F(z, \mathcal{L}(z), t) + h(z, \mathcal{L}(z)) \leq 0, \quad \forall z \in \mathbf{\Omega}_o, \tag{2.5}$$

where $\mathbf{\Omega}_o \subset \mathbf{Z}_n$ is compact, convex, contains the origin in its interior, and is invariant for dynamics (2.1) under application of control policy $\mathcal{L}(z)$. Additionally, $\mathcal{L}(z) \in \mathbf{U}_o, \ \forall z \in \mathbf{\Omega}_o$. ◇

Condition 2.3. There exists closed ball[†] $\mathbf{B}_R$ centered at the origin such that set $\mathbf{\Omega}_o$ satisfies

$$\mathbf{B}_R \subseteq \mathbf{\Omega}_o. \tag{2.6}$$

◇

Note that invariance of $\mathbf{\Omega}_o$ under application of control law $\mathcal{L}(z)$ implies that if $z(t_0) \in \mathbf{\Omega}_o$ for some t_0, then $z(t) \in \mathbf{\Omega}_o, \ \forall t \geq t_0$ [21]. Additionally, Conditions 2.1 and 2.2 imply that cost function J in the FHC is also positive definitive ($J(z, u_o) > 0, \forall z, \forall u_o, \text{with } J(0,0) = 0$).

The following algorithm describes the MPC approach. Note, the superscript k on $u_o^k(t)$ and $z^k(t)$ in the below MPC algorithm denotes the feedforward input and the resulting nominal trajectory, respectively, associated with a re-solve at time t_k.

[†] $\mathbf{B}_\rho \triangleq \{z : \|z\| \leq \rho, \ \rho > 0\}$.

MPC Algorithm

Begin at $k = 0$ with $z(t_0) \in \mathcal{R}_n$ and iterate the following steps over re-solve times t_k for $k \in \mathbb{Z}^+$:

1. Measure state $z(t_k)$ of nominal system (2.1) and solve the FHC at time $t_i = t_k$ with $z(t_i) = z(t_k)$ and $T = T_k$ to obtain $u_o^k(t)$ on $t \in [t_k, t_k + T_k]$.
2. Monitor $z(t)$ while applying $u_o(t) = u_o^k(t)$ to nominal system (2.1) on $t \in [t_k, t_{k+1}]$, with $z(t) = z^k(t)$.
3. Check the following over $t \in [t_k, t_{k+1}]$:
   if $z(\tilde{t}) \in \mathbf{\Omega}_o$ for some $\tilde{t} \geq t_0$, then set $u_o(t) = \mathcal{L}(z), \forall t \geq \tilde{t}$ and stop iteration.

Lemma 2.1 (Re-solvability of the FHC)**.** *Suppose that the FHC is feasible at t_0 with horizon T_0, and let t_k for $k \in \mathbb{Z}^+$ be the times that a solution of the FHC is re-solved. Then, the feasibility of the FHC is guaranteed at t_k with $T_k \geq T_{k-1} - \delta_k$, $\forall k \in \mathbb{Z}^+$, $\delta_k = t_k - t_{k-1}$, $0 \leq \delta_k < T_{k-1}$ provided Condition 2.2 holds.* ◇

Proof. See Appendix A.1 for a proof of Lemma 2.1. □

For proving stability of the MPC algorithm, a sequence of monotonically increasing re-solve times is needed:

Definition 2.1 (Re-Solve Times)**.** Let t_k ($k \in \mathbb{Z}^+$) be *re-solve times* for the FHC satisfying $\inf_k \delta_k \geq \epsilon$ for some $\epsilon > 0$ where $\delta_k = t_k - t_{k-1}$. ◇

Lemma 2.2 (Shrinking Optimal Cost with Compressing or Receding Horizon)**.** *Suppose the FHC is feasible at some re-solve time t_{k-1} and T_{k-1} with optimal cost J_{k-1}^*, and Conditions 2.1, 2.2, and 2.3 hold. Then, the FHC is feasible at re-solve time t_k with $T_k \in [T_{k-1} - \delta_k, T_{k-1}]$ (in fact any $T_k \geq t_{k-1} - \delta_k$), and if $z^{k-1}(t_{k-1}) \notin \mathbf{\Omega}_o$ and $z^{k-1}(t_k) \notin \mathbf{\Omega}_o$, then the optimal cost satisfies*

$$J_k^* - J_{k-1}^* \leq -\beta, \quad \textit{for some} \quad \beta > 0. \tag{2.7}$$

◇

Proof. See Appendix A.2 for a proof of Lemma 2.2. □

While each solution of the FHC provides a feedforward input u_o to drive the nominal system (2.1) toward the origin, the ability to re-solve the FHC and thus update the feedforward input u_o based on the current state provides closed-loop feedback. The following theorem establishes closed-loop asymptotic stability and finite-time convergence of the MPC algorithm:

Theorem 2.1 (Closed-Loop Asymptotic Stability of MPC)**.** *Consider system (2.1) for z and control input u_o described by the MPC algorithm. If Conditions 2.1, 2.2, and 2.3 are satisfied, then the origin ($z = 0$) of the resulting closed-loop system is asymptotically stable with region of attraction $\mathcal{R}_n$.* ◇

Proof. See Appendix A.3 for a proof of Theorem 2.1. □

2.3 Implementation and Limitations

Practical implementation of the MPC algorithm can be difficult due to online computational capability, measurement and computation delay, parametric uncertainty, and unknown exogenous disturbances. These sources of error can lead to difficulty in maintaining feasibility, and thus re-solvability, of the FHC. Rather than providing a specific example to demonstrate the effect of uncertainty, a graphic illustration of the MPC algorithm applied to a constrained nominal system will be contrasted. A specific example demonstrating these issues will be given in the next chapter on Robust MPC, where a contrast is made between the robust method and the baseline MPC method of this chapter.

For applications of the MPC algorithm, the nominal system in (2.1) serves as a model for the actual system

$$\dot{x} = f(x, u, t), \tag{2.8}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$. The actual system contains uncertainty in either the parameters or from unknown, exogenous disturbances, and thus $F(\cdot)$ in (2.1) is a known, idealized model of $f(\cdot)$ in (2.8).

The asymptotic stability guarantees for the MPC algorithm require the nominal trajectory to remain on the computed trajectory under application of feedforward policy u_o between re-solve times. This provides an initial, feasible nominal state for $z(t)$ at subsequent re-solves, as depicted in the left-side sketch in Figure 2.1. However, if there is error

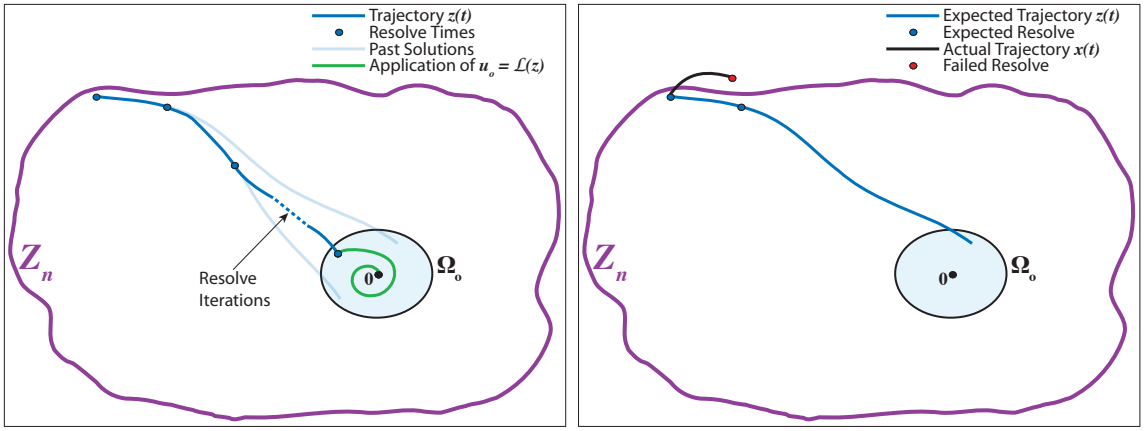


Figure 2.1: Expected MPC trajectories (left) and system uncertainty causing an infeasible state at re-solve (right).

in the nominal model, and $F(\cdot) \neq f(\cdot)$, then the actual trajectory $x(t)$ will not remain on the expected trajectory $z(t)$, thus providing no guarantee of an initial, feasible state for the FHC at the re-solve time. In fact, the actual state $x(t)$ can even violate the nominal system state constraints $\mathbf{Z}_n$, which also renders the MPC algorithm infeasible at a re-solve, as depicted in the right-side sketch of Figure 2.1.

Chapter 3

Robust Model Predictive Control with Guaranteed Re-Solvability

Computation of the feedforward MPC (Model Predictive Control) inputs relies on a nominal model of the actual system being controlled, and robustness to system uncertainties and guarantees of re-solvability (i.e., continued FHC feasibility) can be difficult to establish. The literature review in Section 1.1 of Chapter 1 highlights many significant contributions toward a robust MPC framework. This chapter presents an R-MPC (Robust and re-solvable MPC) algorithm for uncertain, nonlinear continuous-time systems and builds upon the baseline MPC method presented in Chapter 2. The formulation of the R-MPC algorithm is comprised of my joint work with Behçet Açıkmeşe that appeared in [1].

The R-MPC algorithm developed herein utilizes separate feedforward and feedback input components. The feedback is designed offline as a robust control policy based on a characterization of the uncertainty between the actual system and nominal model. The feedforward input is computed online in a compressing- or receding-horizon manner by solving an FHC-R (Finite Horizon optimal Control problem for R-MPC) that differs from the standard FHC by including a relaxation on the initial nominal state for the FHC-R re-solve. The additive feedback policy is applied with the feedforward and forms an invariant tube that is guaranteed to maintain actual trajectories within the proximity of the nominal trajectories. Note that no upper bounds are placed on the re-solve time interval with the R-MPC algorithm.

The generation of an invariant tube for robust MPC has previously been applied to linear discrete-time systems with bounded disturbances by Mayne *et al.* [29, 32], with extensions to time-varying or parameter-uncertain linear discrete-time systems by Langson *et al.* [27]

and with improvements to online computation by Raković and Mayne [40]. Extensions by Raković *et al.* [39] applied invariant tubes to robust MPC of nonlinear discrete-time systems with matched nonlinearities, including some special classes of piecewise affine systems.

The approach herein is developed for general continuous-time, uncertain nonlinear systems, with design methods for a special class of uncertain, nonlinear systems with bounded derivatives. The invariant tube is prescribed based on specified state constraints in addition to a characterization of the uncertainty/nonlinearity. The control policies are designed without any form of feedback linearization or cancellation of nonlinearities through the control policy.

3.1 Description of the Actual System and Control Objective

Consider the following uncertain, nonlinear dynamical system as the *actual* system for application of R-MPC:

$$\dot{x} = f(x, u, t), \tag{3.1}$$

with actual state $x \in \mathbb{R}^n$ and control input $u \in \mathbb{R}^m$. Let a *nominal* system model of the actual system (3.1) be given by

$$\dot{z} = F(z, u_o, t), \tag{3.2}$$

with nominal state $z \in \mathbb{R}^n$ and control input $u_o \in \mathbb{R}^m$, and where $F(\cdot)$ is a known, approximate model of $f(\cdot)$ from (3.1). Without a loss of generality, the origin $x = z = 0$ is considered an equilibrium point shared by both systems; $f(0, 0, t) = F(0, 0, t) = 0$.

The control objective is to obtain control input u that, when applied to actual system (3.1), renders the origin ($x = 0$) asymptotically stable, with a region of attraction $\mathcal{R}_a \subseteq \mathbf{X}$, such that

$$x(t) \in \mathbf{X} \text{ and } u(t) \in \mathbf{U}, \ \forall\, t \geq t_0, \tag{3.3}$$

when $x(t_0) \in \mathcal{R}_a$. Sets $\mathbf{X} \subseteq \mathbb{R}^n$ and $\mathbf{U} \subseteq \mathbb{R}^m$ define actual state and control constraints, respectively: $\mathbf{X}$ is connected and contains the origin in its interior; $\mathbf{U}$ is compact and contains the origin in its interior. The set $\mathcal{R}_a$ will be defined based on the architecture of the R-MPC algorithm.

3.2 Architecture of the R-MPC Algorithm

The control approach for R-MPC splits the control input u for actual system (3.1) into two components:

$$u(t) = u_o(t) + u_f(t), \tag{3.4}$$

where u_o is a feedforward control input and u_f is a feedback control input. The feedforward component u_o of the control input is determined through online solution of an FHC-R that utilizes the nominal system model (3.2), and the feedback component u_f is determined offline as a control policy to handle a characterization of the uncertainty in the actual system (3.1).

Sets $\mathbf{X}$ and $\mathbf{U}$ are given constraints imposed on the overall design of the control input. The following additional constraint sets (all containing the origin in their interiors[†]), are used in constructing the control approach:

$$\mathbf{U}_o + \mathbf{U}_f \subseteq \mathbf{U} \qquad \text{and} \qquad \mathbf{Z}_n + \mathbf{X}_f \subseteq \mathbf{X}. \tag{3.5}$$

Set $\mathbf{Z}_n$ is connected and contains the origin in its interior, and sets $\mathbf{X}_f$, $\mathbf{U}_o$, and $\mathbf{U}_f$ are compact and contain the origin in their interiors.

In preview, the R-MPC algorithm uses online solution of the FHC-R to design feedforward control input $u_o \in \mathbf{U}_o$ to maintain the nominal states within constraint set $\mathbf{Z}_n$. Additionally, the feedback policy $u_f \in \mathbf{U}_f$ is designed to establish invariant tube $\mathbf{X}_f$ about the nominal trajectory (utilized as a feedforward, guidance trajectory) to maintain the actual states in the proximity of the nominal states, providing robustness to dynamics uncertainty and disturbances. The set definitions in (3.5) ensure that the actual constraints in (3.3) are obeyed.

The following FHC-R is similar to the baseline FHC of Chapter 2 but with a relaxation on the initial state, which appears as a constraint on the difference between the actual and nominal state. The initial nominal state for the optimization is not required to be equal to the current actual state. This relaxation, along with feedback u_f, is useful for establishing robust re-solvability of the FHC-R, which is needed to ensure asymptotic stability of the R-MPC algorithm.

[†]For sets A and B, $C = A + B$ implies the following: if $a \in A$ and $b \in B$ then $a + b \in C$.

FHC-R (for Robust and re-solvable MPC)

Find $J^* = \min_{u_o} J(z, u_o;\ t_i, T, z(t_i))$ where

$$J(z, u_o;\ t_i, T, z(t_i)) = \int_{t_i}^{t_i+T} h(z(\tau), u_o(\tau))d\tau + V(z(t_i + T))$$

subject to

$$\left.\begin{aligned} &\dot{z} = F(z, u_o, t), \\ &z(t) \in \mathbf{Z}_n, \\ &u_o(t) \in \mathbf{U}_o, \end{aligned}\right\} \forall\, t \in [t_i,\, t_i + T]$$

$$z(t_i + T) \in \mathbf{\Omega}_o,$$

$$x(t_i) - z(t_i) \in \mathbf{X}_f, \tag{3.6}$$

where $x(t_i)$ is the actual system (3.1) state at initial time t_i.

The set $\mathcal{R}_a$, which specifies the region of attraction for the control objective in (3.3), is defined based on feasibility of the FHC-R:

$$\mathcal{R}_a = \{\xi \in \mathbf{Z}_n + \mathbf{X}_f :\ \text{FHC-R is feasible with } x(t_i) = \xi\}\,. \tag{3.7}$$

The baseline MPC algorithm conditions 2.1, 2.2, and 2.3 are also conditions for proving asymptotic stability of the R-MPC algorithm. The following additional conditions are useful in proving stability for the actual system with control determined with the R-MPC algorithm.

Condition 3.1. There exists a feedback control policy $u_f = \mathcal{K}_f(x, z) \in \mathbf{U}_f$ in (3.4) that renders set $\mathbf{X}_f$ invariant for $\eta(t) \triangleq x(t) - z(t) \in \mathbf{X}_f$ and for all $u_o(t)$, $\forall t \geq t_0$, with dynamics (3.1) for x and (3.2) for z. ◇

Note, Condition 3.1 defines set $\mathbf{X}_f$ as an invariant tube about the nominal states z: if $\eta(t_0) \in \mathbf{X}_f$ for some $t_0 \geq 0$, then $\eta(t) \in \mathbf{X}_f, u_f(t) \in \mathbf{U}_f, \forall t \geq t_0$.

Condition 3.2. There exists function V and control policy $\mathcal{L}(\cdot)$, defined as in Condition 2.2, such that control policy $u(t) = \mathcal{L}(x)$ also ensures V is a Control Lyapunov Function

for (3.1), satisfying

$$\nabla V(x) f(x, \mathcal{L}(x), t) + h(x, \mathcal{L}(x)) \leq 0, \quad \forall\, x \in \mathbf{\Omega}_o, \tag{3.8}$$

where $\mathbf{\Omega}_o \subset \mathbf{Z}_n$ is invariant for dynamics (3.1) under application of control policy $\mathcal{L}(x)$. Additionally, $\mathcal{L}(x) \in \mathbf{U}_o,\ \forall x \in \mathbf{\Omega}_o$. ◇

Condition 3.3. There exists closed ball $\mathbf{B}_r$ centered at the origin with radius $r < R$ such that set $\mathbf{X}_f$ satisfies

$$\mathbf{X}_f \subseteq \mathbf{B}_r \subset \mathbf{B}_R \subseteq \mathbf{\Omega}_o, \tag{3.9}$$

where $\mathbf{B}_R$ is defined in Condition 2.3. ◇

The following algorithm describes the R-MPC approach. Note, the superscript k on $u_o^k(t)$ and $z^k(t)$ in the R-MPC algorithm below denotes the feedforward input and the resulting nominal trajectory, respectively, associated with a re-solve at time t_k.

R-MPC Algorithm

Begin at $k = 0$ with $x(t_0) \in \mathcal{R}_a$ and iterate the following steps over re-solve times t_k for $k \in \mathbb{Z}^+$:

1. Measure state $x(t_k)$ of actual system (3.1) and solve the FHC-R at time $t_i = t_k$ with $x(t_i) = x(t_k)$ and $T = T_k$ to obtain $u_o^k(t)$ on $t \in [t_k, t_k + T_k]$.

2. Monitor $z(t)$ and $x(t)$ while applying $u_o(t) = u_o^k(t)$ to nominal system (3.2) and $u(t) = u_o^k(t) + u_f(t)$ to actual system (3.1) on $t \in [t_k, t_{k+1}]$, with $z(t) = z^k(t)$ and $u_f(t) = \mathcal{K}_f(x(t), z(t))$.

3. Check the following over $t \in [t_k, t_{k+1}]$:

   (a) if $z(\tilde{t}) \in \mathbf{\Omega}_o$ for some $\tilde{t} \geq t_0$, then set $u_o(t) = \mathcal{L}(z(t)), \forall t \geq \tilde{t}$ and skip step 1 in iteration.

   (b) if $x(\bar{t}) \in \mathbf{\Omega}_o$ for some $\bar{t} \geq t_0$, then set $u(t) = \mathcal{L}(x(t)), \forall t \geq \bar{t}$ and stop iteration.

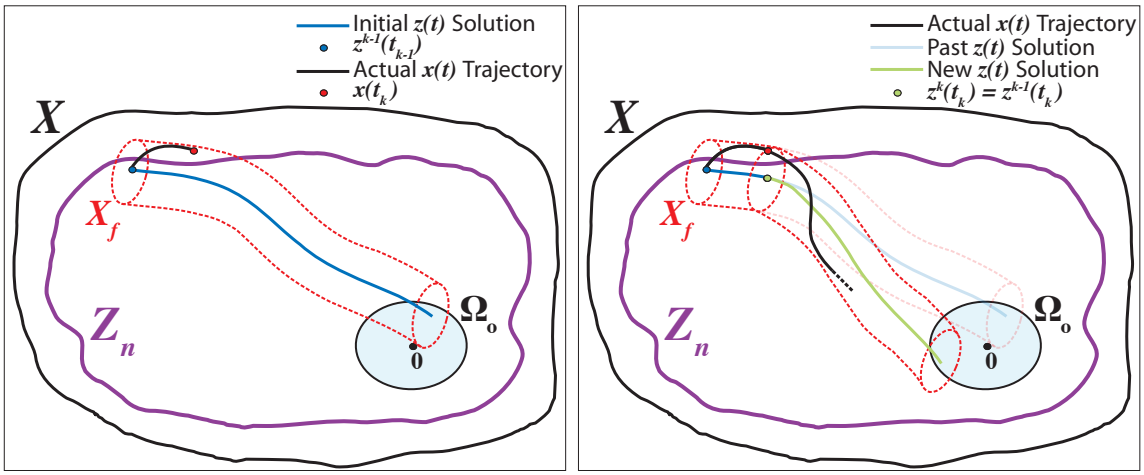
The following Lemma ensures re-solvability of the FHC-R. Note, there are no upper bounds placed on the frequency of re-solves (i.e., the time interval between re-solves) with the R-MPC algorithm.

Lemma 3.1 (Re-solvability of the FHC-R)**.** *Suppose that the FHC-R is feasible at t_0 with horizon T_0, and let t_k for $k \in \mathbb{Z}^+$ be the times that a solution of the FHC-R is re-solved. Then, the feasibility of the FHC-R is guaranteed at t_k with $T_k \geq T_{k-1} - \delta_k$, $\forall k \in \mathbb{Z}^+$, $\delta_k = t_k - t_{k-1},\ 0 \leq \delta_k < T_{k-1}$ provided Conditions 2.2 and 3.1 hold.* ◇

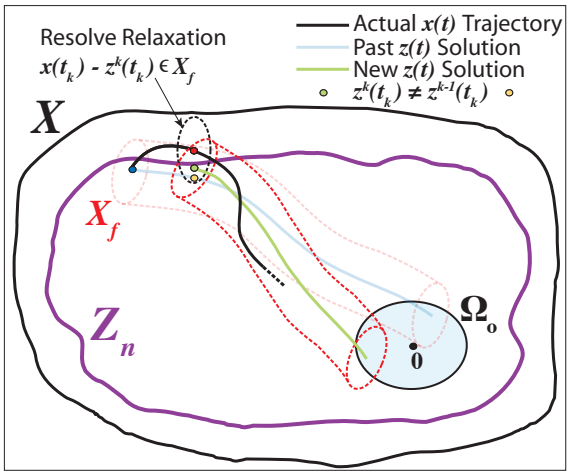
Proof. The proof follows that of Lemma 2.1 in Appendix A.1 for FHC re-solvability, with the FHC-R substituted for the FHC. Additionally, since the actual dynamics (3.1) are not known perfectly, the following addendum is needed for the initial feasible state $z^k(t_k)$ at the re-solve time t_k for the FHC-R:

Set $\mathbf{X}_f$ is invariant due to Condition 3.1 when feedback $u_f(t) = \mathcal{K}_f(x, z)$ is applied alongside $u_o^{k-1}(t)$ to actual system (3.1), which ensures $x(t_k) - z^{k-1}(t_k) \in \mathbf{X}_f$; thus, state $z^k(t_k) = z^{k-1}(t_k)$ is the initial state of a feasible trajectory. □

Remark 3.1 (Relaxation of Initial Nominal State). The nominal trajectory z from solution of the FHC-R is a guidance trajectory that the actual state x follows under application of control policy (3.4). The FHC-R constraint (3.6) provides a relaxation on the initial nominal state $z(t_i)$ for each re-solve; the relaxation comes from the invariance of $\mathbf{X}_f$ ensured by the feedback policy in Condition 3.1. The relaxation allows the FHC-R to select a $z(t_i)$ on re-solves that is either connected or disconnected with the prior nominal solution. Figure 3.1 illustrates these two scenarios. The FHC-R could require the nominal path to remain connected through an equality constraint (i.e., $z^k(t_k) = z^{k-1}(t_k)$), but the relaxation provides the FHC-R a larger solution space in which improved guidance trajectories might be found. Note, the actual path will remain connected throughout the re-solves. ◇



Initial FHC-R Feasible | Fixing $z^k(t_k) = z^{k-1}(t_k)$



Relaxing $x(t_k) - z^k(t_k) \in \mathbf{X}_f$

Figure 3.1: Given a feasible FHC-R (top left), re-solves could fix the nominal initial state (top right), but the relaxed nominal initial state (bottom) offers more flexibility.

Lemma 2.2 from the baseline MPC also applies to the R-MPC algorithm for providing a shrinking optimal cost in a compressing- or receding-horizon implementation. The proof of Lemma 2.2 applied to the R-MPC algorithm follows the baseline MPC with the FHC-R substituted for the FHC, along with use of the addendum discussed in the above proof of Lemma 3.1 that allows for the continued feasibility of the FHC-R on a re-solve.

The following theorem establishes closed-loop asymptotic stability for the R-MPC algorithm. Following the proof, Figure 3.2 presents a pictorial sketch of the R-MPC algorithm.

Theorem 3.1. *Consider system (3.1) for x with a control input u described by the R-MPC algorithm. If Conditions 2.1, 2.2, 3.1, 3.2, and 3.3 are satisfied, then the origin ($x = 0$) of the resulting closed-loop system is asymptotically stable with region of attraction $\mathcal{R}_a$.* $\diamond$

Proof. Given the R-MPC algorithm and $x(t_0) \in \mathcal{R}_a$ such that the FHC-R is feasible with some T_0, suppose there exists $k \in \mathbb{Z}^+$ such that $z^{k-1}(t_{k-1}) \notin \mathbf{\Omega}_o$ and $z^{k-1}(t_k) \notin \mathbf{\Omega}_o$. This implies $z^{k-1}(t) \notin \mathbf{\Omega}_o$ for $t \in [t_{k-1}, t_k]$, and thus (2.7) holds: $J_k^* - J_{k-1}^* \leq -\beta$, for some $\beta > 0$ (Note, β is independent of k, as established in the proof of Lemma 2.2). Consequently, if the nominal trajectory z does not enter $\mathbf{\Omega}_o$ in finite time, then there exists $k \in \mathbb{Z}^+$ such that $J_k^* < 0$, which is a contradiction. This together with Condition 2.2 imply the existence of finite time $\tilde{t} \geq t_0$ such that $z(t) \in \mathbf{\Omega}_o, \forall t \geq \tilde{t}$.

Application of Step 3a in the R-MPC algorithm, $u_o = \mathcal{L}(z)$ for $t \geq \tilde{t}$, and use of Condition 2.2 imply the closed-loop nominal system converges asymptotically to the origin when $x(t_0) \in \mathcal{R}_a$. Note, V is a Control Lyapunov Function for nominal system (3.2) with $\dot{V}(z) < 0, \forall z \in \mathbf{\Omega_o}$, except $\dot{V}(0) = 0$.

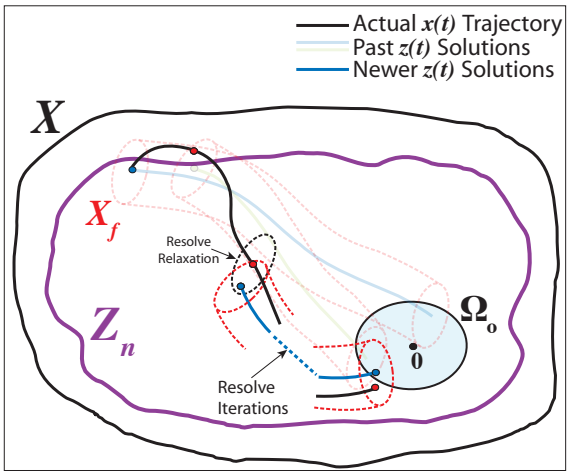
Since, the closed-loop nominal system converges asymptotically to the origin, there exists $\bar{t} \geq \tilde{t} \geq 0$ such that $\|z(t)\| \leq R - r$ for $t \geq \bar{t}$, where $R > r > 0$ are as defined in Condition 3.3. This leads to

$$\|x(t)\| \leq \|x(t) - z(t)\| + \|z(t)\| \leq r + (R - r) = R, \quad \forall t \geq \bar{t},$$

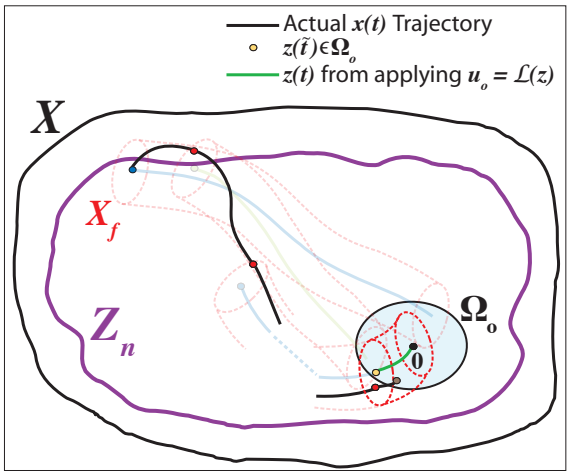
which implies that $x(t) \in \mathbf{\Omega}_o, \forall t \geq \bar{t}$. Application of Step 3b in the R-MPC algorithm, $u = \mathcal{L}(x)$ for $t \geq \bar{t}$, and use of Condition 3.2 implies

$$\lim_{t \to \infty} \|x(t)\| = 0$$

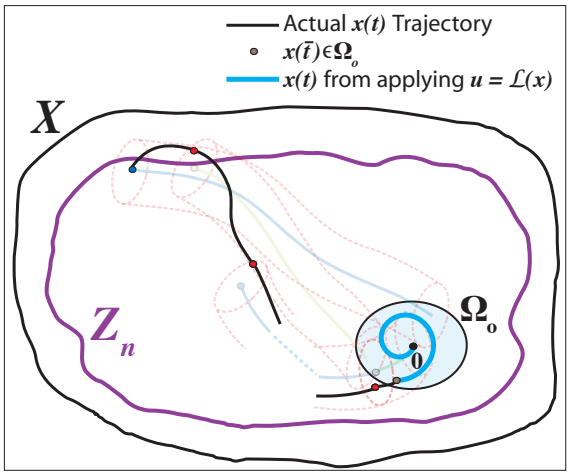
since V is also a Control Lyapunov Function for actual system (3.1) with $\dot{V}(x) < 0, \forall x \in \mathbf{\Omega_o}$, except $\dot{V}(0) = 0$. Therefore, the closed-loop actual system (3.1) converges asymptotically to the origin, $\forall x(t_0) \in \mathcal{R}_a$, with control input u given by the R-MPC algorithm. $\square$



Iterative R-MPC re-solves



Apply $u_o = \mathcal{L}(z)$ when $z \in \mathbf{\Omega}_o$



Apply $u = \mathcal{L}(x)$ when $x \in \mathbf{\Omega}_o$

Figure 3.2: Sketch of the R-MPC algorithm, including re-solves (top left), nominal-system asymptotic convergence (top right), and actual-system asymptotic convergence (bottom).

Remark 3.2 (Asymptotic Stability of the Actual System). If Conditions 3.2 and 3.3 are not specified, then asymptotic converge of the actual system to the origin is not guaranteed in the proof of Theorem 3.1. In this case, there is no local control policy $u = \mathcal{L}(x)$ to drive the actual system to the origin. In addition, there is no guarantee that $\mathbf{X}_f \subset \mathbf{\Omega}_o$, so x might not enter $\mathbf{\Omega}_o$. ◇

Remark 3.3 (Convergence of Actual System into a Terminal Set $\mathbf{X}_f$). Conditions 2.2 and 2.3 on the nominal system establish closed-loop asymptotic convergence to the origin for the nominal system with the R-MPC algorithm. When $z = 0$, the feedback policy $u_f = \mathcal{K}_f(x, 0)$ from Condition 3.1 renders $\mathbf{X}_f$ invariant for $\eta = x$; the feedforward would remain

$u_o = \mathcal{L}(z) = \mathcal{L}(0)$ in control input u from (3.4). Thus, invariant tube $\mathbf{X}_f$ also generates an invariant terminal set for the actual system. This terminal convergence would also result from skipping Step 3b in the R-MPC algorithm, and Conditions 3.2 and 3.3 are unnecessary.

◇

3.3 Specialization to a Class of Systems with Derivatives Contained in Convex Sets

In this section, the R-MPC algorithm is specialized to an important class of systems for which explicit design methods can be developed that satisfy Conditions 2.1, 2.2, 3.1, 3.2, and 3.3. The following characterizes the actual system for this special class of systems:

$$\begin{aligned} \dot{x} &= Ax + Bu + E\phi(t, q), \\ q &= C_q x + D_q u, \end{aligned} \tag{3.10}$$

where $\phi : \mathbb{R} \times \mathbb{R}^{n_q} \to \mathbb{R}^{n_p}$ with $\phi(t, 0) = 0, \forall t$, is a continuously differentiable function representing the uncertain nonlinear part of the dynamics. This form implies $f(x, u, t) = Ax + Bu + E\phi(t, q)$ in (3.1) with $q \in \mathbb{R}^{n_q}$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $E \in \mathbb{R}^{n \times n_p}$, $C_q \in \mathbb{R}^{n_q \times n}$, and $D_q \in \mathbb{R}^{n_q \times m}$.

The nominal model dynamics is assumed to have the following form:

$$\begin{aligned} \dot{z} &= Az + Bu_o + E\psi(t, q_o), \\ q_o &= C_q z + D_q u_o, \end{aligned} \tag{3.11}$$

where $\psi : \mathbb{R} \times \mathbb{R}^{n_q} \to \mathbb{R}^{n_p}$ with $\psi(t, 0) = 0, \forall t$, is an approximation for $\phi(\cdot)$ in actual system (3.10). Thus, $F(z, u_o, t) = Az + Bu_o + \psi(t, q_o)$ in (3.2), and $q_o \in \mathbb{R}^{n_q}$.

Nonlinear functions $\phi(\cdot)$ and $\psi(\cdot)$ in (3.10) and (3.11), respectively, are assumed to have Jacobians in convex sets, along with a bounded mismatch:

Condition 3.4 (Jacobian in Convex Set)**.** Functions ϕ and ψ are continuously differentiable and there exists a closed and convex set of matrices $\Theta \subseteq \mathbb{R}^{n_p \times n_q}$ such that

$$\frac{\partial \phi}{\partial q}(t, q) \in \Theta \quad \text{and} \quad \frac{\partial \psi}{\partial q}(t, q) \in \Theta, \quad \forall q, \forall t. \tag{3.12}$$

◇

Condition 3.5 (Bounded Mismatch)**.** There exists a scalar $\gamma > 0$ such that

$$\|w(t, z, u_o)\| \leq \gamma, \quad \forall t, \forall z \in \mathbf{Z}_n, \forall u_o \in \mathbf{U}_o, \tag{3.13}$$

where $w(t, z, u_o) = \phi(t, q_o) - \psi(t, q_o)$ with $q_o = C_q z + D_q u_o$ from (3.11). ◇

Remark 3.4. Condition 3.5 is satisfied when $\mathbf{Z}_n$ and $\mathbf{U}_o$ are compact sets, and ϕ and ψ are continuous in their arguments and bounded in t. For example, suppose there exists a norm bound on the uncertainty,

$$\|\phi(t, q_o) - \psi(t, q_o)\| \leq \rho \|q_o\|, \quad \forall t, \forall q_o,$$

where $\rho > 0$. Furthermore, suppose there also exist positive scalars ρ_1 and ρ_2 such that

$$\|z\| \leq \rho_1, \ \forall z \in \mathbf{Z}_n, \quad \text{and} \quad \|u_o\| \leq \rho_2, \ \forall u_o \in \mathbf{U}_o.$$

Then, Condition 3.5 is satisfied with $\gamma = \rho(\rho_1 \|C_q\| + \rho_2 \|D_q\|)$. ◇

Remark 3.5. When the nominal system (3.11) is LTI (Linear Time Invariant), which occurs when nonlinearity $\psi(t, z) = \psi(t)$ is an exogenous input, the nominal dynamics provide linear equality constraints in the FHC-R. Further, when all other state and control constraints ($\mathbf{Z}_n$, $\mathbf{X}_f$, and $\mathbf{U}_o$) define a convex feasible domain, then the FHC-R becomes a convex optimization problem[‡], which can be solved in a computationally efficient and reliable manner by using interior-point methods [36]. Analysis of systems of the form (3.10) that have LTI nominal models for (3.11) is useful for real-time autonomous control. ◇

The *error* dynamics between the actual and nominal states in (3.10) and (3.11), respectively, are

$$\begin{aligned} \dot{\eta} &= A\eta + Bu_f + E\left[\phi(t, q) - \psi(t, q_o)\right], && (3.14) \\ &= A\eta + Bu_f + E\left[\phi(t, q) - \phi(t, q_o)\right] + E\left[\phi(t, q_o) - \psi(t, q_o)\right], \\ &= A\eta + Bu_f + E\pi(t, \eta, u_f) + Ew(t, z, u_o), && (3.15) \end{aligned}$$

where $\eta \triangleq x - z$ is the error state, $u_f \triangleq u - u_o$ defines the feedback input, $w(t, z, u_o) = \phi(t, q_o) - \psi(t, q_o)$ is from Condition 3.5, and $\pi(t, \eta, u_f) = \phi(t, C_q x + D_q u) - \phi(t, C_q z + D_q u_o)$.

[‡]All equality constraints in a convex optimization problem must be linear equalities.

The following Lemma is a generalization of the Mean Value Theorem [5, 8] and is used to obtain an LDI (Linear Differential Inclusion) for the error dynamics in (3.15).

Lemma 3.2. *Consider a continuously differentiable function $g : \mathbb{R}^{n_1} \mapsto \mathbb{R}^{n_2}$ with its Jacobian given by $\frac{\partial g}{\partial y}(y)$. Suppose there exists a closed convex set $\Lambda \in \mathbb{R}^{n_2 \times n_1}$ such that*

$$\frac{\partial g}{\partial y}(y) \in \Lambda, \quad \forall y \in \mathbb{R}^{n_1}.$$

Then, for every $y_1, y_2 \in \mathbb{R}^{n_1}$ there exists $\Delta \in \Lambda$ such that

$$g(y_2) - g(y_1) = \Delta(y_2 - y_1).$$

$\diamond$

Applying Lemma 3.2 with Condition 3.4 for function ϕ provides the following relationship for function $\pi(t, \eta, u_f)$ in error dynamics (3.15):

$$\pi(t, \eta, u_f) = \theta(t)(C_q \eta + D_q u_f), \quad \text{where} \quad \theta(t) \in \Theta, \forall\, t. \tag{3.16}$$

This relationship aids in the generation of feedback laws that satisfy Condition 3.1 for the uncertain nonlinear systems in this section.

The following condition gives a design specification for the state and control constraints in (3.5) that will be satisfied in the forthcoming design framework. The condition provides a polytopic description of the invariant tube and terminal set and an ellipsoidal description of the control constraints. More general convex characterizations of the constraint sets are also possible and can easily be integrated into the design framework.

Condition 3.6 (State and Control Constraints)**.**

$$\begin{aligned}
&\mathbf{Z}_n \supseteq \mathcal{Z}_\Omega \triangleq \{z \in \mathbb{R}^n :\ a_i^T z \leq 1,\ i = 1, \ldots, m_o\},\\
&\mathbf{X}_f \subseteq \mathcal{X}_f \triangleq \{\eta \in \mathbb{R}^n :\ b_i^T \eta \leq 1,\ i = 1, \ldots, m_f\},\\
&\mathbf{U}_o \subseteq \{u_o \in \mathbb{R}^m :\ u_o^T \Pi_o u_o \leq 1\},\\
&\mathbf{U}_f \subseteq \{u_f \in \mathbb{R}^m :\ u_f^T \Pi_f u_f \leq 1\},
\end{aligned}$$

where Π_o, Π_f are symmetric positive-definite matrices, and the design-specification for set $\mathcal{X}_f$ is determined such that $\mathcal{X}_f + \mathbf{Z}_n \subseteq \mathbf{X}$: thus, $\mathbf{X}_f \subseteq \mathcal{X}_f$ results in additional design

conservatism to ensure the constraints in (3.5) are satisfied. ◇

Note, set $\mathbf{Z}_n$ need not be convex, but terminal set $\mathbf{\Omega}_o$ does need to be convex and will be constructed so that $\mathbf{\Omega}_o \subseteq \mathcal{Z}_\Omega$.

The following is a corollary of Theorem 3.1 that describes a design procedure for systems with norm-bounded derivatives. An extension to this corollary is provided in [1] for systems with uncertain nonlinear terms that have derivatives contained in polytopes.

Corollary 3.1.1 (Constructive R-MPC Inequalities for a Class of Systems)**.** *Consider an uncertain nonlinear system (3.10) with a nominal model given by (3.11) satisfying Conditions 3.5 and 3.6, and Condition 3.4 with*

$$\Theta = \{\theta \in \mathbb{R}^{n_p \times n_q} \,:\, \|\theta\| \leq 1\}\,. \tag{3.17}$$

Suppose there exist matrices $P = P^T > 0$, $Q = Q^T > 0$, L, Y and positive scalars λ, β, μ, c_1, and c_2 satisfying the following matrix inequalities,

$$\begin{bmatrix} PA^T + AP + BL + L^TB^T + P/\lambda + (\beta + \lambda\gamma^2)EE^T & PC_q^T + L^TD_q^T \\ C_qP + D_qL & -\beta I \end{bmatrix} \leq 0 \tag{3.18}$$

$$\begin{bmatrix} QA^T + AQ + BY + Y^TB^T + \mu EE^T & QC^T + Y^TD^T & QC_q^T + Y^TD_q^T \\ CQ + DY & -I & 0 \\ C_qQ + D_qY & 0 & -\mu I \end{bmatrix} \leq 0 \tag{3.19}$$

$$\begin{bmatrix} P & L^T \\ L & \Pi_f^{-1} \end{bmatrix} \geq 0\,, \quad \begin{bmatrix} Q & Y^T \\ Y & \Pi_o^{-1} \end{bmatrix} \geq 0, \tag{3.20}$$

$$a_i^T Q a_i \leq 1, i = \{1, \ldots, m_o\}, \tag{3.21}$$

$$b_j^T P b_j \leq 1, j = \{1, \ldots, m_f\}, \tag{3.22}$$

$$Q \geq c_1 I > c_2 I \geq P\,, \tag{3.23}$$

where C and D satisfy $C^TD = 0$. Then, ellipsoids $\mathbf{\Omega}_o \triangleq \{x : x^TQ^{-1}x \leq 1\}$ and $\mathbf{X}_f \triangleq \{\eta : \eta^TP^{-1}\eta \leq 1\}$ and the R-MPC algorithm with

$$h(z, u_o) = \|Cz\|^2 + \|Du_o\|^2 \quad \textit{and} \quad V(z) = z^TQ^{-1}z, \tag{3.24}$$

$$\mathcal{L}(z) = Kz\,, \qquad K = YQ^{-1} \tag{3.25}$$

$$\mathcal{K}_f(x,z) = K_f(x-z)\,, \qquad K_f = LP^{-1} \tag{3.26}$$

result in an asymptotically stable closed-loop system for (3.10) with region of attraction $\mathcal{R}_a$ given in (3.7) and satisfaction of the constraints in Condition 3.6. ◇

Proof. See Appendix A.4 for a proof of Corollary 3.1.1. The proof comes from establishing invariant ellipsoids [2, 8] around both the nominal trajectory z and the origin by using the matrix inequalities in the corollary and the R-MPC algorithm conditions. □

Remark 3.6. All of the matrix inequalities in Corollary 3.1.1 are an LMI (Linear Matrix Inequality), except for (3.18) which is a BMI (Bilinear Matrix Inequality). However, the BMI is an LMI for a given $\lambda > 0$. Therefore, a simple line search on λ can be applied to solve the system of matrix inequalities. ◇

3.4 An Illustrative Example to Contrast R-MPC and MPC

The R-MPC algorithm is contrasted with the baseline MPC algorithm for a simple example of a 2-D system. A comparison will also be provided to show the effect of how well the nominal model captures the nonlinearity in the actual system.

Let the actual and nominal system dynamics in (3.10) and (3.11), respectively, have the following parameters:

$$\dot{x} = Ax + Bu + E\phi(t,q), \qquad q = C_q x + D_q u, \qquad \phi(t,q) = \omega(t)\sin^2(C_q x), \tag{3.27}$$

$$\dot{z} = Az + Bu_o + E\psi(t,q_o), \qquad q_o = C_q z + D_q u_o, \qquad \psi(t,q_o) = \omega_0 \sin^2(C_q z), \tag{3.28}$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad E = \begin{bmatrix} 0 \\ -0.1 \end{bmatrix}, \tag{3.29}$$

$$C_q = \begin{bmatrix} 1 & 0 \end{bmatrix}, \qquad D_q = 0, \tag{3.30}$$

where x and z are two-dimensional vectors with position and velocity components:

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}.$$

Function ψ is the nominal model for actual system nonlinearity ϕ, and parameter $\omega(t) \in [0, 0.5]$ for nonlinearity ϕ. The value for ω_0 in nonlinearity ψ will be set to 0 and 0.2 in separate simulations to demonstrate the effect of having a good nominal model for application of baseline MPC.

The nonlinearities satisfy Condition 3.4 with Θ as in (3.17): $\|\frac{\partial \phi}{\partial q}\| \leq 1$ and $\|\frac{\partial \psi}{\partial q}\| \leq 1$. Further, Condition 3.5 is satisfied with $\gamma = 0.5$ when $\omega_0 = 0$ and $\gamma = 0.3$ when $\omega_0 = 0.2$. The FHC and FHC-R cost function is $h(z, u_o)$ from Corollary 3.1.1 with matrices

$$C = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \\ 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{3.31}$$

the time horizon is fixed at $T = 30$ seconds, and the initial condition is

$$x(t_0) = \begin{pmatrix} 4 \\ 0.4 \end{pmatrix},$$

with $t_0 = 0$.

The actual state and control constraints for the example are

$$\begin{aligned} \mathbf{X} &\triangleq \{x : x_1 \in [-0.35, 5], x_2 \in [-1, 1]\}, \\ \mathbf{U} &\triangleq \{u : \|u\| \leq 1.4\}. \end{aligned}$$

These constraints are partitioned into the design specifications from Condition 3.6 as follows:

$$\begin{aligned} \mathcal{Z}_\Omega : &\quad a_i = \left\{ \begin{pmatrix} \frac{1}{4.95} \\ 0 \end{pmatrix}, \begin{pmatrix} -\frac{1}{0.3} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \frac{1}{0.9} \end{pmatrix}, \begin{pmatrix} 0 \\ -\frac{1}{0.9} \end{pmatrix} \right\}, \\ \mathcal{X}_f : &\quad b_i = \left\{ \begin{pmatrix} \frac{1}{0.05} \\ 0 \end{pmatrix}, \begin{pmatrix} -\frac{1}{0.05} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \frac{1}{0.1} \end{pmatrix}, \begin{pmatrix} 0 \\ -\frac{1}{0.1} \end{pmatrix} \right\}, \\ \mathbf{U}_o : &\quad \Pi_o = \tfrac{1}{1.2^2}\,, \\ \mathbf{U}_f : &\quad \Pi_f = \tfrac{1}{0.2^2}\,, \end{aligned}$$

where the a_i values define the full $\mathbf{Z}_n$ as well as $\mathcal{Z}_\Omega$ for the examples, and $i = \{1, 2, 3, 4\}$. Parameters Π_o and Π_f bound components $\|u_o\| \leq 1.2$ and $\|u_f\| \leq 0.2$, respectively, in control u from (3.4). Note, the above sets provide design specifications such that $\mathbf{Z}_n + \mathcal{X}_f \subseteq$

$\mathbf{X}$ and $\mathbf{U}_o + \mathbf{U}_f \subseteq \mathbf{U}$, and control design with Corollary 3.1.1 will ensure that the required R-MPC control and state constraints are met and satisfy (3.5). For application of the baseline MPC, the state constraints will be set to $\mathbf{Z}_n$ and the control constraints to the full set $\mathbf{U}$ since there is no separate feedback input. Utilizing the more-conservative state constraints $\mathbf{Z}_n$ builds some conservatism in the MPC design so that infeasible solutions might recover prior to violation of the full state-constraint set $\mathbf{X}$. This could be important, for example, if the system modeled a mechanical vehicle with the constraint set $\mathbf{X}$ being a physical barrier.

The LMIs in Corollary 3.1.1 are solved by using SDPT3 [50]. The solution provides the controller gains that define the feedback controller (K_f) for invariant tube $\mathbf{X}_f$ and the local controller (K) for terminal set $\mathbf{\Omega}_o$. The matrices P and Q that define tube $\mathbf{X}_f$ and terminal set $\mathbf{\Omega}_o$, respectively, are also part of the solution:

$$K_f = \begin{bmatrix} -3.6242 & -3.0546 \end{bmatrix}, \qquad P = \begin{bmatrix} 0.0012 & -0.0011 \\ -0.0011 & 0.0044 \end{bmatrix},$$

$$K = \begin{bmatrix} -2.8907 & -2.0622 \end{bmatrix}, \qquad Q = \begin{bmatrix} 0.0755 & -0.0498 \\ -0.0498 & 0.2227 \end{bmatrix}.$$

A contrast of MPC and R-MPC for a simulation with $\omega_0 = 0$ is shown in Figure 3.3; in these simulations, parameter $\omega(t)$ is fixed at 0.5 to provide a worst-case disturbance to the actual system. Re-solves are performed on a fixed, 2-second interval in both simulations. The state constraints and markers for each re-solve are included in each plot as well. The smaller plot with axes η_1 and η_2, to the lower right of the bottom plot for R-MPC in Figure 3.3, is the error state between the actual and nominal systems.

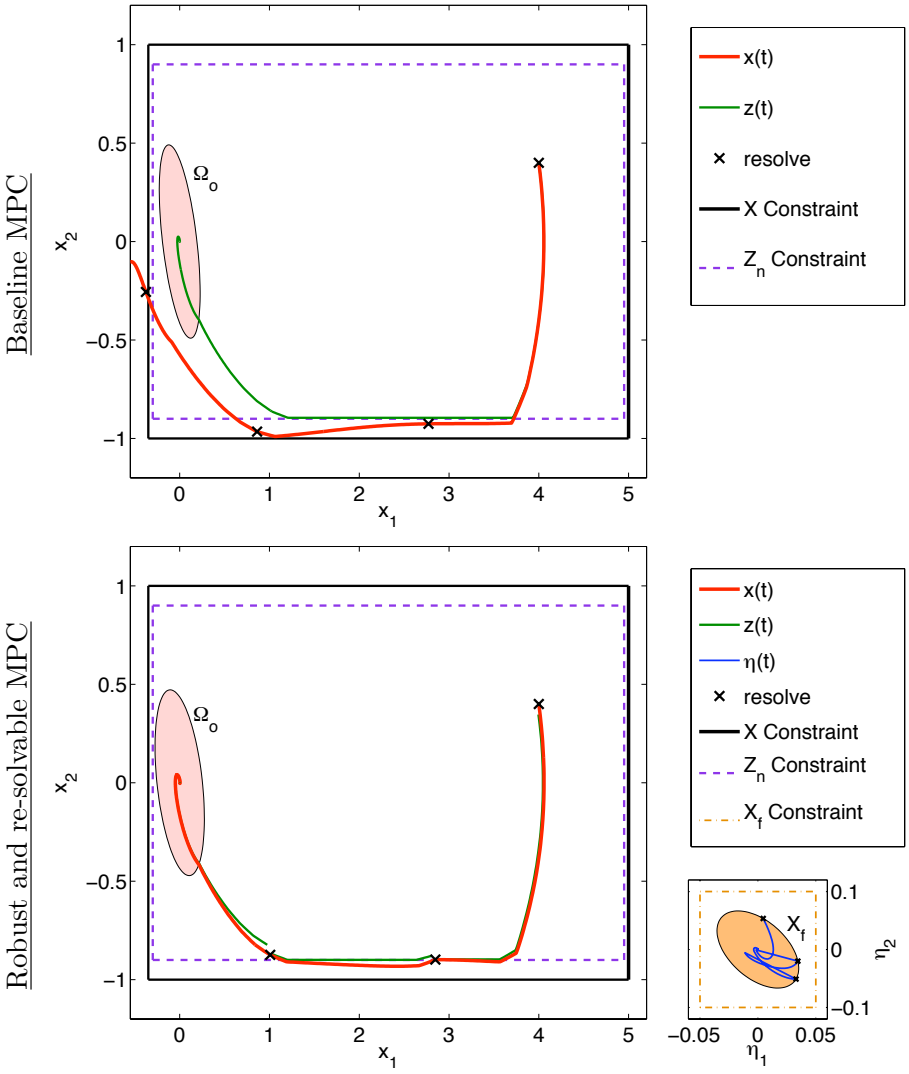


Figure 3.3: The baseline MPC algorithm remains infeasible (top), while the R-MPC algorithm converges (bottom) with $\omega_0 = 0$ in the nominal model and re-solves every 2 seconds.

In the MPC simulation (top plot in Figure 3.3), the 2-second re-solves occur outside of feasible nominal state constraints $\mathbf{Z}_n$, so no solutions are obtained to update the MPC control input. One practical approach to the lack of re-solve solutions is to continue application of the last feasible input until another feasible solution is obtained. This approach assumes that the actual trajectory will eventually reenter the feasible constraints $\mathbf{Z}_n$. As seen in the MPC plot, the feasible region is re-entered, but the fixed 2-second re-solve rate does not occur until the actual trajectory is outside of feasibility again. In fact, the full state constraint $\mathbf{X}$ is violated prior to the third re-solve, and application of the original MPC control input never drives the actual system into the terminal set. Another practical

approach is to re-solve the MPC algorithm more frequently in an attempt to maintain the actual trajectory within the feasible nominal state constraints. However, with more frequent re-solves, there is still no guarantee of continued feasibility, nor is there a general, explicit measure for how frequently re-solves should occur. In fact, the needed re-solve rate for some systems could exceed practical computation capabilities.

In contrast, the R-MPC simulation (bottom plot in Figure 3.3) provides asymptotic stability to the actual system. The separate feedback controller, designed by satisfying the matrix inequalities in Corollary 3.1.1, maintains the actual trajectory within the tube $\mathbf{X}_f$ about the nominal trajectory, enabling feasible re-solves even when the actual trajectory leaves the nominal state constraints $\mathbf{Z}_n$. Further, the tube provides robust re-solvability regardless of the rate of the re-solves. The tube cross section and the error state η is shown to the lower right of the R-MPC simulation. The effect of relaxation constraint (3.6) in the FHC-R is also noticeable in the tube plot; on re-solves, the optimization chooses to place the initial nominal state on the boundary of the tube. Once the actual system enters terminal set $\mathbf{\Omega}_o$, the error state also goes asymptotically to 0.

Both the MPC and R-MPC simulations enforce the prescribed control constraints, shown in Figure 3.4. The top plots are for the baseline MPC control and the bottom plots are for the R-MPC control. Since baseline MPC only utilizes the feedforward u_o, the control to the actual system is identical to the nominal system (i.e., $u = u_o$ and feedback $u_f = 0$ in the top plots). The small jump in the MPC control input at approximately 5 seconds corresponds to when the nominal system enters terminal set $\mathbf{\Omega}_o$ and the controller is switched to the local controller $\mathcal{L}(\cdot)$. As shown prior in Figure 3.3, the baseline MPC control input does not drive the actual system to the origin, and since no feasible solutions are found during the three subsequent re-solves, the original nominal input is utilized for the entire simulation. In comparison, the bottom plots depict the R-MPC control input u, which does drive the actual system (3.10) to the origin. The R-MPC control input is made up of the feedforward u_o and feedback u_f components. As seen, the components and their sum also satisfy the prescribed R-MPC control constraints. The bump in the R-MPC control input also near 5 seconds corresponds to the nominal system, followed very closely by the actual system, entering the terminal set $\mathbf{\Omega}_o$ and the controllers being switched to the local controller $\mathcal{L}(\cdot)$. Note, once the actual system is switched to the local controller, no separate feedback is utilized and $u_f = 0$ as seen in the figure.

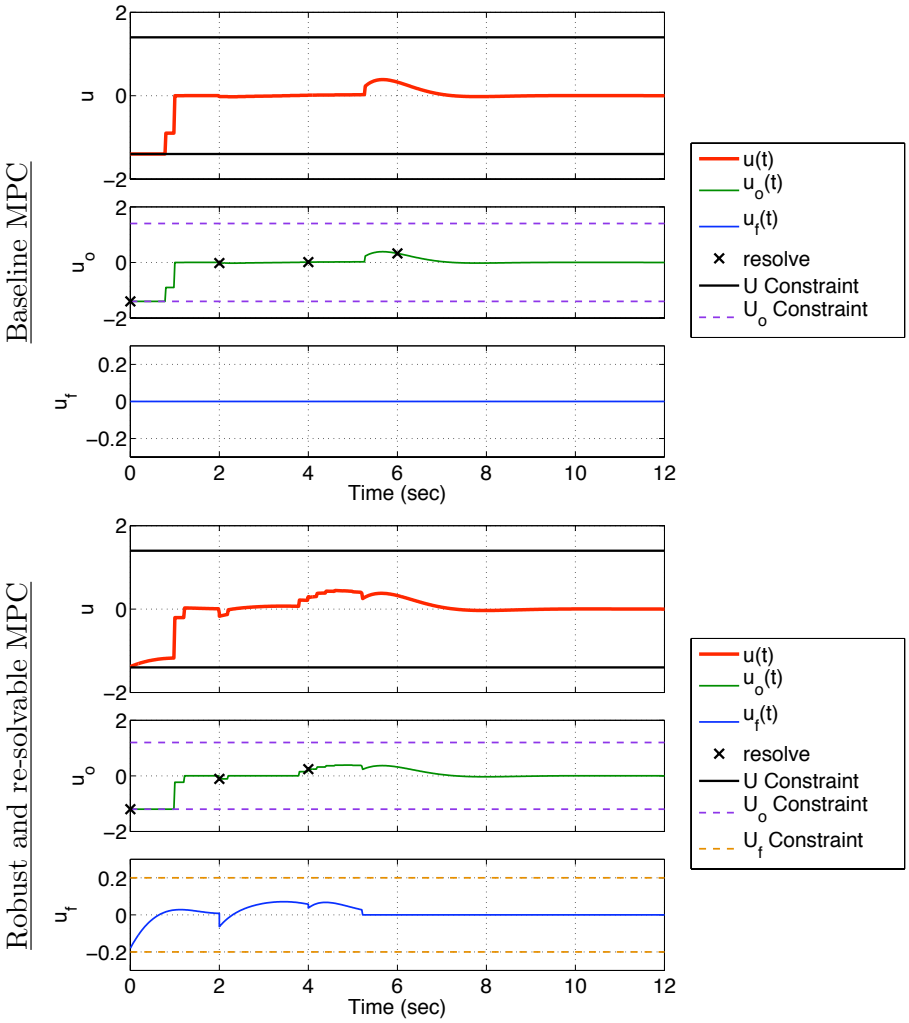


Figure 3.4: Control inputs from both MPC and R-MPC obey prescribed constraints.

The asymptotic convergence of the actual system to the origin is enabled in part by the satisfaction of Condition 3.3. Figure 3.5 shows that Corollary 3.1.1 establishes $\mathbf{X}_f \subset \mathbf{\Omega}_o$.

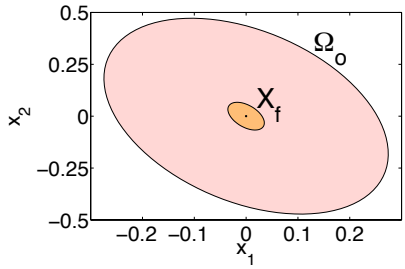


Figure 3.5: Feedback state constraints $\mathbf{X}_f \subset \mathbf{\Omega}_o$ satisfy Condition 3.3.

The asymptotic convergence problem for the MPC algorithm in Figure 3.3 could be remedied by decreasing the re-solve rate (e.g., re-solve every 1 second) so that a re-solve

occurs when the actual trajectory reenters the feasible state constraints $\mathbf{Z}_n$. A benefit of the R-MPC algorithm is that invariant tube $\mathbf{X}_f$ maintains re-solve feasibility regardless of the re-solve rate, so no guessing is required on the frequency of re-solves. The quality of the nominal model also has an effect on the baseline MPC. When $\omega_0 = 0.2$, the nominal uncertainty model better captures the actual nonlinearity. Figure 3.6 contrasts the MPC (top) and R-MPC (bottom) algorithms with this different model; re-solves are still performed every 2 seconds. Several infeasible MPC re-solves still occur, but the improved model keeps the nominal and actual trajectory close enough that one of the 2-second re-solves occurs within feasible nominal state constraints and avoids violation of the actual state constraints.

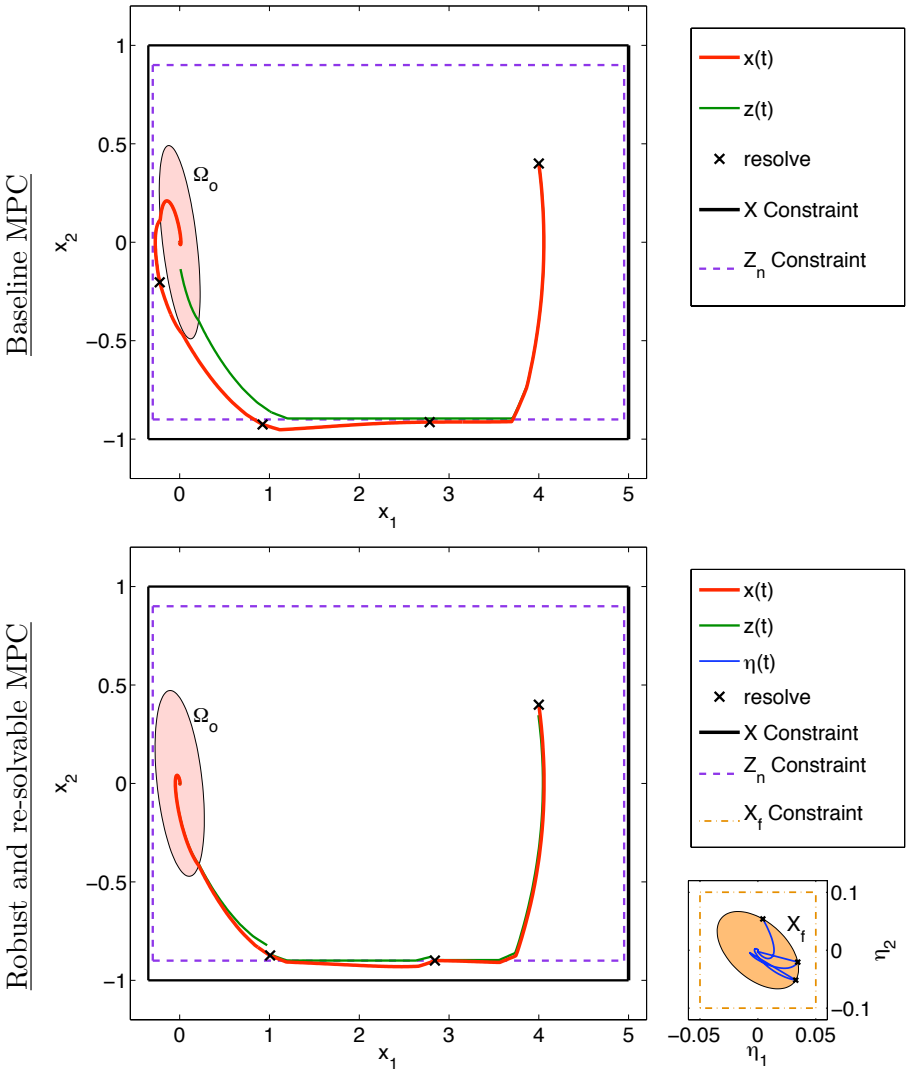


Figure 3.6: An improved nominal model with $\omega_0 = 0.2$ helps baseline MPC feasibility (top), while the R-MPC algorithm (bottom) performs well with either model.

3.5 Summary of the R-MPC Algorithm

The R-MPC algorithm enhances the baseline MPC in Chapter 2 by augmenting the framework with a separate feedback control component that provides robustness to uncertainty and disturbances affecting the actual dynamics. The feedback component creates an invariant tube about the nominal feedforward trajectory, based on an explicit characterization of the uncertainty between the actual system and the nominal model. Further, the online optimization for computing the feedforward control component utilizes the existence of the tube to relax the initial nominal state for each re-solve. This relaxation allows the initial nominal state to simply be within the tube cross section and not fixed to the actual state at the re-solve time, which provides robust feasibility and re-solvability to the R-MPC algorithm.

The R-MPC algorithm assumes that the state constraints imposed in the control objectives are known perfectly. In cases where uncertainty exists in these constraints, the R-MPC re-solvability cannot be guaranteed. For instance, if the state constraints change such that they intersect the invariant tube, the R-MPC algorithm has no means of ensuring re-solvability or avoidance of the changed constraint. The following chapter augments the R-MPC algorithm with a reactive safety mode that addresses changes to state constraints by switching on, if needed, an alternate control policy (called the safety mode) to maintain the system in an invariant set for all time that ensures avoidance of the changed constraint.

Chapter 4

A Reactive Safety Mode Blended into R-MPC

Application of the R-MPC (Robust and re-solvable MPC) algorithm developed in Chapter 3 provides control inputs that ensure re-solvability, robustness, and asymptotic stability for general classes of uncertain, nonlinear systems. However, the R-MPC algorithm does not handle uncertainty in static state constraints (or control constraints, which are not considered herein) and assumes that all state constraints are known perfectly ahead of the time. This chapter addresses uncertainty in static state constraints by augmenting the R-MPC algorithm with a reactive safety mode. The resultant continuous-time SR-MPC (Safety-mode augmented R-MPC) algorithm blends two operational modes: (I) *standard* mode guarantees re-solvability and asymptotic convergence to the origin in a robust receding-horizon manner; (II) *safety* mode, if activated, guarantees containment within an invariant set about a safety reference for all time. The standard mode is a modified version of the R-MPC algorithm that provides safety-mode availability at any time, and the safety mode is a separate control policy computed offline.

An assumption is made that perfect state knowledge is available inside and on the boundary of the invariant safety set so that any static state-constraint changes outside the boundary, once realized, can be handled by safety-mode activation. Technically, changes in static state constraints that do not affect the standard-mode tube trajectory, either current or on a re-solve, will not violate feasibility and could be disregarded with additional checks added into the SR-MPC algorithm. However, the most conservative option is to activate the safety mode for any static state-constraint change. The safety control policy is considered reactive in the sense that onboard information is not utilized proactively to alter the policy;

in this sense, the safety policy provides additional conservatism to trajectory generation. After safety-mode activation, higher-level algorithms can either change the control objective or re-establish initial feasibility for resumption of standard mode operations; this is not addressed within this thesis.

As with the R-MPC algorithm, the SR-MPC algorithm is also developed for general continuous-time, uncertain nonlinear systems, with design methods for a special class of uncertain, nonlinear systems with derivatives in convex sets. Additionally, the invariant safety set is prescribed based on specified state constraints in addition to a characterization of the uncertainty/nonlinearity.

The formulation for the continuous-time SR-MPC algorithm is the primary contribution of this thesis and will appear in [11]. A discrete version of the algorithm, useful for practical applications will follow in Chapter 5. The notion of a safety-augmented robust MPC algorithm applied to discrete, linear time-invariant systems was introduced by Schouwenaars *et al.* [43] and made part of the RSBK (Robust Safe But Knowledgeable) algorithm by Kuwata *et al.* [24]. These algorithms assume perfect state-constraint knowledge during the current planning horizon and were developed to address the feasibility issue of re-solves with a shifted terminal set at the end of the planning horizon, which is required in applications where the planning horizon is shorter than required for the system to reach the intended target state (e.g., the origin is outside of the planning horizon). If a re-solve is infeasible to a shifted terminal set, then the prior feasible solution would provide a feedforward trajectory that ends in a control-invariant safety set about the prior target state.

The objective of the SR-MPC algorithm as stated above is to address state constraint uncertainty during the planning horizon, which is accomplished by building in safety-mode availability at any time along the current time horizon, rather than at just the end. The SR-MPC algorithm essentially adds a safety tube, separate from the feedback invariant tube inherited from R-MPC, in which the actual system can enter the safety mode. The trade off is in conservatism during the planning horizon; by addressing state-constraint uncertainty, the SR-MPC algorithm will select only paths with state corridors that allow reactive safety-mode activation. In contrast, the RSBK algorithm with perfect state-constraint knowledge will allow tighter paths provided the planned trajectory ends in a safety set.

4.1 System Description and Extended Control Objective

The SR-MPC algorithm is developed for the same system definitions as in Chapter 3 for R-MPC but the control objective differs. The system definitions are repeated here for completeness of the SR-MPC chapter.

Consider the following uncertain, nonlinear dynamical system as the *actual* system for application of SR-MPC:

$$\dot{x} = f(x, u, t), \tag{4.1}$$

with actual state $x \in \mathbb{R}^n$ and control input $u \in \mathbb{R}^m$. Let a *nominal* system model of the actual system (4.1) be given by

$$\dot{z} = F(z, u_o, t), \tag{4.2}$$

with nominal state $z \in \mathbb{R}^n$ and control input $u_o \in \mathbb{R}^m$, and where $F(\cdot)$ is a known, approximate model of $f(\cdot)$ from (4.1). Without a loss of generality, the origin $x = z = 0$ is considered an equilibrium point shared by both systems; $f(0, 0, t) = F(0, 0, t) = 0$.

The control objective is to obtain a control input u that, when applied to the actual system (4.1), achieves the following closed-loop system responses in each mode:

I. *standard* mode: the actual system origin ($x = 0$) is asymptotically stable, with a region of attraction $\mathcal{R}_a \subseteq \mathbf{X}$, such that

$$x(t) \in \mathbf{X} \text{ and } u(t) \in \mathbf{U}\,, \ \forall t \geq t_0, \tag{4.3}$$

when $x(t_0) \in \mathcal{R}_a$.

II. *safety* mode: the actual system trajectory x is contained within an invariant set $\mathbf{X}_s$ about a fixed reference r_s such that

$$\tilde{x}(t) \in \mathbf{X}_s, u(t) \in \mathbf{U}, \text{ and } x(t) \in \mathbf{X}, \ \forall t \geq t_s, \tag{4.4}$$

where $\tilde{x}(t) \triangleq x(t) - r_s$, and $t_s \geq t_0$.

Sets $\mathbf{X} \subseteq \mathbb{R}^n$ and $\mathbf{U} \subseteq \mathbb{R}^m$ define actual state and control constraints, respectively: $\mathbf{X}$ is connected and contains the origin in its interior; sets $\mathbf{X}_s$ and $\mathbf{U}$ are compact and contain the origin in their interiors. Set $\mathbf{X}_s \subseteq \mathbb{R}^n$ defines the desired safety state constraints for the

actual system, and reference r_s will be defined within the SR-MPC algorithm (basically, vector r_s will be such that $\{r_s\} + \mathbf{X}_s \subseteq \mathbf{X}$). Additionally, set $\mathcal{R}_a$ will be redefined based on the architecture of the SR-MPC algorithm.

4.2 Architecture of the SR-MPC Algorithm

The control approach builds on the R-MPC algorithm from Chapter 3 where control u is given by

$$u(t) = u_o(t) + u_f(t), \tag{4.5}$$

where u_o and u_f are feedforward and feedback components, respectively. In standard mode, the feedforward component u_o comes from online solution of an FHC-S (Finite Horizon optimal Control problem for SR-MPC) that utilizes the nominal system model (4.2), and like R-MPC the feedback component u_f is determined offline as a control policy to handle a characterization of the uncertainty in the actual system (4.1). In safety mode, the entire control input is from a policy designed offline.

Sets $\mathbf{X}$, $\mathbf{U}$, and $\mathbf{X}_s$ are given constraints imposed on the overall design of the control input. The following additional constraint sets (all containing the origin in their interiors), are used in constructing the control approach:

$$\mathbf{U}_o + \mathbf{U}_f \subseteq \mathbf{U}, \quad \mathbf{Z}_n + \mathbf{X}_s \subseteq \mathbf{X}, \quad \text{and} \quad \mathbf{Z}_s + \mathbf{X}_f \subseteq \mathbf{X}_s. \tag{4.6}$$

Set $\mathbf{Z}_n$ is connected and contains the origin in its interior, and sets $\mathbf{Z}_s$, $\mathbf{X}_f$, $\mathbf{U}_o$, and $\mathbf{U}_f$ are compact and contain the origin in their interiors.

In preview, for standard mode the SR-MPC algorithm uses online solution of the FHC-S to design feedforward control input $u_o \in \mathbf{U}_o$ to maintain the nominal states within constraint set $\mathbf{Z}_n$; a special constraint involving $\mathbf{Z}_s$ in the FHC-S ensures safety-mode availability from standard mode at any time. Then, feedback policy $u_f \in \mathbf{U}_f$ is designed as in R-MPC to establish invariant tube $\mathbf{X}_f$ about the nominal trajectory (utilized as a feedforward, guidance trajectory) to maintain the actual states in the proximity of the nominal states (providing robustness to dynamics uncertainty and disturbances). Finally, the safety-mode control policy is designed to establish invariant set $\mathbf{X}_s$ about any nominal state from the FHC-S solution so that the actual state remains in $\mathbf{X}_s$ after safety-mode activation (providing ro-

bustness to static state-constraint uncertainty). The set relationships in (4.6) ensure that the overall design constraints are not violated.

Remark 4.1 (Notation for Nominal State in Standard and Safety Modes). The variable z denotes the nominal state in both standard and safety modes. To avoid confusion in safety mode, if a standard-mode nominal state is referenced, that state will be denoted as z_{FHC}, implying that it comes from a standard-mode solution of the FHC-S. $\diamond$

4.2.1 Standard-Mode Control

The following FHC-S builds upon the FHC-R of Chapter 3 by adding a constraint on the nominal trajectory that ensures safety-mode availability at any time. Online solution of the FHC-S generates feedforward u_o for standard mode (Control Objective I).

FHC-S (for Safety-mode augmented R-MPC)

Find $J^* = \min_{u_o} J(z, u_o;\ t_i, T, z(t_i))$ where

$$J(z, u_o;\ t_i, T, z(t_i)) = \int_{t_i}^{t_i+T} h(z(\tau), u_o(\tau))d\tau + V(z(t_i+T))$$

subject to

$$\left.\begin{aligned} &\dot{z} = F(z, u_o, t), \\ &z(t) \in \mathbf{Z}_n, \\ &u_o(t) \in \mathbf{U}_o, \\ &z(t) - \mathcal{T}(z(t)) \in \mathbf{Z}_s, \end{aligned}\right\} \forall\, t \in [t_i,\, t_i+T]$$

$$z(t_i+T) \in \mathbf{\Omega}_o,$$

$$x(t_i) - z(t_i) \in \mathbf{X}_f,$$

where $x(t_i)$ is the actual system (4.1) state at initial time t_i.

The set $\mathcal{R}_a$, which specifies the region of attraction for Control Objective I, is defined based on feasibility of the FHC-S:

$$\mathcal{R}_a = \{\xi \in \mathbf{Z}_n + \mathbf{X}_f :\ \text{FHC-S is feasible with } x(t_i) = \xi\}\,. \tag{4.7}$$

The baseline MPC and R-MPC algorithm Conditions 2.1, 2.2, 3.1, 3.2, and 3.3 are also conditions for proving asymptotic stability of the SR-MPC algorithm. Refer back to Chapters 2 and 3 for a full explanation of the conditions, which define cost function $h(\cdot)$, establish set relationships for $\mathbf{X}_f$ and $\mathbf{\Omega}_o$, and establish Control Lyapunov Function V for both the actual and nominal systems with a local controller inside invariant terminal set $\mathbf{\Omega}_o$. Condition 3.1 is restated here for convenience in forthcoming discussions.

Condition 4.1. There exists a feedback control policy $u_f = \mathcal{K}_f(x, z) \in \mathbf{U}_f$ in (4.5) that renders set $\mathbf{X}_f$ invariant for $\eta(t) \triangleq x(t) - z(t) \in \mathbf{X}_f$ and for all $u_o(t)$, $\forall t \geq t_0$, with dynamics (4.1) for x and (4.2) for z. ◇

Note, Condition 4.1 defines set $\mathbf{X}_f$ as an invariant tube about the nominal states z: if $\eta(t_0) \in \mathbf{X}_f$ for some $t_0 \geq 0$, then $\eta(t) \in \mathbf{X}_f, u_f(t) \in \mathbf{U}_f, \forall t \geq t_0$.

The form of the FHC-S is identical to the FHC-R with the addition of the constraint

$$z(t) - \mathcal{T}(z(t)) \in \mathbf{Z}_s, \ \forall\, t \in [t_i,\, t_i + T] \tag{4.8}$$

where function $\mathcal{T} : \mathbf{Z}_n \to \mathbf{Z}_n$ defines a mapping ($\mathcal{T}(\mathbf{Z}_n) \subseteq \mathbf{Z}_n$) that will be used in the safety subsection to define the safety reference r_s. A feasible FHC-S provides nominal trajectories that satisfy this constraint at all times, which will be used in the proof of safety-mode availability in the SR-MPC algorithm.

4.2.2 Reactive Safety-Mode Control

Control $u(\triangleq u_s)$ in safety mode (Control Objective II) comes from an offline design that generates an invariant set $\mathbf{X}_s$ to maintain $x \in \mathbf{X}_s$ about reference r_s for all time after safety activation. Reference r_s is defined with function $\mathcal{T}$ from the FHC-S that maps the standard-mode nominal state $z_{\text{FHC}}(t_s)$ to a desired safety reference state.

Definition 4.1 (Safety Reference)**.** The *safety reference* is fixed at safety-activation time t_s such that

$$r_s = \mathcal{T}(z_{\text{FHC}}(t_s)) \in \mathbf{Z}_n \tag{4.9}$$

where $\mathcal{T} : \mathbf{Z}_n \to \mathbf{Z}_n : z(t_s) \mapsto r_s$. ◇

For example, a mechanical system with non-zero position and non-zero velocity at safety activation t_s may desire r_s to be rest (zero velocity) at the current non-zero position.

The following condition is useful for proving the satisfaction of the safety-mode control objective upon safety-mode activation with the SR-MPC algorithm.

Condition 4.2. There exists control law $u_s = \mathcal{K}_s(t, x, r_s) \in \mathbf{U}$ that renders set $\mathbf{X}_s$ invariant for $\tilde{x}(t) \triangleq x(t) - r_s \in \mathbf{X}_s$, $\forall t \geq t_s$ with dynamics (4.1) for x and $r_s \in \mathbf{Z}_n$. $\diamond$

4.2.3 The SR-MPC Algorithm

The following algorithm describes the SR-MPC approach.

SR-MPC Algorithm

Begin at $k = 0$ in standard mode with $x(t_0) \in \mathcal{R}_a$ and iterate the following steps over computation times t_k for $k \in \mathbb{Z}^+$:

standard mode

1. Measure state $x(t_k)$ of actual system (4.1).
2. Solve the FHC-S at time $t_i = t_k$ with $T = T_k$ to obtain $u_o^k(t)$ on $t \in [t_k, t_k + T_k]$.
3. Monitor $z(t)$ and $x(t)$ while applying $u_o(t) = u_o^k(t)$ to nominal system (4.2) and $u(t) = u_o^k(t) + u_f(t)$ to actual system (4.1) on $t \in [t_k, t_{k+1}]$, with $z(t) = z_{\text{FHC}}(t) = z^k(t)$ and $u_f(t) = \mathcal{K}_f(x(t), z(t))$.
4. Check the following over $t \in [t_k, t_{k+1}]$:
   (a) If safety event detected at $t_s \geq t_k$, set $r_s = \mathcal{T}(z_{\text{FHC}}(t_s))$, then switch to safety mode and stop iteration.
   (b) if $z(\tilde{t}) \in \mathbf{\Omega}_o$ for some $\tilde{t} \geq t_0$, then set $u_o(t) = \mathcal{L}(z(t)), \forall t \geq \tilde{t}$ and skip step 2 in iteration.
   (c) if $x(\bar{t}) \in \mathbf{\Omega}_o$ for some $\bar{t} \geq t_0$, then set $u(t) = \mathcal{L}(x(t)), \forall t \geq \bar{t}$ and stop iteration.

safety mode

For $t \geq t_s$, apply $u(t) = u_s(t) = \mathcal{K}_s(t, x(t), r_s)$ to actual system (4.1).

Remark 4.2 (Safety and Terminal Set). An assumption is made that once the terminal set $\mathbf{\Omega}_o$ is entered by the actual system, then the system will not require safety mode. $\diamond$

For proving stability of the standard-mode portion of SR-MPC algorithm, the sequence of monotonically increasing computation times, as defined in Definition 2.1, for the baseline MPC algorithm is used. Lemmas 3.1 and 2.2 apply directly to the SR-MPC algorithm for establishing re-solvability of the FHC-S (Lemma 3.1) and shrinking optimal cost in a compressing- or receding-horizon implementation (Lemma 2.2). No changes are necessary for the added FHC-S constraint (4.8) on the nominal trajectory, and proofs of the lemmas applied to the SR-MPC algorithm follow identically with those from the R-MPC algorithm. Refer back to the comments in the proof of Lemma 3.1 for application to the FHC-S, along with the comments prior to the R-MPC algorithm Theorem 3.1 that discusses the minor modifications to the proof of Lemma 2.2 for R-MPC.

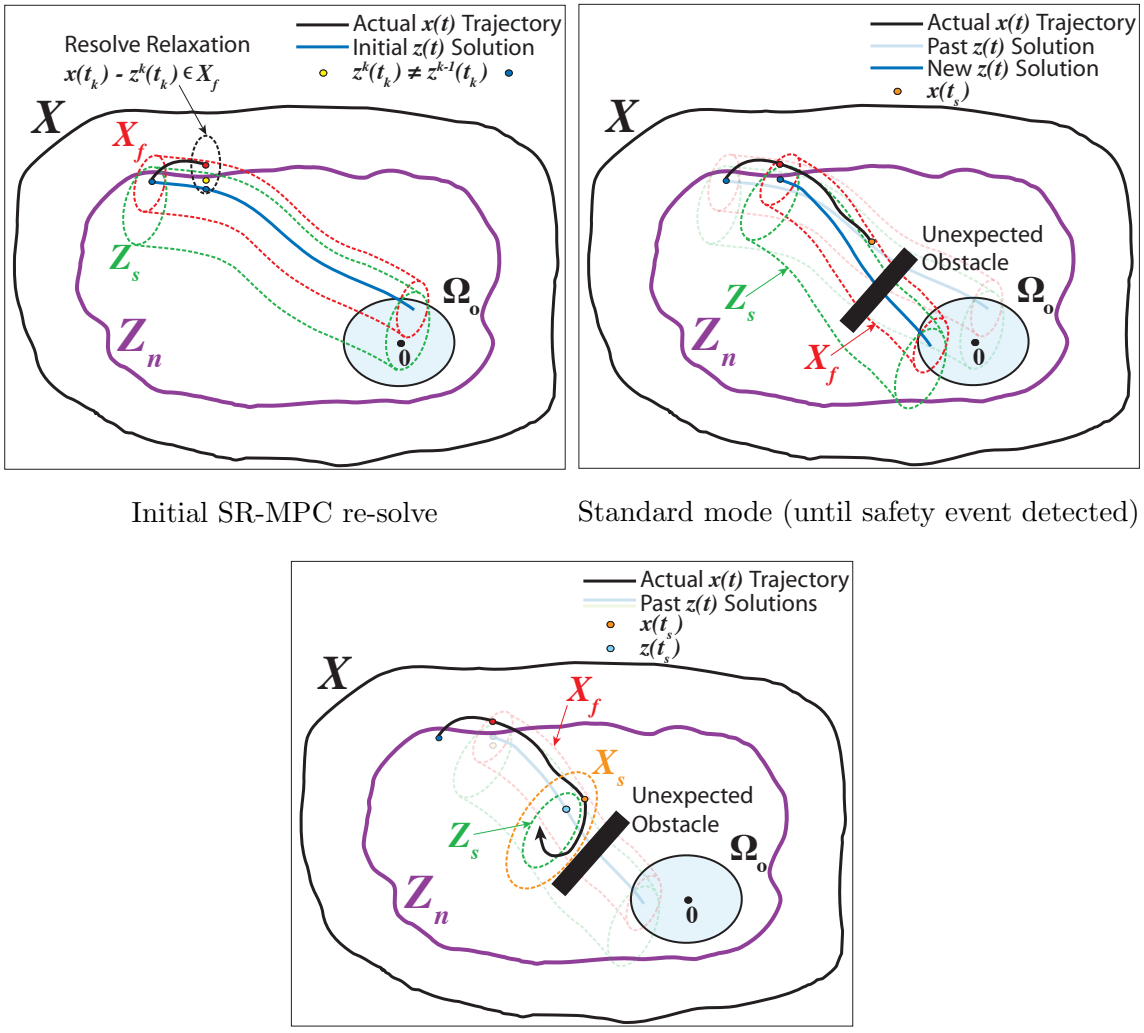
The following theorem establishes closed-loop asymptotic stability for the standard-mode portion of the SR-MPC algorithm, along with safety-mode availability and invariance. Following the proof, a pictorial sketch of safety-mode activation with the SR-MPC algorithm appears in Figure 4.1; a sketch of the SR-MPC standard mode is already provided in Figure 3.2 for R-MPC.

Theorem 4.1. *Consider system (4.1) for x with a control input u described by the SR-MPC algorithm. If Conditions 2.1, 2.2, 3.2, 3.3, 4.1, and 4.2 are satisfied, then the resulting closed-loop system satisfies Control Objectives I and II.* ◇

Proof. The proof is split into two pieces

I. standard mode: The proof of asymptotic stability with region of attraction $\mathcal{R}_a$ from (4.7) is identical to the proof of Theorem 3.1 for the R-MPC algorithm in Chapter 3.

II. safety mode: The control input in standard mode guarantees that $x(t_s) - z(t_s) \in \mathbf{X}_f$ since tube $\mathbf{X}_f$ is invariant under application of feedback u_f from Condition 4.1. Further, the FHC-S is satisfied in standard mode, thus constraint (4.8) guarantees that $z(t_s) - r_s = z(t_s) - \mathcal{T}(z(t_s)) \in \mathbf{Z}_s$ with r_s from (4.9). Thus, $x(t_s) - r_s = (x(t_s) - z(t_s)) + (z(t_s) - r_s) \in \mathbf{X}_f + \mathbf{Z}_s \subseteq \mathbf{X}_s$ where the set definition in (4.6) for $\mathbf{X}_s$ is used. Now, by using Condition 4.2, application of safety-mode control input $u = u_s = \mathcal{K}_s(t, x, r_s)$ ensures $\tilde{x}(t) = x(t) - r_s \in \mathbf{X}_s$ for all $t \geq t_s$. Further, since $r_s \in \mathbf{Z}_n$, then $\{r_s\} + \mathbf{X}_s \subseteq \mathbf{X}$ and $x(t) = r_s + \tilde{x}(t) \in \mathbf{X}$ for all $t \geq t_s$, which satisfies Control Objective II. □



Initial SR-MPC re-solve

Standard mode (until safety event detected)

Safety mode activated

Figure 4.1: Sketch of the SR-MPC algorithm activating safety mode due to a static-constraint violation.

Figure 4.1 depicts how FHC-S constraint (4.8) involving set $\mathbf{Z}_s$ provides a second tube that ensures safety-mode availability from all nominal states. The second tube is arbitrarily shown larger than tube $\mathbf{X}_f$ and containing, but offset from, the nominal trajectory. The tube for $\mathbf{Z}_s$ will actually be centered on $r_s = \mathcal{T}(z) \in \mathbf{Z}_n$ in standard mode, so the sketch is a representative example for some function $\mathcal{T}$. The idea of the sketch is to illustrate that when safety mode is activated that the actual system dynamics will be maintained within invariant set $\mathbf{X}_s$, where $\mathbf{Z}_s + \mathbf{X}_f \subseteq \mathbf{X}_s$ per (4.6). Further, from (4.6) the full state constraints $\mathbf{X}$ will not be violated since $\mathbf{Z}_n + \mathbf{X}_s \subseteq \mathbf{X}$.

4.3 Specialization to a Class of Systems with Derivatives Contained in Convex Sets

This section enhances the R-MPC specialization in Section 3.3 by adding components for SR-MPC safe mode. Explicit design methods are developed that satisfy the safety-mode Condition 4.2 for two further subclasses of the systems considered in Section 3.3. Corollary 3.1.1 for R-MPC applies directly to the SR-MPC standard mode, with the substitution of 'SR-MPC algorithm' in place of 'R-MPC algorithm' in the Corollary statement. The proof of Corollary 3.1.1 when applied to SR-MPC standard mode follows identically, and the FHC-S feasibility ensures satisfaction of constraint (4.8) for safety-mode availability at any time.

The form of this special class of systems is repeated here for convenience. The following characterizes the actual system:

$$\begin{aligned} \dot{x} &= Ax + Bu + E\phi(t, q), \\ q &= C_q x + D_q u, \end{aligned} \tag{4.10}$$

where $\phi : \mathbb{R} \times \mathbb{R}^{n_q} \to \mathbb{R}^{n_p}$ with $\phi(t, 0) = 0, \forall t$, is a continuously differentiable function representing the uncertain nonlinear part of the dynamics. This form implies $f(x, u, t) = Ax + Bu + E\phi(t, q)$ in (4.1) with $q \in \mathbb{R}^{n_q}$, $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $E \in \mathbb{R}^{n \times n_p}$, $C_q \in \mathbb{R}^{n_q \times n}$, and $D_q \in \mathbb{R}^{n_q \times m}$.

The nominal model dynamics is assumed to have the following form:

$$\begin{aligned} \dot{z} &= Az + Bu_o + E\psi(t, q_o), \\ q_o &= C_q z + D_q u_o, \end{aligned} \tag{4.11}$$

where $\psi : \mathbb{R} \times \mathbb{R}^{n_q} \to \mathbb{R}^{n_p}$ with $\psi(t, 0) = 0, \forall t$, is an approximation for $\phi(\cdot)$ in actual system (3.10). Thus, $F(z, u_o, t) = Az + Bu_o + \psi(t, q_o)$ in (4.2), and $q_o \in \mathbb{R}^{n_q}$.

The R-MPC conditions of Section 3.3 still apply to nonlinear functions $\phi(\cdot)$ and $\psi(\cdot)$ in (4.10) and (4.11), respectively: the functions are assumed to have Jacobians in convex sets (Condition 3.4), along with a bounded mismatch (Condition 3.5).

The *error* dynamics between the actual and nominal states from (4.10) and (4.11),

respectively, are

$$\dot{\eta} = A\eta + Bu_f + E\pi(t, \eta, u_f) + Ew(t, z, u_o), \tag{4.12}$$

where $\eta \triangleq x - z$ is the error state, $u_f \triangleq u - u_o$ defines the feedback input, $w(t, z, u_o) = \phi(t, q_o) - \psi(t, q_o)$ is from Condition 3.5, and $\pi(t, \eta, u_f) = \phi(t, C_q x + D_q u) - \phi(t, C_q z + D_q u_o)$.

The design specifications for the state and control constraints add onto the specifications used for R-MPC in Condition 3.6, with an addendum for the safety state specifications required in SR-MPC. An ellipsoidal description is used to specify the safety state constraints.

Condition 4.3 (State and Control Constraints)**.**

$$\begin{aligned}
\mathbf{Z}_n \supseteq \mathcal{Z}_\Omega &\triangleq \{z \in \mathbb{R}^n : \ a_i^T z \leq 1, \ i = 1, \ldots, m_o\}, \\
\mathbf{X}_f \subseteq \mathcal{X}_f &\triangleq \{\eta \in \mathbb{R}^n : \ b_i^T \eta \leq 1, \ i = 1, \ldots, m_f\}, \\
\mathbf{Z}_s \subseteq \mathcal{Z}_s &\triangleq \{\tilde{z} : \tilde{z}^T C_s^T \Pi_s C_s \tilde{z} \leq 1\}, \\
\mathbf{U}_o \subseteq &\{u_o \in \mathbb{R}^m : \ u_o^T \Pi_o u_o \leq 1\}, \\
\mathbf{U}_f \subseteq &\{u_f \in \mathbb{R}^m : \ u_f^T \Pi_f u_f \leq 1\},
\end{aligned}$$

where Π_o, Π_f, and Π_s are symmetric positive-definite matrices, matrix C_s allows for safety specification on a linear combination or portion of state $\tilde{z}$, and the design specifications for sets $\mathcal{X}_f$ and $\mathcal{Z}_s$ are determined such that $\mathcal{Z}_s + \mathcal{X}_f \subseteq \mathbf{X}_s$: thus, $\mathbf{X}_f \subseteq \mathcal{X}_f$ and $\mathbf{Z}_s \subseteq \mathcal{Z}_s$ result in additional design conservatism to ensure the constraints in (4.6) are satisfied. $\diamond$

Note, specifying safety constraint $\mathbf{Z}_s$ in terms of a portion of state $\tilde{z}$ with matrix C_s, as in Condition 4.3, can be useful in many practical applications; e.g., vehicles with relative-position sensors may only require safety in terms of relative distance to another object.

The safety mode is now developed for two special subclasses of the actual and nominal systems in (4.10) and (4.11), respectively. The special cases presented herein are motivated by practical application of the SR-MPC algorithm. The following form for nominal system (4.11) in terms of safety state $\tilde{z} \triangleq z - r_s$, with z the state of the nominal system, is used in the design of a nominal control policy u_{os} for safety mode:

$$\begin{aligned}
\dot{\tilde{z}} &= A\tilde{z} + Ar_s + Bu_{os} + E\psi(t, q_o) \\
q_o &= C_q \tilde{z} + D_q u_{os} + C_q r_s.
\end{aligned} \tag{4.13}$$

4.3.1 Subclass I (contains velocity-dependent nonlinearity)

The following condition and corollary provide the safety-mode component of Theorem 4.1 that is representative, for example, of mechanical systems that can come to rest at arbitrary positions and that have velocity-dependent nonlinearities (e.g., hovercraft/road vehicles with velocity-dependent drag).

Condition 4.4. Safety reference r_s satisfies

$$r_s \in \mathcal{N}(A) \cap \mathcal{N}(C_q), \tag{4.14}$$

where $\mathcal{N}(X)$ is the null-space of a matrix X. ◇

Reminder, safety reference r_s is defined in (4.9), based on the mapping $\mathcal{T}$ of the nominal state at safety-activation time. The above condition further implies that $\mathcal{T}$ is such that $Ar_s = 0$ and $C_q r_s = 0$ in system (4.13).

Corollary 4.1.1. *Consider a class of systems modeled by (4.11) with r_s satisfying Condition 4.4. Suppose there exist matrices $S = S^T > 0$ and R and positive scalar β satisfying the following linear matrix inequalities:*

$$\begin{bmatrix} SA^T + AS + BR + R^T B^T + \beta EE^T & SC_q^T + R^T D_q^T \\ C_q S + D_q R & -\beta I \end{bmatrix} \leq 0, \tag{4.15}$$

$$\begin{bmatrix} S & SC_s^T \\ C_s S & \Pi_s^{-1} \end{bmatrix} \geq 0, \text{ and } \begin{bmatrix} S & R^T \\ R & \Pi_o^{-1} \end{bmatrix} \geq 0\,. \tag{4.16}$$

If safety-mode control $u_s = \mathcal{K}_s(t, x, r_s) \in \mathbf{U}$ is given by

$$\mathcal{K}_s(t, x, r_s) = K_s(z - r_s) + K_f(x - z), \quad K_s = RS^{-1}, \tag{4.17}$$

where $r_s = \mathcal{T}(z(t_s))$, and K_f and $\mathbf{X}_f$ obtained as described in Corollary 3.1.1 applied to SR-MPC, then $\mathbf{Z}_s = \{\tilde{z} : \tilde{z}^T S^{-1} \tilde{z} \leq 1\}$ satisfies the constraint in Condition 4.3, $\{r_s\} + \mathbf{Z}_s + \mathbf{X}_f$ is invariant for actual dynamics (4.10), and $\mathbf{Z}_s + \mathbf{X}_f \subseteq \mathbf{X}_s$. Further, $u_s = \mathcal{K}_s(t, x, r_s) \in \mathbf{U}$ for all $x \in \{r_s\} + \mathbf{Z}_s + \mathbf{X}_f$ and $r_s \in \mathbf{Z}_n$. ◇

Proof. Let positive-definite function $V_s(\tilde{z}) = \tilde{z}^T S^{-1} \tilde{z}$ be a Lyapunov function candidate.

Pre- and post-multiply (4.15) by $\mathbf{diag}(S^{-1}, I)$ and use $K_s = RS^{-1}$ from (4.17)

$$\begin{bmatrix} \begin{pmatrix} (A+BK_s)^T S^{-1} + S^{-1}(A+BK_s) \\ +\beta S^{-1} E E^T S^{-1} \end{pmatrix} & C_q^T + K_s^T D_q^T \\ C_q + D_q K_s & -\beta I \end{bmatrix} \leq 0.$$

Utilize Schur complements, and pre- and post-multiply by ζ^T and ζ, respectively, where $\zeta = (\tilde{z}^T, \psi^T)^T$:

$$\tilde{z}^T(A^T S^{-1} + S^{-1}A)\tilde{z} + 2\tilde{z}^T S^{-1}(Bu_{os} + E\psi) + \tfrac{1}{\beta}(q_o^T q_o - \psi^T \psi) \leq 0,$$

with $u_{os} = K_s\tilde{z}$ and $q_o = C_q\tilde{z} + D_q u_{os}$. Making use of (3.17) and Condition 3.4 gives

$$q_o^T q_o - \psi^T \psi \geq 0,$$

which implies

$$\tilde{z}^T(A^T S^{-1} + S^{-1}A)\tilde{z} + 2\tilde{z}^T S^{-1}(Bu_{os} + E\psi) \leq 0,$$

and hence $\dot{V}_s(\tilde{z}) \leq 0$. Thus, $\mathbf{Z}_s$ is an invariant set for $\tilde{z}$ [2, 21].

Pre- and post-multiply the first LMI in (4.16) by matrix $\mathbf{diag}(S^{-1}, I)$, use a Schur complement, and pre- and post-multiply by $\tilde{z}^T$ and $\tilde{z}$, respectively:

$$\tilde{z}^T C_s^T \Pi_s C_s \tilde{z} \leq \tilde{z}^T S^{-1} \tilde{z},$$

which implies $\mathbf{Z}_s \subseteq \mathcal{Z}_s$ from Condition 4.3.

Pre- and post-multiply the second LMI in (4.16) by $\mathbf{diag}(S^{-1}, I)$, use a Schur complement, and pre- and post-multiply by $\tilde{z}^T$ and $\tilde{z}$, respectively:

$$u_{os}^T \Pi_o u_{os} \leq \tilde{z}^T S^{-1} \tilde{z}$$

where $u_{os} = K_s\tilde{z}$ and $K_s = RS^{-1}$. Thus, for $\tilde{z} \in \mathbf{Z}_s$, $u_{os}^T \Pi_o u_s \leq 1$, so $u_{os} \in \mathbf{U}_o$, with $\mathbf{U}_o$ defined in Condition 4.3. Further, since $u_s = u_{os} + u_f$, where $u_f = K_f(x - z)$, and $u_f \in \mathbf{U}_f$ for all $x - z \in \mathbf{X}_f$ (as guaranteed by Corollary 3.1.1), the safety-mode control $u_s \in \mathbf{U}$, with $\mathbf{U}$ defined in (4.6). The remainder of this proof follows that of Theorem 4.1, part II. □

4.3.2 Subclass II (contains position-dependent nonlinearity)

The following conditions and corollary provide the safety-mode component of Theorem 4.1 that is representative, for example, of mechanical systems that have position-dependent nonlinearities that do not disappear when the system comes to rest at arbitrary positions (e.g., spacecraft hovering in a gravity field).

Condition 4.5. Safety reference r_s satisfies

$$r_s \in \mathcal{N}(A), \tag{4.18}$$

where $\mathcal{N}(X)$ is the null-space of a matrix X. ◇

The above condition implies that $\mathcal{T}$, which defines safety reference r_s in (4.9), is such that $Ar_s = 0$ in system (4.13).

Condition 4.6. There exists scalar $\delta > 0$ such that

$$\|\psi(t, z, u_o)\| \leq \delta, \quad \forall t, z \in \mathbf{Z}_n + \mathbf{Z}_s, u_o \in \mathbf{U}_o, \tag{4.19}$$

where function $\psi(\cdot)$ is from nominal system (4.13). ◇

Note, Condition 4.6, along with Condition 3.5, also imposes a bound on actual system nonlinearity ϕ; this bound is not necessary to establish the safety control policy.

Corollary 4.1.2. *Consider a class of systems modeled by (4.11) with Condition 4.6 bounding the nonlinearity and r_s satisfying Condition 4.5. Suppose there exist matrices $S = S^T > 0$ and R and positive scalar α satisfying the following matrix inequalities:*

$$\begin{bmatrix} SA^T + AS + BR + R^T B^T + \alpha S & E \\ E^T & -\frac{\alpha}{\delta^2} I \end{bmatrix} \leq 0, \tag{4.20}$$

$$\begin{bmatrix} S & SC_s^T \\ C_s S & \Pi_s^{-1} \end{bmatrix} \geq 0, \text{ and } \begin{bmatrix} S & R^T \\ R & \Pi_o^{-1} \end{bmatrix} \geq 0\,.$$

If safety-mode control $u_s = \mathcal{K}_s(t, x, r_s) \in \mathbf{U}$ is given by

$$\mathcal{K}_s(t, x, r_s) = K_s(z - r_s) + K_f(x - z), \quad K_s = RS^{-1}, \tag{4.21}$$

where $r_s = \mathcal{T}(z(t_s))$, and K_f and $\mathbf{X}_f$ obtained as described in Corollary 3.1.1 applied to SR-MPC, then $\mathbf{Z}_s = \{\tilde{z} : \tilde{z}^T S^{-1} \tilde{z} \leq 1\}$ satisfies the constraint in Condition 4.3, $\{r_s\} + \mathbf{Z}_s + \mathbf{X}_f$ is invariant for actual dynamics (4.10), and $\mathbf{Z}_s + \mathbf{X}_f \subseteq \mathbf{X}_s$. Further, $u_s = \mathcal{K}_s(t, x, r_s) \in \mathbf{U}$ for all $x \in \{r_s\} + \mathbf{Z}_s + \mathbf{X}_f$ and $r_s \in \mathbf{Z}_n$. $\diamond$

Proof. Let positive-definite function $V_s(\tilde{z}) = \tilde{z}^T S^{-1} \tilde{z}$ be a Lyapunov function candidate. Pre- and post-multiply (4.20) by $\mathbf{diag}(S^{-1}, I)$, use $K_s = R\bar{S}$ from (4.21), and then pre- and post-multiply by ζ^T and ζ, respectively, where $\zeta = (\tilde{z}^T, \psi^T)^T$:

$$\tilde{z}^T (A^T S^{-1} + S^{-1} A)\tilde{z} + 2\tilde{z}^T S^{-1}(B u_{os} + E\psi) + \alpha(\tilde{z}^T S^{-1} \tilde{z} - \tfrac{1}{\delta^2} \psi^T \psi) \leq 0,$$

with $u_{os} = K_s \tilde{z}$. Condition 4.6 ensures

$$\tfrac{1}{\delta^2} \psi^T \psi \leq \tilde{z}^T S^{-1} \tilde{z} \text{ when } \tilde{z}^T S^{-1} \tilde{z} \geq 1,$$

which implies that when $\tilde{z}^T S^{-1} \tilde{z} \geq 1$,

$$\tilde{z}^T (A^T S^{-1} + S^{-1} A)\tilde{z} + 2\tilde{z}^T S^{-1}(B u_{os} + E\psi) \leq 0,$$

and hence $\dot{V}_s(\tilde{z}) \leq 0$ when $\tilde{z}^T S^{-1} \tilde{z} \geq 1$. Thus, $\mathbf{Z}_s$ is an invariant set for $\tilde{z}$ [2]. The remainder of the proof follows that of Corollary 4.1.1. $\square$

4.4 An Illustrative Example to Compare the SR-MPC and R-MPC Algorithms

The SR-MPC and R-MPC algorithms are compared with the two-dimensional example from Section 3.4, which also satisfies Corollary 4.1.2. The actual and nominal system dynamics (3.10) and (3.11), respectively, for the example are repeated here for convenience:

$$\dot{x} = Ax + Bu + E\phi(t, q), \qquad q = C_q x + D_q u, \qquad \phi(t, q) = \omega(t) \sin^2(C_q x), \tag{4.22}$$

$$\dot{z} = Az + Bu_o + E\psi(t, q_o), \qquad q_o = C_q z + D_q u_o, \qquad \psi(t, q_o) = \omega_0 \sin^2(C_q z). \tag{4.23}$$

The state matrices and FHC-R/FHC-S cost function matrices are provided in (3.29), (3.30), and (3.31) within Section 3.4. In the simulations of this section, $\omega(t) \in [0, 0.5]$ and $\omega_0 = 0.2$,

which provides $\gamma = 0.3$ in Condition 3.5 for the bounded uncertainty between ϕ and ψ. Further, Condition 4.6 is satisfied with $\delta = 0.2$. The time horizon is fixed at $T = 30$ seconds, and the initial condition with $t_0 = 0$ is

$$x(t_0) = \begin{pmatrix} 4 \\ 0.4 \end{pmatrix}.$$

The actual state and control constraints remain the same, with the addition of the actual safety-state constraint:

$$\begin{aligned} \mathbf{X} &\triangleq \{x : x_1 \in [-0.35, 5], x_2 \in [-1, 1]\}, \\ \mathbf{U} &\triangleq \{u : \|u\| \leq 1.4\}, \\ \mathbf{X}_s &\triangleq \{\tilde{x}_1 : \tilde{x}_1 \in [-0.2, 0.2]\}, \end{aligned}$$

where $\tilde{x} = x - r_s$. The safety reference is chosen as rest at the nominal position at safety-activation time:

$$r_s = T z_{\text{FHC}}(t_s) = \begin{pmatrix} z_{\text{FHC},1}(t_s) \\ 0 \end{pmatrix}, \quad T = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \tag{4.24}$$

where $\mathcal{T}(\cdot) = T$ in (4.9) and $z_{\text{FHC},1}(t_s)$ is the first component of z_{FHC} (from standard mode) at safety activation time t_s. Note, r_s satisfies Condition 4.5.

The actual state and control constraints are partitioned into the design specifications from Condition 4.3 as follows:

$$\begin{aligned} \mathcal{Z}_\Omega : &\quad a_i = \left\{ \begin{pmatrix} \frac{1}{4.8} \\ 0 \end{pmatrix}, \begin{pmatrix} -\frac{1}{0.15} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \frac{1}{0.9} \end{pmatrix}, \begin{pmatrix} 0 \\ -\frac{1}{0.9} \end{pmatrix} \right\}, \\ \mathcal{X}_f : &\quad b_i = \left\{ \begin{pmatrix} \frac{1}{0.05} \\ 0 \end{pmatrix}, \begin{pmatrix} -\frac{1}{0.05} \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \frac{1}{0.1} \end{pmatrix}, \begin{pmatrix} 0 \\ -\frac{1}{0.1} \end{pmatrix} \right\}, \\ \mathcal{Z}_s : &\quad \Pi_s = \tfrac{1}{0.15^2}, \quad C_s = \begin{bmatrix} 1 & 0 \end{bmatrix}, \\ \mathbf{U}_o : &\quad \Pi_o = \tfrac{1}{1.2^2}, \\ \mathbf{U}_f : &\quad \Pi_f = \tfrac{1}{0.2^2}, \end{aligned}$$

where the a_i values define the full $\mathbf{Z}_n$ as well as $\mathcal{Z}_\Omega$ for the examples, and $i = \{1, 2, 3, 4\}$. Parameters Π_o and Π_f are identical to those in the prior R-MPC example and bound

components $\|u_o\| \leq 1.2$ and $\|u_f\| \leq 0.2$, respectively, in control u from (4.5). The above sets provide design specifications such that $\mathbf{Z}_n + \mathcal{Z}_s + \mathcal{X}_f \subseteq \mathbf{X}$, $\mathcal{Z}_s + \mathcal{X}_f \subseteq \mathbf{X}_s$, and $\mathbf{U}_o + \mathbf{U}_f \subseteq \mathbf{U}$.

Control design with Corollaries 3.1.1 and 4.1.2 ensures that the required control and state constraints are met by application of both R-MPC and SR-MPC algorithms. The LMIs in both corollaries are solved by using SDPT3 [50]. The solution provides the controller gains that define the feedback controller (K_f) for invariant tube $\mathbf{X}_f$, the local controller (K) for terminal set $\mathbf{\Omega}_o$, and the control policy (K_s) that is a part of the safety-mode control policy ($\mathcal{K}_s$ from (4.21)) for invariant set $\mathbf{X}_s$. The matrices P, Q, and S that define tube $\mathbf{X}_f$, terminal set $\mathbf{\Omega}_o$, and set $\mathbf{Z}_s$, respectively, are also part of the solution:

$$K_f = \begin{bmatrix} -3.5967 & -3.0595 \end{bmatrix}, \qquad P = \begin{bmatrix} 0.0011 & -0.0010 \\ -0.0010 & 0.0041 \end{bmatrix},$$

$$K = \begin{bmatrix} -5.4417 & -2.8751 \end{bmatrix}, \qquad Q = \begin{bmatrix} 0.0205 & -0.0206 \\ -0.0206 & 0.1461 \end{bmatrix},$$

$$K_s = \begin{bmatrix} -7.8642 & -2.8681 \end{bmatrix}, \qquad S = \begin{bmatrix} 0.0203 & -0.0371 \\ -0.0371 & 0.2050 \end{bmatrix}.$$

The R-MPC algorithm makes no consideration for uncertainty in static state constraints. Thus, a useful comparison of the algorithm with SR-MPC involves no safety activation, so only SR-MPC standard mode is compared (See Figure 4.2 below). The comparison demonstrates the effect of SR-MPC constraint (4.8) in the FHC-S, which provides safety-mode availability at any time during standard mode. Re-solves were performed on a fixed, 2-second interval in both simulations. The state constraints and markers for each re-solve are included in each plot as well.

As seen in Figure 4.2, both R-MPC and SR-MPC are robust to the dynamics uncertainty and are re-solvable due to application of feedback policy u_f to maintain the actual state within invariant tube $\mathbf{X}_f$ about the nominal trajectory. The velocity allowed by the SR-MPC algorithm (bottom plot) is significantly less than the velocity allowed by the R-MPC algorithm. This is an intuitive result for the particular system: to keep a system safe within a desired stopping distance, the maximum allowable velocity must be bounded. The conservatism of SR-MPC is solely the result of the safety constraint (4.8) built into the FHC-S. For this simple example, more conservative velocity bounds could have been

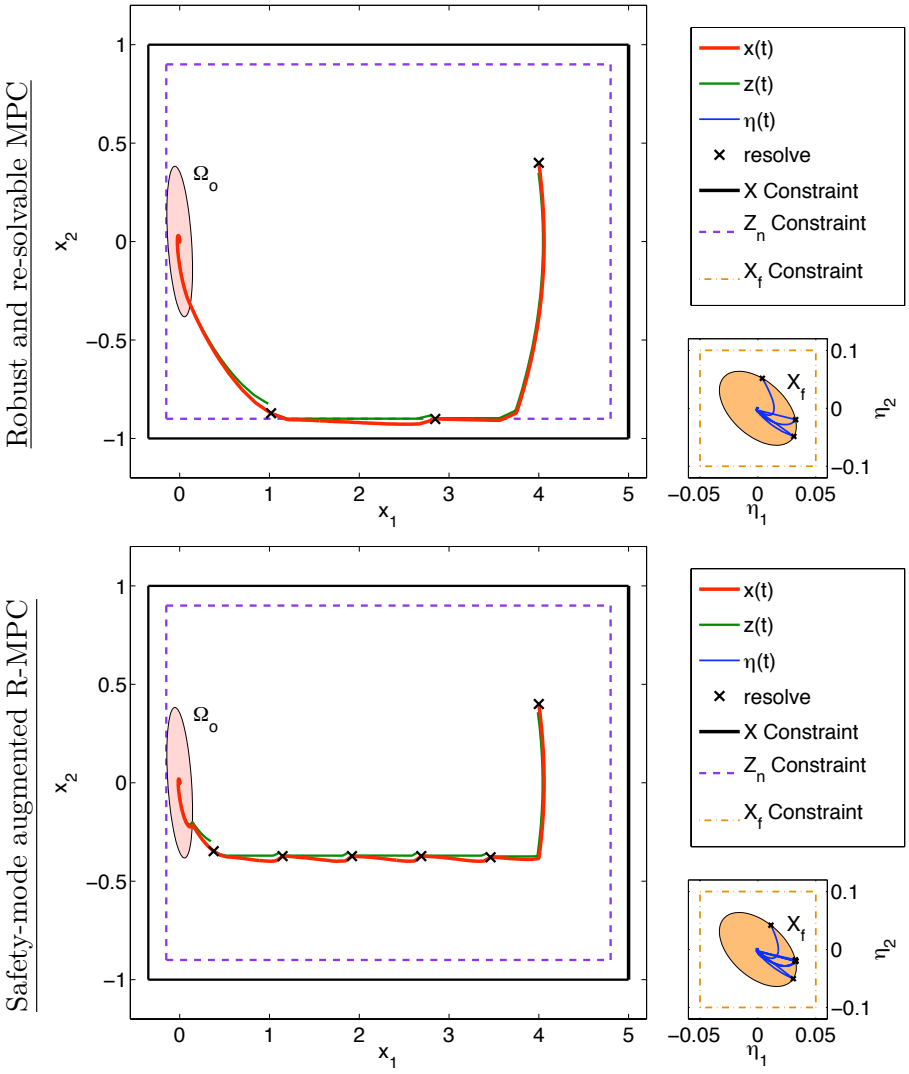


Figure 4.2: Comparison of R-MPC (top) and SR-MPC (bottom) algorithms with 2 second re-solves.

determined a priori without reliance on the FHC-S safety constraint. However, for more general examples, the safety constraint is invaluable because determination of appropriate conservative bounds can be quite difficult.

The slower velocity from applying the SR-MPC algorithm causes the actual system to take longer to enter terminal set $\mathbf{\Omega}_o$. This can be seen in the plots of Figure 4.3 that depict the control policies from the R-MPC (top) and SR-MPC (bottom) algorithms. The bump in the R-MPC control inputs (top plots) near 5 seconds indicates the switch to the local controller upon entry into $\mathbf{\Omega}_o$. The equivalent SR-MPC entry into $\mathbf{\Omega}_o$ occurs near 10 seconds. Note, both the R-MPC and SR-MPC algorithms achieve convergence with the

control policies obeying the specified control constraints, as seen in the plots.

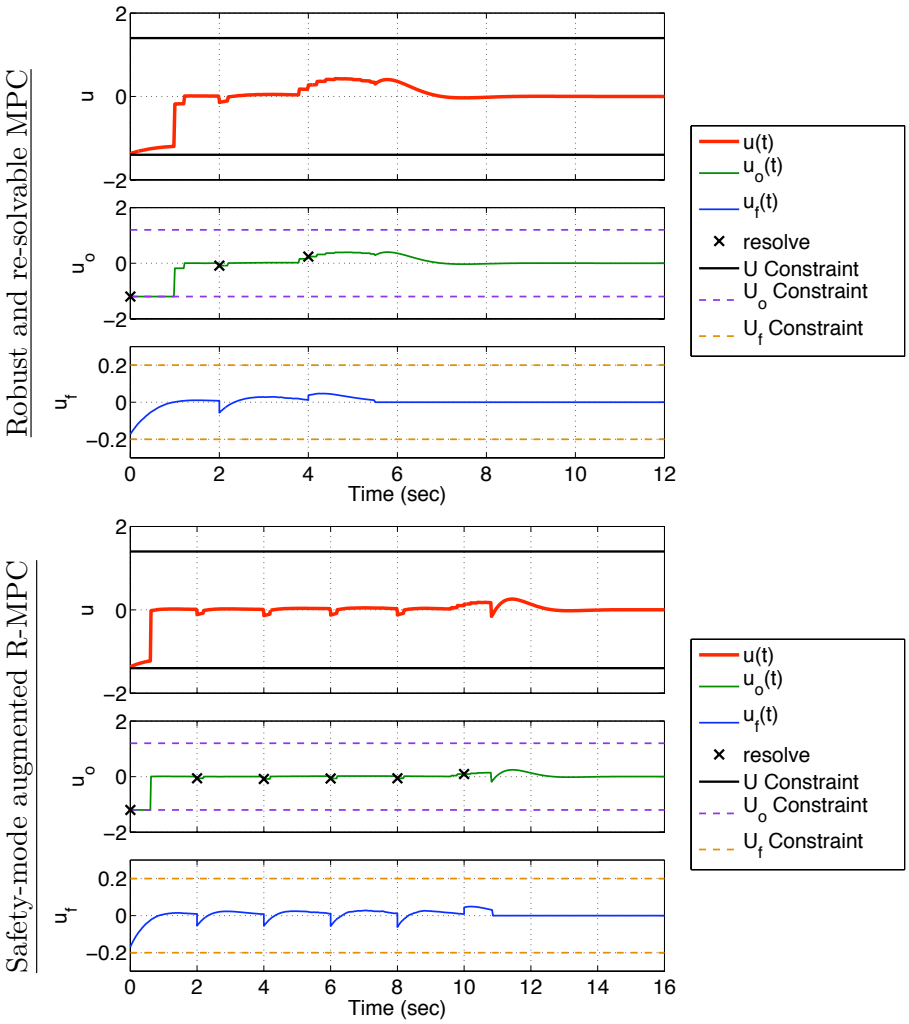


Figure 4.3: Control inputs from the R-MPC (top) and SR-MPC (bottom) algorithms.

The conservatism of SR-MPC algorithm slows down the standard-mode convergence toward set $\mathbf{\Omega}_o$, but the trade off is beneficial if the static state constraints are uncertain or an unknown static state constraint appears in the trajectory path during standard mode. Figure 4.4 shows the response of the SR-MPC algorithm when an unexpected obstacle in the path is detected just outside the safety constraints for $\mathbf{X}_s$; on a vehicle, this would assume that onboard sensors can detect just outside of the defined safety set. The SR-MPC algorithm is able to switch from standard mode into safety mode and maintain the actual state within the desired safety set, as indicated in the top figure; this is due to constraint (4.8) in the FHC-S. The safety reference is $r_s = (1.5493, 0)^T$, which comes from relationship (4.24) and the nominal state at safety-mode activation time t_s.

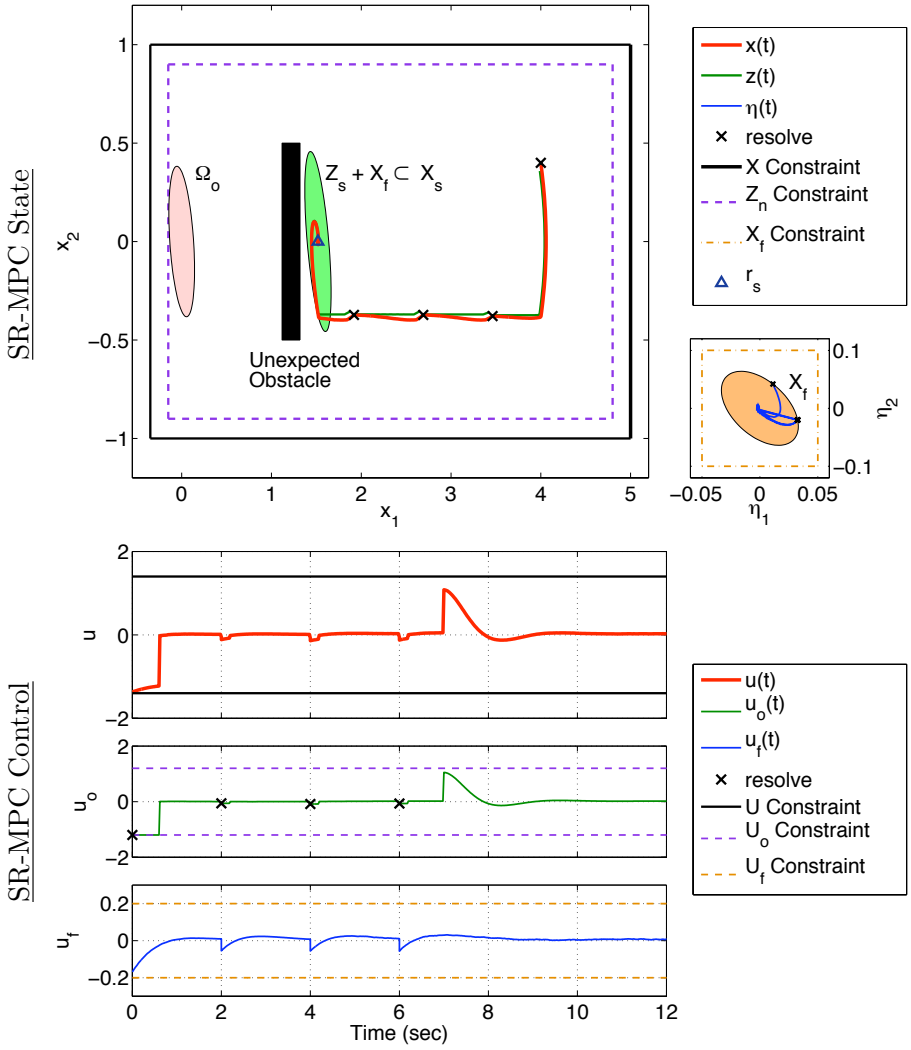


Figure 4.4: The SR-MPC algorithm activating safety mode upon detecting unexpected obstacle.

The unexpected obstacle is detected at $t_s = 7$ seconds, as seen by the significant jump in the control signals (bottom plots) at that time. The control inputs remain within the desired control constraints during standard mode and safety mode, including during the switch between the modes. This response was expected since it was a part of the SR-MPC algorithm construction through Corollaries 3.1.1 and 4.1.2. Note, the actual and nominal nonlinearities ϕ and ψ, respectively, are not 0 at the safety reference r_s, so a non-zero, although small, control input is required to hold the actual system near the reference; this is difficult to see in Figure 4.4.

4.5 Summary of the SR-MPC Algorithm

The SR-MPC algorithm builds upon the R-MPC algorithm in Chapter 3 by augmenting the framework with a separate, reactive safety mode that provides robustness (or safety) to changes in static state constraints. The standard mode implements a modified version of the R-MPC algorithm that further constrains the nominal states such that the separate reactive safety mode is available at any time. The safety mode, if needed, provides an alternate control policy to maintain the system in an invariant set to ensure avoidance of changed state constraints.

Implementation of the continuous-time SR-MPC algorithm in many practical applications requires the use of a computer. Given the computational limitations of real computers, implementation of continuous-time control algorithms can be unrealistic. The following chapter develops a computationally-efficient, discrete version of the SR-MPC algorithm that incorporates the innovations of both the standard and safety modes from the continuous-time SR-MPC algorithm.

Chapter 5

A Discrete SR-MPC Algorithm Applicable to Vehicle Control

Practical application of the SR-MPC algorithm onboard a robotic vehicle such as a spacecraft or aircraft requires the use of a computer for solving online the standard-mode feedforward finite-horizon optimal control problem. In addition, implementation on a computer requires some form of approximation of the continuous-time system dynamics and constraints in the optimization; thus, both accuracy and computational efficiency are important considerations in selecting the method of approximation. The contribution of this chapter is the development of a discrete SR-MPC algorithm that is well suited for online optimization onboard mechanical systems where the cost function is only on the control inputs.

The chosen approximation method for standard mode utilizes a combination of zero-order hold and linearization of the nonlinear functions to provide a convexification of the governing dynamics and control and state constraints for solving the feedforward optimization online; the convexification converts the optimization problem into a SOCP (Second-Order Cone Program) [9]. The resulting SOCP can be solved with interior-point algorithms in a computationally efficient manner that provides both constraint guarantees and convergence to within a specified accuracy in polynomial time [36, 48, 50]. This approach was chosen because it is well suited for the application example in Chapter 6 for autonomous-spacecraft proximity operations around small celestial bodies such as asteroids and comets. Other autonomous spacecraft guidance and control research in Mars pinpoint landing and precision formation flying (e.g., Açıkmeşe *et al.* [3, 4]) have also used SOCP formulations for trajectory generation.

Another common approximation method is to use polynomial basis functions to pa-

rameterize the optimization function, system dynamics, and constraints. For instance, Milam [34, 35] utilizes computationally-efficient B-splines that have local support (hence fewer numbers of basis functions) to parameterize the optimization problem and solve for the feedforward trajectories and control inputs through a nonlinear programming problem. Chebyshev polynomials [3, 15, 51] are another common choice of polynomial basis function that provides approximations with a reduced number of basis functions. The use of polynomials for approximation does not preclude the possibility of transforming the standard-mode optimization into a convex optimization problem; clever formulations for the dynamics and non-convex control constraints in Açıkmeşe and Ploen [3] transformed the optimization into a convex optimization problem that made use of Chebyshev polynomials for providing continuous thrust inputs.

The discrete SR-MPC algorithm will only guarantee satisfaction of state constraints at discrete times, even though the control policies are intended for application to continuous-time dynamics. This issue is not resolved within this thesis, however, a common approach in some applications (e.g., Schouwenaars *et al.* [43]) is to enforce more-conservative state constraints on the discrete-time system to help minimize the risk to continuous-time system of violating the state constraints. Other applications (e.g., Açıkmeşe *et al.* [4]) have dynamics that can be throughly analyzed such that control constraints can be set to ensure continuous-time state-constraint satisfaction.

The organization of this chapter starts with an overview of the class of systems and control objectives used for developing the discrete SR-MPC algorithm. The feedback policy is formulated first, as it is applied to both standard and safety operational modes. The safety-mode feedforward policy is developed next, followed by the standard-mode feedforward policy, which incorporates constraints from both the feedback development and the safety-mode policy (in a manner similar to the continuous-time framework). The standard-mode optimization in this discrete implementation puts a cost function only on the feedforward control sequence, which is solved over a finite horizon that connects the current state to the target state (which might not be the origin). Since the cost function is only on the control input, convergence properties for the discrete SR-MPC algorithm are only proven in a compressing-horizon implementation. For a receding-horizon implementation, a cost on the state would also be needed. As such, the discrete SR-MPC algorithm is not as general as the continuous-time counterpart in Chapter 4.

5.1 System and Algorithm Overview

The class of systems for which the discrete SR-MPC algorithm is developed have continuous-time actual dynamics of the form

$$\dot{x} = Ax + Bu + E\phi\left(C_q x\right) + Ed, \tag{5.1}$$

where ϕ is a continuously-differentiable, uncertain, nonlinear function of state x, d is a bounded exogenous disturbance, and control input u is split into separate feedforward (u_o) and feedback (u_f) components: $u(t) = u_o(t) + u_f(t)$. The continuous-time *nominal* model of the system, used in the development of the discrete SR-MPC algorithm, is as follows:

$$\dot{z} = Az + Bu_o + E\psi\left(C_q z\right), \tag{5.2}$$

where z is the nominal state, u_o is the nominal (feedforward) control input, and ψ is a continuously-differentiable function approximating the actual nonlinearity ϕ. Note, the nonlinearities ϕ and ψ could also include dependence on the inputs u and u_o, respectively, as was done in the continuous-time formulations; this is left out to simplify the discrete formulation but can be included with minor modifications.

The control policies will be developed based on a discretization of the actual and nominal systems in (5.1) and (5.2), respectively. The dynamics discretization is based on a fixed time-step interval Δt. Depending on the operation mode (standard or safety), slightly different conditions will govern the discrete dynamics formulation; these models will be discussed in subsequent sections of this chapter. The discrete SR-MPC algorithm will only provide guarantees for satisfying the state constraints in the control objectives at discrete times, as mentioned in the introduction to this chapter. Since the control policy is discretized with zero-order hold, satisfaction of the control constraints at discrete times also implies satisfaction in between (i.e., the control input will satisfy the control constraints in continuous time as well).

The objectives for the discrete SR-MPC algorithm are similar to those of the continuous-time implementation. Obtain a control input $u(t) \in \mathbf{U}, \forall t$, for actual system (5.1) that achieves the following closed-loop responses in each mode:

I. *Standard* mode: the actual state $x(t)$ converges into terminal set $\mathbf{\Omega}$ about the fixed

target state x_F, with a region of attraction $\mathcal{R}_d \subseteq \mathbf{X}$, such that when $x(t_0) \in \mathcal{R}_d$,

$$x(t_k) \in \mathbf{X},\ \forall t_k \geq t_0, \quad \text{and} \quad x(t_k) \in \{x_F\} + \mathbf{\Omega},\ \forall t_k \geq T_f, \tag{5.3}$$

where T_f is finite, discrete time $t_k = k \cdot \Delta t + t_0, k \in \mathbb{Z}_0^+$, and $\mathbf{\Omega}$ is invariant for $x(t)$ at discrete times $t_k \geq T_f$.

II. *Safety* mode: the actual state $x(t)$ at discrete times $t = \tilde{t}_l = l \cdot \Delta t + t_s, \forall l \in \mathbb{Z}_0^+$, is contained within invariant set $\mathbf{X}_s$ about a fixed reference r_s such that

$$x(\tilde{t}_l) \in \{r_s\} + \mathbf{X}_s \quad \text{and} \quad x(\tilde{t}_l) \in \mathbf{X},\ \forall \tilde{t}_l \geq t_s \geq t_0, \tag{5.4}$$

where $l = 0$ corresponds to safety switch-on time $t = t_s$.

The discrete SR-MPC algorithm is developed for systems where both the target state x_F (which might not be the origin) and the safety reference r_s satisfy further conditions that will be specified in Sections 5.2.2 and 5.2.3, respectively. Set $\mathcal{R}_d$ will be defined based on the architecture of the discrete SR-MPC algorithm.

The constraint sets in discrete SR-MPC are specified in the same manner as the continuous-time algorithm:

$$\mathbf{U}_o + \mathbf{U}_f \subseteq \mathbf{U}, \tag{5.5}$$

$$\mathbf{Z}_s + \mathbf{X}_f \subseteq \mathbf{X}_s, \tag{5.6}$$

$$\mathbf{Z}_n + \mathbf{X}_s \subseteq \mathbf{X}, \tag{5.7}$$

where each subset contains the origin. The feedback and safety sets $\mathbf{X}_f$ and $\mathbf{Z}_s$, respectively, are both convex sets prescribed as design specifications for development of the feedback and safety control policies. For the discrete algorithm, the nominal state constraints $\mathbf{Z}_n$ must also be convex. Additionally, the terminal set $\mathbf{\Omega}$ is specified as follows in the discrete implementation:

$$\mathbf{\Omega}_n + \mathbf{X}_f \subseteq \mathbf{\Omega} \tag{5.8}$$

where set $\mathbf{\Omega}_n \subset \mathbf{Z}_n$ defines the nominal terminal set and is utilized in the standard-mode computations for feedforward u_o. Sets $\mathbf{\Omega}_n$ and $\mathbf{\Omega}$ are both compact, convex, and contain the origin in their interiors.

Remark 5.1. For implementations of the discrete SR-MPC algorithm with terminal state x_F not at the origin, the sets $\mathbf{Z}_n$ and $\mathbf{X}$ do not actually need to contain the origin. Further, the origin need not be an equilibrium point for the actual or nominal dynamics in these cases. Again, x_F will be subject to additional conditions to be specified in Section 5.2.3. $\diamond$

The control components are designed to obey the separate control constraints $u_o \in \mathbf{U}_o$ and $u_f \in \mathbf{U}_f$. For the discrete implementation, these constraints are specified by a Euclidean norm:

$$\mathbf{U}_o \triangleq \{u_o : \|u_o\| \leq U_{max}\}, \tag{5.9}$$

$$\mathbf{U}_f \triangleq \{u_f : \|u_f\| \leq V_{max}\}. \tag{5.10}$$

General geometric descriptions for the feedback set $\mathbf{X}_f$, the safety set $\mathbf{Z}_s$, and the nominal terminal set $\mathbf{\Omega}_n$, expressed as the intersection of half planes and ellipsoids, are as follows. Note, the matrices C_s and C_f in Conditions 5.2 and 5.3, respectively, allow for specification of the constraints with respect to a subset of the state:

Condition 5.1. Feedback State Constraints $\mathbf{X}_f$:

$$\mathbf{X}_f \subseteq (\Gamma_1 \cap ... \cap \Gamma_{n_1}) \cap (\Lambda_1 \cap ... \cap \Lambda_{n_2}) \tag{5.11}$$

where $$\Gamma_i = \{\eta : a_i^T \eta \leq 1\}, \qquad i = 1, ..., n_1 \tag{5.12}$$

$$\Lambda_j = \{\eta : \eta^T Y_j \eta \leq 1;\ Y_j = Y_j^T > 0\}, \qquad j = 1, ..., n_2. \tag{5.13}$$

$\diamond$

Condition 5.2. Feedforward State Constraints $\mathbf{Z}_s$:

$$\mathbf{Z}_s \subseteq (\Sigma_1 \cap ... \cap \Sigma_{m_1}) \cap (\Upsilon_1 \cap ... \cap \Upsilon_{m_2}) \tag{5.14}$$

where $$\Sigma_i = \{\tilde{z} : b_i^T C_s \tilde{z} \leq 1\}, \qquad i = 1, ..., m_1 \tag{5.15}$$

$$\Upsilon_j = \{\tilde{z} : \tilde{z}^T C_s^T \Pi_j C_s \tilde{z} \leq 1;\ \Pi_j = \Pi_j^T > 0\}, \qquad j = 1, ..., m_2. \tag{5.16}$$

$\diamond$

Condition 5.3. Nominal Terminal State Constraints $\mathbf{\Omega}_n$:

$$\mathbf{\Omega}_n \subseteq (\Psi_1 \cap ... \cap \Psi_{h_1}) \cap (\Xi_1 \cap ... \cap \Xi_{h_2}) \tag{5.17}$$

where $$\Psi_i = \{\hat{z} : c_i^T C_f \hat{z} \leq 1\}, \qquad i = 1, ..., h_1 \tag{5.18}$$

$$\Xi_j = \{\hat{z} : \hat{z}^T C_f^T W_j C_f \hat{z} \leq 1;\ W_j = W_j^T > 0\}, \qquad j = 1, ..., h_2. \tag{5.19}$$

$\diamond$

5.2 Algorithm Construction

5.2.1 Feedback Policy

The feedback policy u_f is designed to ensure that the difference between the actual and nominal states remains bounded inside the state constraint set $\mathbf{X}_f$, which is a given design specification.

The design of u_f utilizes the *error* dynamics between systems (5.1) and (5.2):

$$\dot{\eta} = A\eta + Bu_f + E(\pi + w) + Ed, \tag{5.20}$$

where $\eta = x - z$, $\pi = [\phi(C_q x) - \phi(C_q z)]$, and $w = [\phi(C_q z) - \psi(C_q z)]$. Variable w represents the mismatch between actual nonlinearity ϕ and the nonlinearity model ψ.

Assuming π, w, and d are constant over a time step, and using zero-order hold on input u_f, the following discrete dynamics model approximates system (5.20):

$$\eta_{k+1} = A_d \eta_k + B_d u_{f,k} + E_d(\pi_k + w_k) + E_d d_k. \tag{5.21}$$

Note, disturbance d_k will be dropped from the dynamics in (5.21), based on the assumption that it is captured in the mismatch w_k, which will be treated like a bounded disturbance in a forthcoming condition. The revised discrete dynamics are

$$\eta_{k+1} = A_d \eta_k + B_d u_{f,k} + E_d(\pi_k + w_k). \tag{5.22}$$

Conservative characterization of π_k and w_k provides feedback designs valid for the continuous-time systems. The below conditions are instrumental in characterizing these terms to prescribe a degree of robustness to the control algorithm.

Condition 5.4. Function ϕ is continuously differentiable and there exists a closed and convex set of matrices $\mathcal{G}$ such that

$$\frac{\partial \phi}{\partial q}(q) \in \mathcal{G}, \quad \forall q, \tag{5.23}$$

where

$$\mathcal{G} = \{\Theta : \ \Theta^T \Theta \leq \gamma^2 I\}, \tag{5.24}$$

and $\gamma > 0$. ◇

Condition 5.5. There is a known bound on uncertainty $w(q) = \phi(q) - \psi(q)$, i.e., there exists $\sigma > 0$ such that

$$\|w(q)\| \leq \sigma, \quad \forall q. \tag{5.25}$$

◇

Note, σ can be selected to conservatively bound the exogenous disturbance d as well: if $\|w\| \leq \sigma_1$ and $\|d\| \leq \sigma_2$, then $\|w + d\| \leq \sigma_1 + \sigma_2 \leq \sigma$. If this combined bound provides too much conservatism in feedback policy design, then the forthcoming framework in Theorem 5.1 can be modified to treat the bounds separately.

The discrete feedback $u_{f,k}$ is designed to ensure that $\eta_k \in \mathbf{X}_f, \forall k$.* The form of the feedback policy for a time step $t \in [t_k, t_{k+1})$ is

$$u_{f,k} = K_f \eta_k, \tag{5.26}$$

with K_f being the constant feedback gain matrix generated offline by satisfying the matrix inequalities in the following theorem. Note, the theorem also establishes a design for $u_{f,k}$ that also ensures that when $\eta_k \in \mathbf{X}_f$, the feedback control constraint $\mathbf{U}_f$ in (5.10) is not violated.

Theorem 5.1. *Consider the discrete time system (5.22) satisfying Conditions 5.4 and 5.5. Suppose there exist matrices $P = P^T > 0$ and L and scalars $\alpha > 0$ and $\lambda \in (0, 1)$ such that*

*There is no guarantee of state bounds between each step k. More in-depth analysis is required in the Δt selection to ensure bounds between discrete steps. See reference [4] for a specific consideration of this issue.

the following set of matrix inequalities are satisfied:

$$\begin{bmatrix} -(1-\lambda)P & 0 & 0 & PA_d^T + L^T B_d^T & PC_q^T \\ 0 & -\alpha I & 0 & \alpha\gamma E_d^T & 0 \\ 0 & 0 & -\lambda I & \sigma E_d & 0 \\ A_d P + B_d L & \alpha\gamma E_d & \sigma E_d & -P & 0 \\ C_q P & 0 & 0 & 0 & -\alpha I \end{bmatrix} \leq 0, \tag{5.27}$$

$$a_i^T P a_i \leq 1, \quad i = 1, \ldots, n_1, \tag{5.28}$$

$$\begin{bmatrix} P & P \\ P & Y_j^{-1} \end{bmatrix} \geq 0, \quad j = 1, \ldots, n_2, \tag{5.29}$$

$$\begin{bmatrix} P & L^T \\ L & V_{max}^2 I \end{bmatrix} \geq 0, \tag{5.30}$$

where V_{max} is a bound on the norm of the feedback control. Then, the feedback policy given by

$$u_{f,k} = K_f \eta_k, \text{ where } K_f = LP^{-1}, \tag{5.31}$$

renders $\mathbf{X}_f = \{\eta : \eta^T P^{-1} \eta \leq 1\}$ an invariant set for (5.22), i.e., $\eta_k \in \mathbf{X}_f, k \in \mathbb{Z}_0^+$ for any solution of system (5.22) with $\eta_0 \in \mathbf{X}_f$. Additionally, $\mathbf{X}_f$ satisfies (5.11), and $u_{f,k} \in \mathbf{U}_f, \forall \tilde{\eta}_k \in \mathbf{X}_f$, with $\mathbf{U}_f$ from (5.10). ◇

Proof. See Appendix B.1 for a proof of Theorem 5.1. □

Remark 5.2. Most of the matrix inequalities in Theorem 5.1 are an LMI (Linear Matrix Inequality), except for (5.27) which is a BMI (Bilinear Matrix Inequality). Note, for a fixed λ, (5.27) is an LMI, and $\lambda \in (0, 1)$ for a feasible solution. This is similar to Remark 3.6 for the continuous-time Corollary 3.1.1. ◇

Remark 5.3. An LMI-based design procedure can easily be constructed by using Theorem 5.1 in a line search on λ to maximize the volume of $\mathbf{X}_f$. For a given λ, the volume of $\mathbf{X}_f$

can be maximized by using the following semi-definite program [8]:

$$\begin{aligned} &\text{Minimize } \log(\det P^{-1}) \\ &\text{subject to } P = P^T > 0, \alpha > 0, \text{ and equations (5.27)–(5.30).} \end{aligned} \tag{5.32}$$

◇

5.2.2 Safety-Mode Feedforward Policy

The feedforward policy u_{os} for safety mode together with the feedback policy u_f establishes safety mode for the actual system (Control Objective II); subscript s is appended to u_o here to clarify safety-mode feedforward from standard-mode feedforward. The feedforward u_{os} is designed to ensure that the nominal system states z remain in the proximity of safety reference r_s (i.e., $z - r_s \in \mathbf{Z}_s$), and the feedback u_f maintains the error state $\eta = x - z \in \mathbf{X}_f$. Thus, safety-mode control $u = u_{os} + u_f$ ensures actual state $x - r_s \in \mathbf{X}_f + \mathbf{Z}_s \subseteq \mathbf{X}_s$ as in constraint (5.6).

Safety reference r_s for the discrete algorithm is defined through a mapping T_s of the standard-mode nominal position at safety switch-on time so that:

$$r_s = T_s z_{stan}(t_s), \tag{5.33}$$

where $z_{stan}(t_s) \in \mathbf{Z}_n$ is the nominal state z from standard mode at the safety-activation time t_s. Mapping $T_s : \mathbf{Z}_n \to \mathbf{Z}_n : z_{stan}(t_s) \mapsto r_s$ such that $T_s\mathbf{Z}_n \subseteq \mathbf{Z}_n$ and $r_s \in T_s\mathbf{Z}_n$, which ensures that $T_s\mathbf{Z}_n + \mathbf{X}_s \subseteq \mathbf{X}$ so that Control Objective II is satisfied with the safety-mode control policy.

The nominal dynamics in (5.2) for safety mode are given by

$$\dot{\tilde{z}} = A\tilde{z} + Bu_{os} + Ar_s + E\psi(C_q\tilde{z} + C_q r_s), \tag{5.34}$$

where $\tilde{z} \triangleq z - r_s$.

The following condition is useful in formulating the discrete safety-mode dynamics:

Condition 5.6. There is a g_c such that safety reference r_s satisfies

$$Ar_s = Eg_c. \tag{5.35}$$

◇

Note, Condition 5.6 is a weaker (i.e., less restrictive) statement of Condition 4.5 for the related continuous-time formulation of SR-MPC; for $g_c = 0$, the two conditions are identical. The application in Chapter 6 will provide one example for which this weaker condition is advantageous.

Utilizing Condition 5.6, system (5.34) can be rewritten as

$$\dot{\tilde{z}} = A\tilde{z} + Bu_{os} + E\left(g_c + \psi\left(C_q\tilde{z} + C_q r_s\right)\right), \tag{5.36}$$

$$= A\tilde{z} + Bu_{os} + E\bar{g}(\tilde{z}, r_s). \tag{5.37}$$

The following additional condition (similar to Condition 4.6 for continuous-time SR-MPC) is useful in development of the feedforward policy:

Condition 5.7. There is a known bound on $\bar{g}$; i.e., there exists $\delta > 0$ such that

$$||\bar{g}(\tilde{z}, r_s)|| \leq \delta, \quad \forall z = \tilde{z} + r_s. \tag{5.38}$$

◇

The feedback policy in Section 5.2.1 used a bound on the difference between ϕ and ψ (Condition 5.5), so the addition of Condition 5.7 implies a bound on ϕ as well.

Assuming $\bar{g}$ is constant over a time step, and using zero-order hold on input u_{os}, the following discrete dynamics model approximates system (5.37):

$$\tilde{z}_{l+1} = A_d\tilde{z}_l + B_d u_{os,l} + E_d\bar{g}_l, \tag{5.39}$$

where $l = 0$ corresponds to $t = t_s$ (safety switch-on time), with discrete time in safety mode defined as $\tilde{t}_l = l \cdot \Delta t + t_s, \forall l \in \mathbb{Z}_0^+$. Note, Condition 5.7 also implies that $||\bar{g}_l|| \leq \delta, \forall l$.

The discrete feedforward $u_{os,l}$ is designed to ensure that $\tilde{z}_l \in \mathbf{Z}_s, \forall l$. The form of the safety feedforward policy for a time step $t \in [\tilde{t}_l, \tilde{t}_{l+1})$ is

$$u_{os,l} = K_s\tilde{z}_l, \tag{5.40}$$

with K_s being a constant gain matrix generated offline according to the following theorem. Note, the theorem also provides the design for $u_{os,l}$ to ensure that when $\tilde{z}_l \in \mathbf{Z}_s$, the

feedforward control constraint $\mathbf{U}_o$ in (5.9) is not violated.

Theorem 5.2. *Consider the discrete time system (5.39) with $\bar{g}_l$ satisfying Condition 5.7. Suppose there exist matrices $S = S^T > 0$ and R and scalar $\lambda \in (0, 1)$ such that the following set of matrix inequalities are satisfied:*

$$\begin{bmatrix} -(1-\lambda)S & 0 & SA_d^T + R^T B_d^T \\ 0 & -\lambda I & \delta E_d^T \\ A_d S + B_d R & \delta E_d & -S \end{bmatrix} \leq 0, \tag{5.41}$$

$$b_i^T C_s S C_s{}^T b_i \leq 1, \quad i = 1, \ldots, m_1, \tag{5.42}$$

$$\begin{bmatrix} S & SC_s^T \\ C_s S & \Pi_j^{-1} \end{bmatrix} \geq 0, \quad j = 1, \ldots, m_2, \tag{5.43}$$

$$\begin{bmatrix} S & R^T \\ R & U_{max}^2 I \end{bmatrix} \geq 0, \tag{5.44}$$

where U_{max} is a bound on the norm of the feedforward control. Then, the feedforward policy given by

$$u_{os,l} = K_s \tilde{z}_l \text{ where } K_s = RS^{-1} \tag{5.45}$$

renders $\mathbf{Z}_s = \{\tilde{z} : \tilde{z}^T S^{-1} \tilde{z} \leq 1\}$ an invariant set for (5.39), i.e., $\tilde{z}_l \in \mathbf{Z}_s, l \in \mathbb{Z}_0^+$ for any solution of system (5.39) with $\tilde{z}_0 \in \mathbf{Z}_s$. Additionally, $\mathbf{Z}_s$ satisfies (5.14), and $u_{os,l} \in \mathbf{U}_o, \forall \tilde{z}_l \in \mathbf{Z}_s$, with $\mathbf{U}_o$ from (5.9). ◇

Proof. See Appendix B.2 for a proof of Theorem 5.2. □

5.2.3 Standard-Mode Feedforward Policy

The standard-mode feedforward policy u_o together with the feedback policy u_f achieves Control Objective I for the actual system and also ensures safety-mode availability if it is needed. Like the continuous-time SR-MPC algorithm, the discrete SR-MPC algorithm ensures safety-mode availability at each time step by adding a safety constraint into the standard-mode algorithm. The feedforward u_o from solution of the FHC-D (Finite Horizon optimal Control problem for Discrete SR-MPC) provides a guidance policy to drive the

nominal system through a series of waypoints (both position and velocity) generated between the initial state and the desired target state. The FHC-D is essentially a conversion of the continuous-time FHC into a SOCP, which requires the nominal state constraints $\mathbf{Z}_n$ to be convex in applications of the discrete SR-MPC algorithm. The waypoints satisfy the nominal state constraints (i.e., $z \in \mathbf{Z}_n$), and the added feedback u_f keeps the actual state in proximity of the nominal state (i.e., $x - z \in \mathbf{X}_f$) so that overall state constraints are not violated ($x \in \mathbf{X}$); refer back to (5.7).

A linearization of the nominal-system nonlinearity ψ is used in formulating the FHC-D. The resulting nominal dynamics from (5.2) used in the discrete standard mode are given by

$$\dot{z} = F_k z + B u_o + E\psi_k, \quad \text{with} \quad F_k = A + E G_{q_k} C_q, \tag{5.46}$$

where F_k incorporates the linear terms for $\psi(\cdot)$ linearized about a particular reference $q_k = C_q z(t_k)$. The linearization is as follows:

$$\psi(q) \approx G_{q_k} q + \psi_k, \tag{5.47}$$

where $q(t) = C_q z(t)$, $q_k = C_q z(t_k)$, $G_{q_k} = \left.\frac{\partial \psi}{\partial q}\right|_{q_k}$, and $\psi_k = \psi(q_k) - G_{q_k} q_k$. In a discretization of the dynamics in (5.46), time-varying F_k and ψ_k can be incorporated by updating them at each discrete state $z(t_k)$ along the trajectory.

Assuming q_k is constant (thus ψ_k and G_{q_k} are constant) over a time step, and using zero-order hold on input u_o, the following discrete dynamics model approximates (5.46):

$$z_{k+1} = A_k z_k + B_k u_{o,k} + E_k \psi_k. \tag{5.48}$$

The input $u_{o,k}$ comes from solution of the FHC-D, which also enforces the feedforward control constraint $\mathbf{U}_o$ from (5.9).

For the terminal target set $\mathbf{\Omega}_n$, a slightly different discretization is used for generating a nominal local controller about the target state x_F. This is analogous to the local controller from the continuous-time algorithms that is switched on once the state enters the terminal set. As with the safety-mode controller, the local controller about x_F is developed for

nominal dynamics shifted so that x_F is at the origin:

$$\dot{\hat{z}} = A\hat{z} + Bu_o + Ax_F + E\psi(C_q\hat{z} + C_q x_F), \tag{5.49}$$

where $\hat{z} \triangleq z - x_F$. This form is identical to the nominal dynamics used in the development of the safety mode control policy. Requiring terminal state x_F to satisfy Conditions 5.6 and 5.7 (with r_s replaced by x_F) provides a discrete model identical to (5.39) from the safety-mode formulation. The terminal local controller can then be generated such that

$$u_{o,k} = L_n\hat{z}_k, \tag{5.50}$$

where L_n is a constant gain matrix generated offline by using the matrix inequalities in Theorem 5.2 for the generation of the invariant terminal set $\mathbf{\Omega}_n = \{\hat{z} : \hat{z}^T Q^{-1}\hat{z} \leq 1\}$, which also satisfies 5.17. Note, Theorem 5.2 for the terminal controller swaps the safety-mode design specification in Condition 5.2 for those of the nominal terminal set in Condition 5.3; additionally, matrices S and R are replaced by equivalents (e.g., Q and W, respectively) for clarifying that the theorem is being applied to generate the terminal invariant set and local controller. Since the theorem is redundant, it is not repeated. Note, the theorem also guarantees that the local control policy (5.50) satisfies the control constraints in (5.9) for all $\hat{z}_k \in \mathbf{\Omega}_n$.

The FHC-D for the standard-mode feedforward optimization can now be defined. Since matrices A_k, B_k, and E_k in (5.48) and ψ_k are time varying, based on the guidance path z_k, an iteration process is needed to find an initial feasible solution. In the FHC-D, a superscript j will be appended to z_k^j, A_k^j, B_k^j, E_k^j, and ψ_k^j to denote this iteration, with $j = 0, \ldots, j_{max}$ set by the designer. Given an initial state x_0, a desired target state x_F, and a desired planning horizon T_M (such that horizon length $N = \frac{T_M}{\Delta t} \in \mathbb{Z}^+$), an initial polynomial guess (iteration $j = 0$) is made for the nominal guidance path $\{z_k^0\}$ for $k = 1, \ldots, N$ such that $z_0^0 = x_0$ and $z_N^0 = x_F$. From this initial guess, the FHC-D can be applied to produce an admissible set of inputs $u_{o,k}$ and states z_k that satisfy (5.48).

FHC-D (for discrete SR-MPC)

Given current state $x_0 = x(t_i)$, desired final state x_F, and initial guess $\{z_0^0, \ldots, z_{N-1}^0, x_F\}$, iterate the following for $j = 0, \ldots, j_{max} - 1$:

1. Compute A_k^j, B_k^j, E_k^j, and ψ_k for $k = 0, ..., N-1$, by using $\{z_0^j, \ldots, z_{N-1}^j\}$.

2. Solve the following SOCP problem for $\{z_0^{j+1}, \ldots, z_N^{j+1}\}$ and $\{u_{o,0}^{j+1}, \ldots, u_{o,N-1}^{j+1}\}$:

$$\text{Minimize} \quad \sum_{k=0}^{N-1} \left(\alpha \|u_{o,k}^{j+1}\| + \beta \|u_{o,k}^{j+1}\|^2 \right)$$

subject to

$$\begin{aligned}
& z_{k+1}^{j+1} = A_k^j z_k^{j+1} + B_k^j u_{o,k}^{j+1} + E_k^j \psi_k^j, && k = 0, \ldots, N-1 && \\
& u_{o,k}^{j+1} \in \mathbf{U}_o, && k = 0, \ldots, N-1 && \\
& z_k^{j+1} \in \mathbf{Z}_n, && k = 1, \ldots, N-1 && \\
& (z_0^{j+1} - x_0)^T P^{-1} (z_0^{j+1} - x_0) \leq 1 && && (FHC-1) \\
& (z_k^{j+1})^T D_s^T S^{-1} D_s z_k^{j+1} \leq 1, && k = 0, \ldots, N-1 && (FHC-2) \\
& (z_N^{j+1} - x_F)^T Q^{-1} (z_N^{j+1} - x_F) \leq 1 && && (FHC-3)
\end{aligned}$$

where $\epsilon \geq 0$, $D_s = I - T_s$, with T_s from (5.33), and $(\alpha, \beta) = (1, 0)$ or $(0, 1)$.

Variables α and β in the FHC-D provide cost options: for example, in mechanical vehicle applications, $(\alpha, \beta) = (1, 0)$ would minimize fuel, and $(\alpha, \beta) = (0, 1)$ would minimize input energy. Feasibility of the FHC-D is used to define set $\mathcal{R}_d$ from Control Objective I:

Definition 5.1 (Feasibility Region $\mathcal{R}_d$)**.**

$$\mathcal{R}_d = \{\xi \in \mathbf{Z}_n + \mathbf{X}_f : \text{ the FHC-D is feasible with } x_F \text{ and } x(t_i) = \xi\}. \tag{5.51}$$

$\diamond$

The three inequality constraints (FHC-1), (FHC-2), and (FHC-3) in step 2 of the FHC-D are contributions from the feedback policy design, the feedforward safety-mode policy design, and the nominal terminal-set local-control-policy design, respectively. The feedback constraint in (FHC-1) comes directly from the definition of constraint set $\mathbf{X}_f = \{\eta : \eta^T P^{-1} \eta \leq 1\}$ from Theorem 5.1. Inequality (FHC-1) on the initial state ensures that the actual state remains in the proximity of the nominal guidance states when feedback $u_{f,k}$ is simultaneously applied with the feedforward policy $u_{o,k}, \forall k$. Further, the constraint (FHC-1) is useful in guaranteeing re-solvability for the FHC-D (discussed in the

next section). Similarly, terminal-set constraint (FHC-3) comes directly from constraint set $\mathbf{\Omega}_n = \{\hat{z} : \hat{z}^T Q^{-1} \hat{z} \leq 1\}$ by using Theorem 5.2 for the local control policy design. Inequality (FHC-3) on the terminal nominal z_N ensures that the trajectory ends in control-invariant terminal set $\mathbf{\Omega}_n$.

For safety-mode availability at any time step k, the nominal states are constrained in the FHC-D so that $z_k - T_s z_k \in \mathbf{Z}_s, \forall k$. If safety mode is activated at step k, then $t_s = t_k$, safety reference r_s is fixed to $r_s = T_s z_k$ as in (5.33), and thus, $\tilde{z}_0 = z_k - r_s \in \mathbf{Z}_s$. The safety-mode feedforward u_{os} designed with Theorem 5.2 is then assured to provide safety.

Incorporation of constraint $z_k - T_s z_k \in \mathbf{Z}_s$ in the FHC-D is accomplished by using matrix S generated from the inequalities of Theorem 5.2. Matrix S defines the safety state set $\mathbf{Z}_s = \{\tilde{z} : \tilde{z}^T S^{-1} \tilde{z} \leq 1\}$. Utilizing this definition, along with $\tilde{z} = z_k - r_s = z_k - T_s z_k \in \mathbf{Z}_s, \forall k$ as discussed above, provides the safety constraint (FHC-2) in the FHC-D:

$$\tilde{z}^T S^{-1} \tilde{z} \leq 1 \quad \Longrightarrow \quad (z_k - T_s z_k)^T S^{-1} (z_k - T_s z_k)^T \leq 1 \quad \Longrightarrow \quad z_k^T D_s^T S^{-1} D_s z_k \leq 1$$

where $D_s = I - T_s$.

Remark 5.4. In general there is no guarantee that the initial set of FHC-D iterations will converge to a solution trajectory, even for a feasible problem. This has not been an observed issue in applications such as Chapter 6 where a small, finite number of iterations generates a solution to the FHC-D. $\diamond$

5.3 Discrete SR-MPC Algorithm

The discrete SR-MPC algorithm propagates forward in discrete time t_k with a fixed time interval $t_{k+1} - t_k = \Delta t, \forall k \in \mathbb{Z}_0^+$. The following definition for re-solve times and the horizon lengths used in the FHC-D allows for re-solves at intervals that are variable multiples of the discrete time step Δt.

Definition 5.2 (Re-solve Times and Horizon Lengths)**.** Define the following:

- $r = 0, 1, \ldots, r_{max}$ as the index number of re-solves.
- $\{m_r\} \in \mathbb{Z}_0^+$ as the monotonically increasing sequence of re-solve steps with $m_0 = 0$ and $\delta_r = m_r - m_{r-1}, 1 \leq \delta_r < N_{r-1}, \forall r \geq 1$.

- $\{N_r\} \in \mathbb{Z}^+$ as the sequence of *horizon lengths* with $N_r \geq N_{r-1} - \delta_r$ when $r \geq 1$.

Re-solve times coincide with $t_k \in \{t_{m_r}\}$; i.e., when a step k in the discrete SR-MPC algorithm coincides with an element of $\{m_r\}$. $\diamond$

Since sequence $\{m_r\}$ is monotonically increasing there are no repeated re-solve times, and the initial computation coincides with $k = m_0 = 0$ for $r = 0$. The constraints on the horizon N_r require that each re-solve ($\forall r \geq 1$) occurs before the end of the prior horizon, and the constraints also imply that $N_r \geq 1, \forall r$. Note, for a compressing-horizon implementation, only a finite number of re-solves of the FHC-D would be implemented, resulting in a finite $r_{max} \leq N_0 - 1$.

The following algorithm describes the Discrete SR-MPC approach.

Discrete SR-MPC Algorithm

Begin at $k = m_0 = 0$ in standard mode with $x(t_0) \in \mathcal{R}_d$ and iterate the following steps over re-solve times $\{t_{m_r}\}, \forall r$:

standard mode

1. Measure state $x(t_{m_r})$ of actual system (5.1).
2. Solve the FHC-D at time $t_i = t_{m_r}$ with $N = N_r$ to obtain $\{\mathbf{z}_k\} = \{z_k^{j_{max}}\}$ for $k = m_r, \ldots, m_r + N_r$ and $\{\mathbf{u}_{o,k}\} = \{u_{o,k}^{j_{max}}\}$ for $k = m_r, \ldots, m_r + N_r - 1$.
3. Monitor $x(t)$ while applying $u(t) = u_o(t) + u_f(t)$ to actual system (5.1) where $u_o(t) = \mathbf{u}_{o,k}$ and $u_f(t) = u_{f,k} = K_f(x(t_k) - \mathbf{z}_k)$ on $t \in [t_k, t_{k+1})$.
4. Check the following at each t_k:
   (a) If safety event detected at $t_s = t_k$, set $r_s = T_s \mathbf{z}_k$ and $\tilde{z}_0 = \mathbf{z}_k - r_s$, with $\tilde{t}_0 = t_s$, then switch to safety mode and stop iteration.
   (b) If $\hat{z}_k = z_k - x_F \in \mathbf{\Omega}_n$, then set $u_o(t) = u_{o,k} = L_n \hat{z}_k, \forall t \geq t_k$ and skip step 2 in iteration.

safety mode

For $t \geq t_s$, apply $u(t) = u_o(t) + u_f(t)$ to actual system (5.1) where $u_o(t) = u_{os,l} = K_s \tilde{z}_l$ and $u_f(t) = u_{f,l} = K_f(x(\tilde{t}_l) - z_l)$ on $t \in [\tilde{t}_l, \tilde{t}_{l+1})$, $l \in \mathbb{Z}_0^+$, with $z_l = \tilde{z}_l + r_s$ and $\tilde{z}_l$ from discrete nominal system (5.39).

The ability to re-solve the FHC-D in standard mode allows for model updating and improvements to the feedforward u_o and guidance trajectory based on current actual states and parameters. Re-solvability implies that once initial feasibility is established for the FHC-D, than future re-computations remain feasible.

Lemma 5.1 (Re-solvability of FHC-D)**.** *Suppose the FHC-D is feasible at t_0 ($k = 0$) with horizon N_0. Then, feasibility of the FHC-D is guaranteed at re-solve times $\{t_{m_r}\}$ with horizon lengths $\{N_r\}, \forall r$, from Definition 5.2 provided that feedback $u_{f,k} = K_f(x_k - z_k)$ is applied, $\forall k \in \mathbb{Z}_0^+$, with K_f from Theorem 5.1 and $x_k = x(t_k)$.* $\diamond$

Proof. (by induction) Suppose at $t_{m_{r-1}}$ (i.e., $k = m_{r-1}$) the FHC-D is feasible with horizon N_{r-1}. The solution trajectory is $\{\mathbf{z}_k\}_{r-1} = \{z_k^{j_{max}}\}_{r-1}$ for $k = m_{r-1}, \ldots, m_{r-1} + N_{r-1}$ with element $k = m_{r-1} + N_{r-1}$ being $z_k^{j_{max}} - x_F \in \mathbf{\Omega}_n$. The solution feedforward input is $\{\mathbf{u}_{o,k}\}_{r-1} = \{u_{o,k}^{j_{max}}\}_{r-1}$ for $k = m_{r-1}, \ldots, m_{r-1} + N_{r-1} - 1$.

Propagate discrete time t_k forward δ_r time steps and re-solve the FHC-D at t_{m_r} (i.e., $k = m_r$, $m_r = m_{r-1} + \delta_r$) with horizon N_r. Consider the following candidate solution for the re-solve, which starts the new FHC-D iteration cycle with a feasible solution:

$$\begin{aligned} \{z_k^0\}_r &= \begin{cases} \{\mathbf{z}_k\}_{r-1}, & k = m_r, \ldots, m_{r-1} + N_{r-1}, \\ x_F, & k = m_{r-1} + N_{r-1} + 1, \ldots, m_r + N_r, \end{cases} \\ \{u_{o,k}^0\}_r &= \begin{cases} \{\mathbf{u}_{o,k}\}_{r-1}, & k = m_r, \ldots, m_{r-1} + N_{r-1} - 1, \\ L_n(\{z_k^0\}_r - x_F), & k = m_{r-1} + N_{r-1}, \ldots, m_r + N_r - 1. \end{cases} \end{aligned} \tag{5.52}$$

Since $N_r \geq N_{r-1} - \delta_r > 0$ (per Definition 5.2), then $m_r + N_r \geq m_{r-1} + N_{r-1}$. Additionally, since $\{\mathbf{z}_k\}_{r-1}$ at $k = m_{r-1} + N_{r-1}$ satisfies inequality (FHC-3) (i.e., $\{\mathbf{z}_k\}_{r-1} - x_F \in \mathbf{\Omega}_n$ for $k = m_{r-1} + N_{r-1}$), then $\{u_{o,k}^0\}_r = L_n(\{z_k^0\}_r - x_F), \forall k \geq m_{r-1} + N_{r-1}$ implies $\{z_k^0\}_r$ satisfies (FHC-3), $\forall k \geq m_{r-1} + N_{r-1}$.

Feedforward $\{\mathbf{u}_{o,k}\}_{r-1}$ and $L_n \hat{z}_k$ (with $\hat{z}_k = \{z_k^0\}_r - x_F, \forall k \geq m_{r-1} + N_{r-1}$) satisfy all control constraints, and thus $\{u_{o,k}^0\}_r$ does too. Waypoints $\{\mathbf{z}_k\}_{r-1}$ satisfy all nominal state constraints, including constraint (FHC-2), and thus $\{z_k^0\}_r$ does as well. Further, since feedback $u_{f,k}$ maintains $x_k - z_k \in \mathbf{X}_f$ for $k = m_{r-1}, \ldots, m_r$, then $x_{m_r} - z_{m_r} \in \mathbf{X}_f$ is a feasible initial condition satisfying constraint (FHC-1) at re-solve time t_{m_r}.

Thus, candidate solution (5.52) is a feasible solution for the FHC-D re-solve at time

t_{m_r} with horizon N_r once a prior feasible solution is established at time $t_{m_{r-1}}$ with horizon N_{r-1}. The proof concludes by induction. □

Theorem 5.3 (Compressing-Horizon Satisfaction of Discrete SR-MPC Control Objectives)**.** *Consider actual system (5.1) for x satisfying Conditions 5.4, 5.5, 5.6, and 5.7 with a control input u described by a compressing-horizon implementation of the Discrete SR-MPC algorithm. If Theorems 5.1 and 5.2 are also satisfied, then the resulting closed-loop system has discrete states and discrete control inputs that satisfy Control Objectives I and II for standard and safety modes, respectively.* ◇

Proof. The proof is split into two pieces

I. standard mode: Given the Discrete SR-MPC algorithm and $x(t_0) \in \mathcal{R}_d$, the FHC-D is initially feasible with some initial horizon $N_0 = \frac{T_M}{\Delta t} \in \mathbb{Z}^+$ and Lemma 5.1 is valid. Since the number of re-solves r_{max} from Definition 5.2 is finite in a compressing-horizon implementation ($r_{max} \leq N_0 - 1$), there exists a final feedforward nominal trajectory z_k and feedforward input $u_{o,k}$ generated from a re-solve of the FHC-D. This nominal trajectory will end in nominal terminal set $\mathbf{\Omega}_n$, so there exists some finite time $T_f \leq T_M$ such that $z_k - x_F \in \mathbf{\Omega}_n, \forall t_k \geq T_f$, corresponding to step 4b in the Discrete SR-MPC algorithm with nominal local controller (5.50).

   Feedback $u_{f,k} = K_f(x(t_k) - z_k) \in \mathbf{U}_f$ with K_f from Theorem 5.1 guarantees tracking of the feedforward trajectory such that $\eta_k = x(t_k) - z_k \in \mathbf{X}_f, \forall k$. Since the FHC-D remains feasible, the feedforward trajectory $z_k \in \mathbf{Z}_n, \forall k$, and the feedforward control input $u_{o,k} \in \mathbf{U}_o, \forall k$. Then, the actual trajectory will satisfy $x(t_k) = z_k + \eta_k \in \mathbf{Z}_n + \mathbf{X}_f \subseteq \mathbf{Z}_n + \mathbf{X}_s \subseteq \mathbf{X}, \forall t_k \geq t_0$ where the sets in (5.6) and (5.7) are used. Further, the actual trajectory will also satisfy $x(t_k) - x_F = (x(t_k) - z_k) + (z_k - x_F) \in \mathbf{X}_f + \mathbf{\Omega}_n \subseteq \mathbf{\Omega}, \forall t_k \geq T_f$, where the set addition in (5.8) is used.

   Since $u_{o,k} \in \mathbf{U}_o$ and $u_{f,k} \in \mathbf{U}_f, \forall k$, then input $u(t) = u_o(t) + u_f(t)$ defined in step 3 in the Discrete SR-MPC algorithm satisfies $u(t) \in \mathbf{U}_o + \mathbf{U}_f \subseteq \mathbf{U}, \forall t$, where the set addition in (5.5) is used. Thus, Control Objective I is satisfied.

II. safety mode: The feedback input $u_{f,k} \in \mathbf{U}_f$ applied during standard mode guarantees that $x(t_s) - z_k \in \mathbf{X}_f$ at k corresponding to $\tilde{t}_0 = t_s = t_k$ (safety activation time). Further, since the FHC-D is satisfied in standard mode, constraint (FHC-2) guarantees

that $z_k - r_s = z_k - T_s z_k \in \mathbf{Z}_s$ at $\tilde{t}_0$ with $r_s \in T_s\mathbf{Z}_n \subseteq \mathbf{Z}_n$ from (5.33). Thus, $x(t_s) - r_s = (x(t_s) - z_k) + (z_k - r_s) \in \mathbf{X}_f + \mathbf{Z}_s \subseteq \mathbf{X}_s$ at $\tilde{t}_0$, where the set definition in (5.6) for $\mathbf{X}_s$ is used.

Feedforward input $u_{os,l} = K_s\tilde{z}_l \in \mathbf{U}_o$ with K_s from Theorem 5.2 applied to nominal system (5.34) guarantees $\tilde{z}_l = z_l - r_s \in \mathbf{Z}_s, \forall l \in \mathbb{Z}_0^+$, where z_0 has been re-set to the standard-mode z_k at safety activation time $\tilde{t}_0 = t_s = t_k$. Thus, applying feedforward $u_{os,l} = K_s\tilde{z}_l \in \mathbf{U}_o$ and feedback $u_{f,l} = K_f(x(\tilde{t}_l) - z_l) \in \mathbf{U}_f$ to actual system (5.1) ensures $x(\tilde{t}_l) - r_s = (x(\tilde{t}_l) - z_l) + (z_l - r_s) \in \mathbf{X}_f + \mathbf{Z}_s \subseteq \mathbf{X}_s, \forall \tilde{t}_l$, where $\tilde{t}_l \geq t_s, \forall l \in \mathbb{Z}_0^+$. Further, since $r_s \in T_s\mathbf{Z}_n \subseteq \mathbf{Z}_n$, then $x(\tilde{t}_l) = r_s + (x(\tilde{t}_l) - r_s) \in \mathbf{Z}_n + \mathbf{X}_s \subseteq \mathbf{X}, \forall \tilde{t}_l$, where the set definition in (5.7) is used.

Since $u_{os,l} \in \mathbf{U}_o$ and $u_{f,l} \in \mathbf{U}_f, \forall l$, then input $u(t) = u_o(t) + u_f(t)$ defined in the safety mode stage of the Discrete SR-MPC algorithm satisfies $u(t) \in \mathbf{U}_o + \mathbf{U}_f \subseteq \mathbf{U}, \forall t \geq t_s$, where the set addition in (5.5) is used. Thus, Control Objective II is satisfied. □

5.4 Summary of the Discrete SR-MPC Algorithm

The discrete SR-MPC algorithm provides a practical means of applying the general continuous-time theory from Chapter 4 to applications that require a computer both to solve online the standard-mode finite-horizon optimization and to implement the standard- and safety-mode control policies. The discrete standard-mode finite-horizon optimization is formulated as a second-order cone program that can be solved online in a computationally efficient and accurate manner by using interior-point algorithms. Further, the discrete algorithm retains the innovations of the continuous-time SR-MPC algorithm: the feedback policy guarantees re-solvability of discrete SR-MPC and maintains the actual state in the proximity of the nominal discrete state at all discrete times; and, the safety-mode policy guarantees the actual state is within the desired safety set at all discrete times. The following chapter presents a detailed spacecraft example that implements the compressing-horizon version of the discrete SR-MPC algorithm.

Chapter 6

Application of SR-MPC to Spacecraft Proximity Operations

The objective of this chapter is to apply the theoretical SR-MPC framework developed in the proceeding chapters to a realistic engineering example requiring significantly more detail than that of the earlier, more-academic examples. Realistic effects such as control-input uncertainty, sensor noise, and unknown disturbances are included to further demonstrate the applicability of the discrete SR-MPC algorithm in a realistic implementation. The chosen application example is a robotic spacecraft conducting an autonomous descent toward the surface of a small celestial body (e.g., asteroid or comet). An assumption is made that onboard computational capability is sufficient to solve the standard-mode optimization and apply the resultant feedforward control policy with insignificant delay. This assumption is reasonable given that the optimization in discrete SR-MPC has a computationally-efficient formulation; refer back to Chapter 5 for the details.

Robotic spacecraft missions to small celestial bodies require a degree of onboard G&C (Guidance and Control) autonomy due to the long light-time delays for ground-based communication and a need for rapid, online decision making to mitigate risk from unexpected disturbances (e.g., comet outgassing) or unexpected obstacles. Proximity operations (e.g., descent, contact, ascent, hopping, etc.) in particular require G&C methods that are robust to model uncertainty (e.g., gravity model errors) and incorporate state and control constraints to ensure that the proximity operation can be conducted with available thrust and also minimize risk to the mission, such as keep the spacecraft from impacting the small body. These algorithms must further be computationally robust, providing some form of guaranteed G&C solution in real time. The discrete SR-MPC algorithm developed in Chapter 5

is well suited to address these G&C requirements.

The two operational modes in the SR-MPC algorithm provide an autonomous G&C capability for robotic spacecraft that addresses the mission requirements for small-body proximity operations. The standard mode maneuvers the spacecraft toward the target state in a robust and re-solvable model-predictive manner, and the safety mode provides a means of maintaining the spacecraft in a safe hover region if required. The safety mode would be triggered in the event that onboard sensors detect unexpected static state constraints (e.g., the surface is closer than expected) during the G&C planning and maneuver horizon.

Figure 6.1 provides a graphic illustration of the operational modes developed in Chapter 5. On the left is a sketch of a spacecraft in standard mode being maneuvered from an initial state x_0 to a desired target state x_F based on the expected small-body surface location. On the right is a sketch of the spacecraft being held in a safety state to avoid surface impact once onboard sensors (e.g., an altimeter) detect that the actual surface is much closer than originally expected and a switch is made from standard mode into safety mode. From a safety or target state, higher-level mission algorithms or ground-based commands (i.e., remote flight operators communicating with the spacecraft from Earth) can determine and instruct a new course of action for the spacecraft G&C. For example, once the spacecraft is in the proximity of the target state, a local contact controller could be switched on to guide the spacecraft to the surface for sample acquisition.

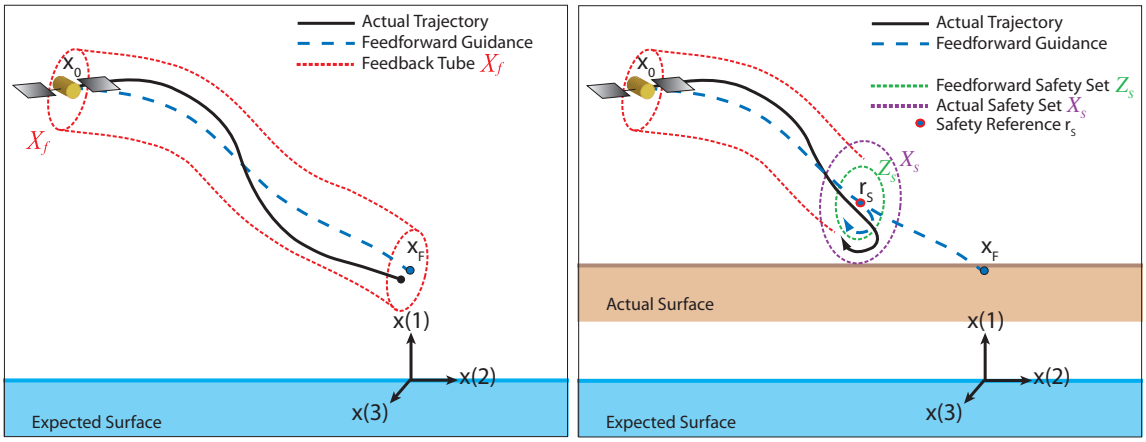


Figure 6.1: Operational modes of the guidance and control algorithm: standard mode (left) and safety mode (right)

Mathematical details on the G&C design for tube $\mathbf{X}_f$, sets $\mathbf{Z}_s$ and $\mathbf{X}_s$, and r_s from Figure 6.1 are provided in Chapter 5. The spacecraft sensors are assumed to have a measurement

range beyond the safety-mode region $\mathbf{X}_s$. The remaining state constraints $\mathbf{X}$ and $\mathbf{Z}_n$ (not shown in the figure) that define the discrete SR-MPC algorithm in Chapter 5 are as follows: constraint set $\mathbf{X}$ on the actual system would be all positions above the expected surface; and, constraint set $\mathbf{Z}_n$ on the nominal system would be more-conservative position and velocity constraints defined such that $\mathbf{Z}_n + \mathbf{X}_s \subseteq \mathbf{X}$.

6.1 Formulation for Applying Discrete SR-MPC Algorithm

Application of the discrete SR-MPC algorithm to small-body proximity operations requires formulating the governing dynamics in the framework of SR-MPC. Additionally, specific requirements on the control inputs and the cost function need to be cast appropriately.

6.1.1 Spacecraft Dynamics

The equations of motion for the actual spacecraft dynamics are expressed in the form of (5.1) from Chapter 5. The dynamics are formulated in a rotating frame and represent the spacecraft in orbit about a small celestial body with a constant rotation rate:

$$\dot{x} = Ax + Bu + B\phi\left(C_q x\right) + Bd, \qquad x = \begin{pmatrix} r \\ \dot{r} \end{pmatrix}, \tag{6.1}$$

where

$$A = \begin{bmatrix} 0 & I \\ -\widehat{\omega}^2 & -2\widehat{\omega} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad C_q = \begin{bmatrix} I & 0 \end{bmatrix},$$

$r, \dot{r} \in \mathbb{R}^3$ are spacecraft position and velocity relative to the small-body center of mass, $\phi(\cdot) \in \mathbb{R}^3$ is gravitational acceleration as a function of spacecraft position, $u \in \mathbb{R}^3$ is the spacecraft thrust-acceleration input, and $d \in \mathbb{R}^3$ lumps other bounded exogenous disturbances. Variable $\widehat{\omega} \in \mathbb{R}^{3\times 3}$ is a matrix representation of the vector cross product $\omega \times (\cdot)$, where $\omega \in \mathbb{R}^3$ is the constant small-body rotation rate ($\dot{\omega} = 0$).

The nominal model of the spacecraft dynamics is expressed in the form of (5.2):

$$\dot{z} = Az + Bu_o + B\psi\left(C_q z\right), \qquad z = \begin{pmatrix} p \\ \dot{p} \end{pmatrix}, \tag{6.2}$$

where $z \in \mathbb{R}^6$ is the nominal position p and velocity $\dot{p}$, $u_o \in \mathbb{R}^3$ is the nominal control input, and $\psi(\cdot) \in \mathbb{R}^3$ is a model approximating the actual gravity ϕ.

6.1.2 Control Input Requirements

From the architecture of the SR-MPC algorithm, the control input to actual system (6.1) is split into separate feedforward and feedback components:

$$u(t) = u_o(t) + u_f(t) \tag{6.3}$$

where the feedforward $u_o(t)$ is designed for each mode, standard and safety, by using the known nominal system (6.2), and feedback $u_f(t)$ is designed to maintain the uncertain actual system (6.1) states within the proximity of the feedforward nominal states. The specific details of the design for each component are provided in Chapter 5.

The optimization criteria for the standard mode is to reach the target state with a minimal fuel usage, which in the FHC-D in Chapter 5 sets $(\alpha, \beta) = (1, 0)$ to minimize the 1-norm of the nominal input u_o; in contrast, $(\alpha, \beta) = (0, 1)$ would minimize input thrust-acceleration energy.

For small-body proximity operations, a design requirement for thruster silence time is placed on the development of the control inputs in (6.3) [10]. After each finite burn time of δ_f seconds, a minimum thruster silence period of δ_s seconds is required. This specification provides thruster silence for other online algorithms (e.g., estimators, image sensors) that benefit from minimal operation interruptions. The following input profile is used for each time step (where $t_{k+1} - t_k = \Delta t$) in the discrete SR-MPC algorithm:

$$u(t) = \begin{cases} u_k, & t \in [t_k, t_k + \delta_f] \\ 0, & t \in (t_k + \delta_f, t_{k+1}) \end{cases}, \quad k \in \mathbb{Z}_0^+, \tag{6.4}$$

where δ_f is the fixed firing time and $u_k \in \mathbb{R}^3$ is constant (but may differ for each k). Note, $\Delta t \geq \delta_f + \delta_s$ by design. Since the control is split into two components (feedforward and feedback), u_k in (6.4) is made up of two components as well: in standard mode, $u_k = u_{o,k} + u_{f,k}$; in safety mode $u_k = u_{os,k} + u_{f,k}$.

The discrete SR-MPC algorithm formulation in Chapter 5 assumes a zero-order hold per time step for the inputs and nonlinearities. The input profile from (6.4) is not fixed for

the entire time step but its form is readily incorporated in the existing discrete SR-MPC algorithm formulation in Chapter 5 for both feedforward and feedback components. To explain this, the derivation of the standard-mode discrete nominal model (5.48) will be provided, starting with the continuous-time model in (5.46):

$$\dot{z} = F_k z + B u_o + B\psi_k. \tag{6.5}$$

Refer back to Section 5.2.3 in Chapter 5 for an full explanation of time-step varying matrix F_k and gravity vector ψ_k; they essentially incorporate a linearization of gravity model $\psi(\cdot)$ to improve the nominal trajectory computations in standard mode. The general continuous-time solution to (6.5) for $t \geq t_0$ has the form

$$z(t) = e^{F_k(t-t_0)} z(t_0) + \int_{t_0}^{t} e^{F_k(t-\tau)} B(u_o(\tau) + \psi_k) d\tau. \tag{6.6}$$

During the thrusting portion of a time step, $t \in [t_k, t_k+\delta_f]$, the general solution at $t = t_k+\delta_f$ can be written as

$$z(t_k + \delta_f) = e^{F_k\delta_f} z(t_k) + \int_{0}^{\delta_f} e^{F_k(\delta_f-\tau)} B d\tau \cdot (u_{o,k} + \psi_k), \tag{6.7}$$

where vectors ψ_k and $u_{o,k}$ are constant and removed from the integral. Note, $u_o(t) = u_{o,k}$ per the formulation in (6.4). During the silence portion of a time step, $t \in [t_k + \delta_f, t_{k+1}]$, the general solution at $t = t_{k+1}$ can be written as

$$z(t_{k+1}) = e^{F_k(\Delta t-\delta_f)} z(t_k + \delta_f) + \int_{\delta_f}^{\Delta t} e^{F_k(\Delta t-\tau)} B d\tau \cdot \psi_k, \tag{6.8}$$

where $\Delta t = t_{k+1} - t_k$, ψ_k remains the same constant as in (6.7), and $u_o(t) = 0$ per the formulation in (6.4). Combining (6.7) and (6.8) produces

$$z(t_{k+1}) = e^{F_k\Delta t} z(t_k) + e^{F_k(\Delta t-\delta_f)} \int_{0}^{\delta_f} e^{F_k(\delta_f-\tau)} B d\tau \cdot u_{o,k} + \int_{0}^{\Delta t} e^{F_k(\Delta t-\tau)} B d\tau \cdot \psi_k, \tag{6.9}$$

or

$$z_{k+1} = A_k z_k + B_k u_{o,k} + E_k \psi_k, \tag{6.10}$$

which is the desired form of (5.48), with matrix B_k capturing the silent-time requirement

on the input as given in (6.4).

6.1.3 Other Application-Specific Implementation Issues

The algorithm formulation for standard mode incorporates Conditions 5.4 and 5.5. For small-body proximity operations, these conditions bound the gradient of the gravity model $\psi(\cdot)$ and bound the difference between that model and the actual gravity $\phi(\cdot)$, respectively. In reality, these conditions cannot be satisfied everywhere in position-space. This is easily seen by considering a typical point-mass gravity field model (e.g., $\psi = \frac{\mu}{r^2}$ with μ the small-body gravity constant) that tends toward infinity in the closed neighborhood around the origin. In actuality, close to the origin, this particular gravity model is invalid. The justification of these conditions comes from the idea that proximity operations take place in a restricted region of position-space specified by the constraints $\mathbf{X}$ and $\mathbf{Z}_n$ that the feedforward standard mode enforces to maintain the spacecraft away from the center-of-mass and from collision with the small body. Thus, the gravity field is bounded and Conditions 5.4 and 5.5 can be satisfied.

The safety reference r_s for this application is defined as a state of rest/hover (zero velocity) at the standard-mode nominal position at safety switch-on time:

$$r_s = T_s z_{stan}(t_s) = \begin{pmatrix} p_s \\ 0 \end{pmatrix}, \quad T_s = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \tag{6.11}$$

where z_{stan} implies the nominal state z during standard mode, which is the operation mode of the system prior to safety switch-on at time t_s, and $p_s = p(t_s)$ from (6.2).

The safety-mode nominal dynamics in Chapter 5 are given in (5.34) as

$$\dot{\tilde{z}} = A\tilde{z} + Bu_{os} + Ar_s + B\psi(C_q\tilde{z} + C_q r_s), \tag{6.12}$$

where $\tilde{z} \triangleq z - r_s$. The term Ar_s in the small-body application represents the centripetal acceleration g_c that affects holding the spacecraft in a fixed position in a rotating frame; the term also arises from shifting the origin of (6.2) to $z = r_s$. The form of matrix A and vector r_s for this spacecraft application fits Condition 5.6, which allows for the subsequent bounding argument of Condition 5.7 to combine gravity and centripetal acceleration ($\bar{g} = g_c + \psi(C_q\tilde{z} + C_q r_s)$) under a common, worst-case bound δ. Refer back to these conditions for

specifics of the formulation. Note, δ can be selected as follows: if $||\psi(\cdot)|| \leq \delta_1$ and $||g_c|| \leq \delta_2$ over the region of the proximity operation, then $||\bar{g}|| \leq ||\psi(\cdot)|| + ||g_c|| \leq \delta_1 + \delta_2 \leq \delta$.

Re-solvability of the discrete SR-MPC algorithm is best performed in a compressing-horizon implementation. The convexification of the discrete SR-MPC algorithm imposes a discrete number of steps N to drive the system to the target state, with a cost function only on the control input. With the objective being to minimize fuel over the N steps, the algorithm logically chooses to maintain the spacecraft away from the target state for as long as possible, which corresponds to a lower gravitational force and thus lower fuel expenditure. Descent to the target state occurs only toward the end of the planning horizon. Thus, compressing-horizon provides a means of minimizing overall fuel usage and maneuver time while also obtaining the benefits of model predictive control by using current state measurements on re-solves to improve the remaining control inputs.

6.2 Simulation of Asteroid Landing

The SR-MPC algorithm is demonstrated for a spacecraft descent toward an asteroid in preparation for landing. The asteroid gravity and rotation rate are based on data from asteroid Eros [52]. This section will provide a comparison of the algorithm performance with and without safety mode; this comparison is between the discrete SR-MPC algorithm in this thesis and an earlier version [10] without the safety mode available. The results will show that incorporation of the safety mode enables autonomous spacecraft response and mitigation of risk posed by an altitude error in the state constraints

To provide additional realism to the scenario, the spacecraft mass changes with thruster firing, the thrusters are given thruster-execution error (up to 10% error from commanded thrust), the assumed sensors (inertial measurement unit and altimeter) have added random noise, and a random disturbance is also imparted to simulate solar radiation pressure, outgassing (from the spacecraft or asteroid), and other small forces that can affect the spacecraft. To handle these noise sources, a Kalman filter is implemented to provide inertial state estimates that are used for the actual states in the G&C algorithm.

The safety-mode is triggered on the altimeter, which is smoothed independently through a low-pass filter as a proof of concept for the algorithm. Triggering off of noisy measurements can lead to early or late safety-mode switch on, the later of which could pose unrecoverable

mission risk. Note, a similar delay in safety-mode switch on can occur from the discrete-time measurement interval, which can delay detection of a changed state constraint. An assumption in the SR-MPC algorithm is that the sensors have perfect knowledge within the specified safety constraints. To handle the additional sensor noise and discretization-induced delay, the detection range of the sensor is chosen larger than the specified safety constraints.

Knowledge of the surface is assumed to be in error by 100 meters, which causes an incorrect state constraint to be imposed in the standard-mode guidance planning. The gravity model of the asteroid is assumed to be accurate to within 5%. The navigation error is assumed to be within ± 5 m and ± 2 cm/s in position and velocity, respectively, relative to the small-body coordinate frame. The altimeter error is modeled as random noise with a standard deviation of ± 40 m (perhaps excessive, but utilized to test the safety mode). The altimeter range capability is 200 meters (i.e., the range at which altimeter data is considered useful for triggering safety). These values are arbitrarily chosen for the sake of demonstrating the SR-MPC algorithm in an example; real altimeters can have significantly different range capabilities.

The simulations were written in Matlab, with SDPT3 [50] used to efficiently re-solve the second-order cone program that generates the standard-mode feedforward control policy in discrete SR-MPC; these re-solves took less than 0.8 seconds. The maneuver simulation has a desired completion time that is accomplished by applying the SR-MPC algorithm in a compressing-horizon manner, with re-solves every two time steps. The simulation parameters and a depiction of the coordinate system (in Figure 6.2) are as follows:

Parameter	Value
Spacecraft mass (m)	400 kg
Specific impulse (I_{sp})	300 sec
Max. feedforward (U_{max})	$\frac{125}{m}$ m/s^2
Max. feedback (V_{max})	$\frac{20}{m}$ m/s^2
Silence time (δ_s)	15 sec
Firing time (δ_f)	15 sec
Maneuver time (T_M)	300 sec

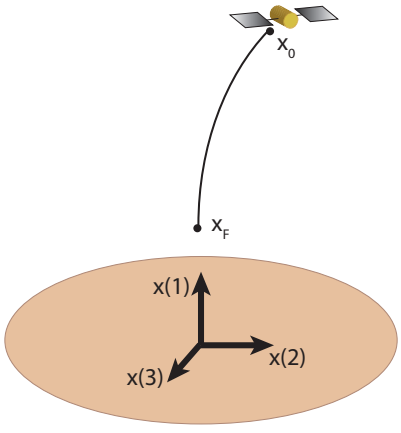


Figure 6.2: Parameters and coordinate system for simulation.

The maximum feedforward thrust acceleration (U_{max}) and maximum feedback thrust acceleration (V_{max}) define the control constraint sets from (5.9) and (5.10), respectively. Based on the parameters above, the simulation time step is $\Delta t = \delta_f + \delta_s = 30$ sec. The maneuver initial state x_0 and desired target state x_F are

$$x_0 = (\ 8950,\ \ 100,\ \ 0,\ \ 1.5,\ \ 2,\ \ 0\ )^T \quad \text{and} \quad x_F = (\ 8450,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0\ )^T, \tag{6.13}$$

in units of (m, m/s). These states specify a descent maneuver toward the asteroid surface, along the $x(1)$ radius coordinate. The surface position along this coordinate is $x(1) = 8340$ m. The standard-mode nominal state constraints $\mathbf{Z}_n$ and the feedback position constraints $\mathbf{X}_f$ are defined so that the combined position states keep the spacecraft from impacting the surface: $z(1) \geq 8350$ m for $\mathbf{Z}_n$ and $\{\eta(1), \eta(2), \eta(3)\} \leq \frac{1}{3}$ m for $\mathbf{X}_f$. Further, the feedback constrains the actual velocity around the nominal guidance to within $\frac{1}{20}$ m/s. The target state x_F is far enough from the surface so that safety can be activated to keep the spacecraft nominal state within a 70 m offset from the activation altitude once the altimeter picks up the surface error (from 200 m away).

In set notation, these are

$$\begin{aligned}
\mathbf{Z}_n &= \{z : c_1^T z(t) \geq 1\},\ \text{with } c_1 = (\ 1/8350,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 0\ )^T \\
\mathbf{X}_f &= \{\eta : a_1^T \eta(t) \leq 1\},\ \text{with } a_1 = (\ 1/3,\ \ 1/3,\ \ 1/3,\ \ 1/20,\ \ 1/20,\ \ 1/20\ )^T \\
\mathbf{Z}_s &= \{\tilde{z} : b_1^T C_s \tilde{z}(t) \leq 1\},\ \text{with } b_1 = (\ 1/70,\ \ 0,\ \ 0\ )^T
\end{aligned}$$

where $C_s = \begin{bmatrix} I & 0 \end{bmatrix}$, which allows safety to be specified for the position states alone. Note, the nominal terminal state set was specified as $\mathbf{\Omega}_n = \mathbf{Z}_s$.

Offline design of gains K_s and K_f for safety and feedback, respectively, are based on the discrete SR-MPC theorems in Chapter 5. With respect to the safety gain K_s, satisfaction of the inequalities in Theorem 5.2 is part of the design process for simultaneously selecting the design parameters for firing time δ_f, silence time δ_s, and the safety state constraints in $\mathbf{Z}_s$. If a desired set of constraints cannot satisfy the inequalities, then the inequalities themselves can be used in the design process. The same is true for feedback gain K_f and the firing and silence times, along with the feedback constraint definition for $\mathbf{X}_f$. The final design settings for the constraints then influence the nominal state constraints in $\mathbf{Z}_n$ that

will influence how close to the surface the target state x_F in Figure 6.2 can be so that safety can still be guaranteed.

Consideration must also be given to the fact that the algorithm is based on a discretization of the dynamics, so additional conservatism is necessary in specifying the target state since thruster firings only occur on a discrete interval yet the spacecraft continues to drift due to gravity in between the discrete time steps. This can be handled through shorter time steps, changes to the maneuver time T_M, or putting a constraint on the nominal descent velocity through $\mathbf{Z}_n$ to provide more conservatism so that the spacecraft does not move too quickly toward the surface between discrete time steps.

Figure 6.3 depicts the descent scenario where safety mode is activated at 210 seconds due to an error in the constraints specifying where the asteroid surface is located. Since the altimeter range readings come online at 200 m from the surface, once the error is discovered in the surface position, safety-mode is instantly activated. The immediate activation is a conservative design decision to aid with the discrete time steps governing the algorithm and thruster firings.

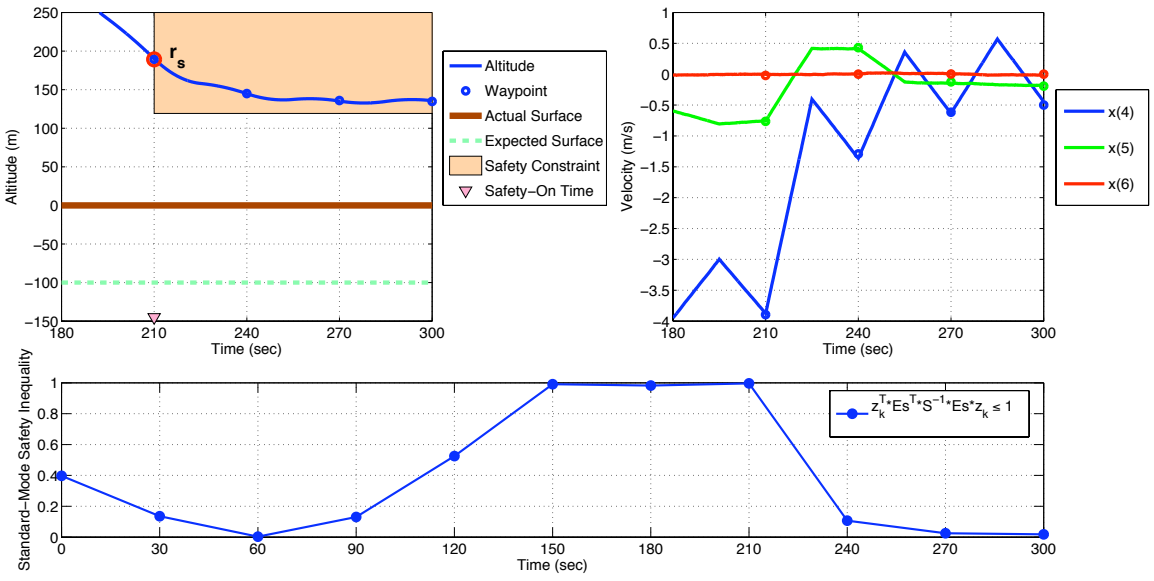


Figure 6.3: Safety mode activated upon altimeter identifying incorrect surface constraint.

The altitude profile (upper left plot in Figure 6.3) shows the safety constraint $\mathbf{Z}_s$ turned on at a safety-activation time of 210 seconds. This corresponds to the altimeter identifying that the actual surface is 100 m closer to the spacecraft than the expected surface utilized in the state constraints $\mathbf{Z}_n$ for the standard-mode guidance policy. The safety constraint in

$\mathbf{Z}_s$ tells the safety-mode control to maintain the spacecraft within 70 meters of the actual altitude (approximately 175 m) at safety-activation time; thus, the spacecraft is held above an altitude of 105 meters as seen in the plot.

The velocity profiles (upper right plot in Figure 6.3) are driven toward rest, which is the velocity reference state in safety reference r_s (Refer back to (6.11)). The spacecraft velocity cannot be held exactly at rest due to the thruster silence requirement, so instead the algorithm has the velocity oscillate about rest. Additionally, the thrust profile has the firing at the start of the time step, followed by silence until the next time step. Thus, the guidance waypoints for velocity predict what spacecraft velocity is expected after gravity causes the spacecraft to drift back toward the surface by the next thruster firing. The saw-tooth velocity is also indicative of the spacecraft being modeled as a double integrator with a constant on or off thrust acceleration, along with very weak gravity. Note, the thrust profile can also have the firing at the end of the time step (with silence first), which would switch the order of the thrust input profile in (6.4). The form of the integral that produces the B_k matrix in discrete dynamics (6.9) would also change.

The bottom plot in Figure 6.3 shows the safety constraint from the standard-mode FHC-D in Chapter 5, Section 5.2.3, that ensures safety mode is always available. Since the constraint remains less than 1, the safety feedforward controller can be activated at 210 seconds and ensure that the spacecraft remains near an altitude that is safe from surface impact.

The overall maneuver profile from the starting location x_0 through the maneuver time T_M is depicted in Figure 6.4. During standard mode, the feedforward u_o is recomputed every two time steps, in a compressing-horizon manner. The figure provides the spacecraft position and velocity (top two plots) in the asteroid coordinates from Figure 6.2, with the positions translated relative to the target position in x_F. The plots show that standard mode maneuvers the spacecraft toward the target state until the altimeter detects the surface location error in the constraints and activates safety mode (which is zoomed in on in Figure 6.3). A comparison of the actual trajectory with the guidance waypoints indicates the benefit of the FHC-D for standard mode and the improvement to guidance from incorporating a linearization of the gravity model and solving for waypoints in both position and velocity.

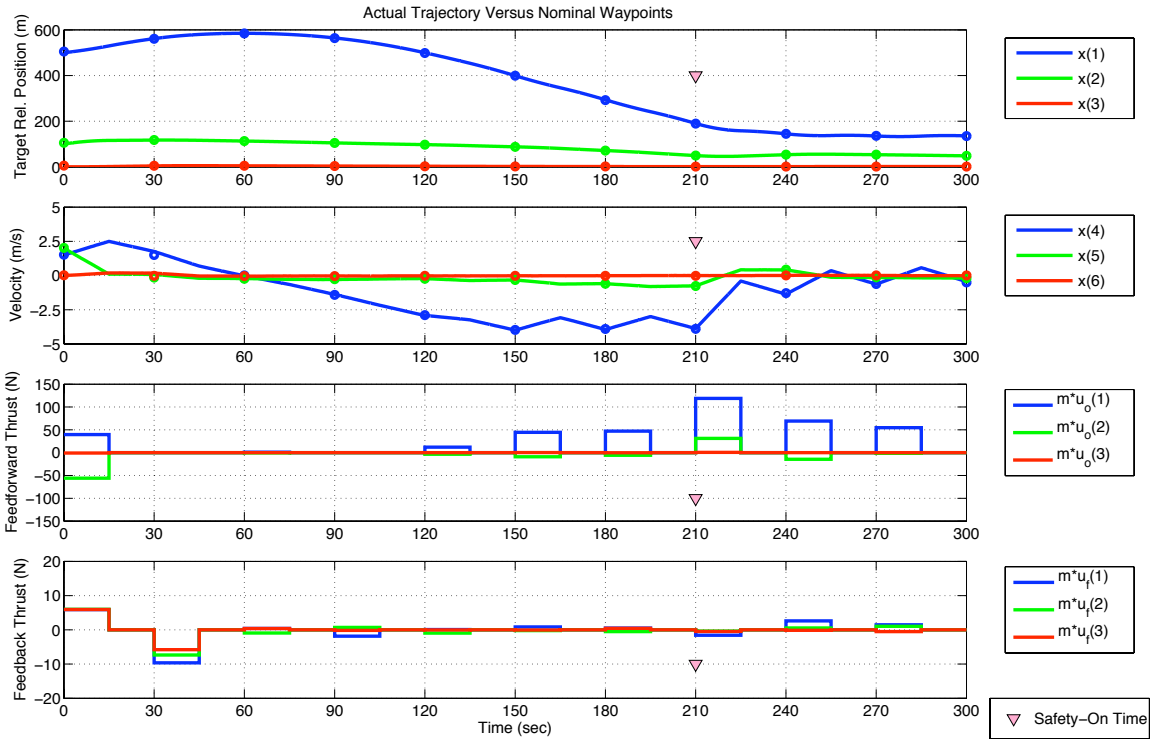


Figure 6.4: Maneuver profile for standard mode through safety-mode activation.

The bottom two plots in Figure 6.4 show the thrust profiles for the feedforward and feedback components, also along the asteroid coordinate directions. The feedforward profile shows that standard mode does not implement a thruster firing at each time step, which again indicates the benefit of the FHC-D waypoints and the conservation of fuel. Neither the feedforward nor the feedback components exceed the maximum thruster firings, $mU_{max} =$ 125 N and $mV_{max} = 20$ N, respectively. Even at safety-mode activation the constraints are obeyed. Notice that the feedforward thrust increases significantly at safety-activation time (210 sec), indicating the G&C algorithm brings the spacecraft into the safety hover mode at the current altitude.

As a means of comparison, a descent scenario is shown in Figure 6.5 where the FHC-D without the safety mode constraint (FHC-2) provides a guidance policy to maneuver the spacecraft to the target x_F. This algorithm assumes the expected surface position is correct, and there are no alternative operational modes considered in the algorithm architecture. As a result, the spacecraft impacts the asteroid surface near 270 seconds. In this particular example, the ground is placed right at the desired target state, so the surface impact occurs in between the discrete waypoints.

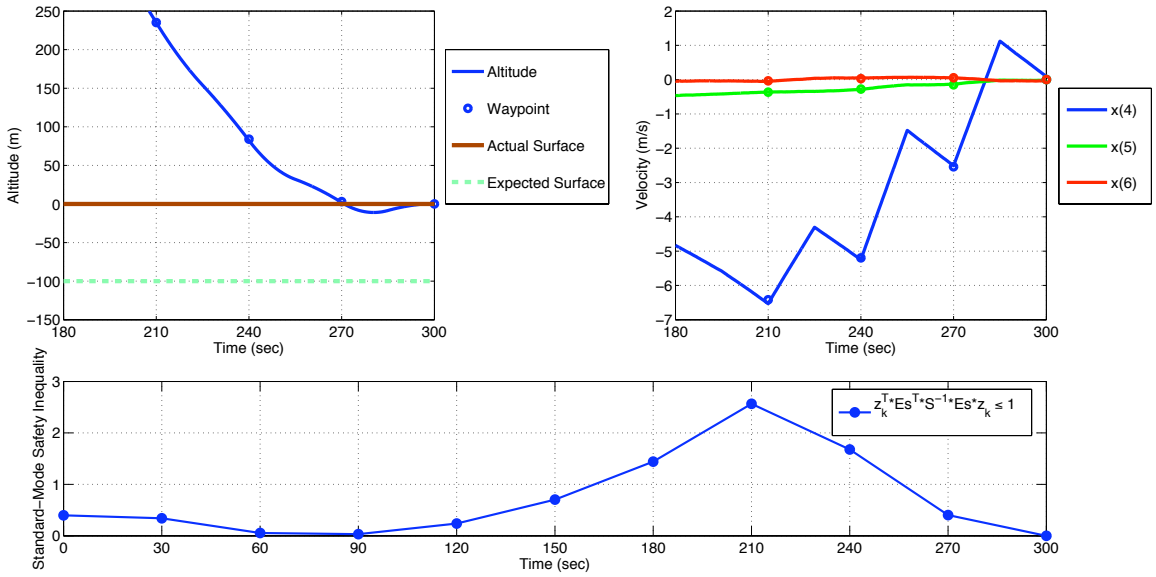


Figure 6.5: Standard-mode algorithm (without safety) causes surface impact due to incorrect surface constraint.

The bottom plot in Figure 6.5 shows the equivalent values of the safety constraint along the descent trajectory. The constraint is significantly violated during the descent between $160 - 260$ seconds because it has been removed from this implementation of the FHC-D. The constraint violation further indicates that the nominal guidance states are not inside the safety set $\mathbf{Z}_s$, so even if the safety mode feedforward policy were switched on, there is no guarantee that the controller could keep the spacecraft safe from impacting the ground. In fact, Figure 6.6 shows this very scenario, utilizing the same altimeter as in the prior example to trigger safety mode. The descent speed of the spacecraft is such that safety is switched on at 240 seconds, since the prior waypoint is above 200 m altitude and outside the altimeter measurement range. In order for the safety feedforward policy to keep the spacecraft from impacting the surface and maintaining the desired safety mode, the feedforward thrust constraint of 125 N (bottom-left plot in Figure 6.6) is significantly violated. If this constraint were the maximum thrust available from the onboard engines, then the spacecraft would likely impact the surface.

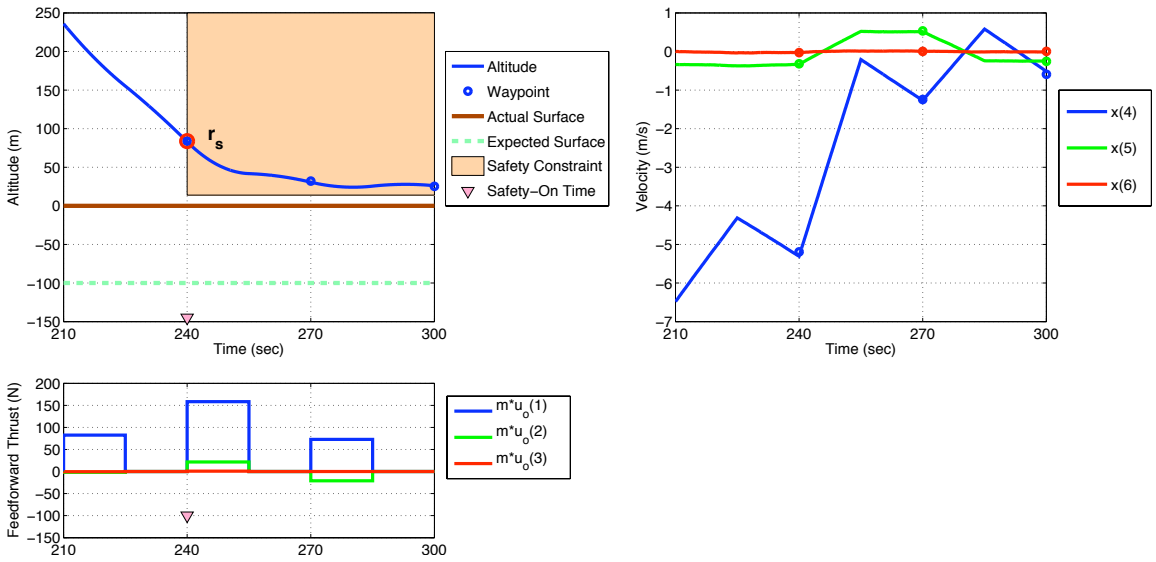


Figure 6.6: Feedforward thrust limit violated on safety activation if safety constraint is not enforced in standard mode.

The results from Figure 6.3 clearly indicate that implementation of the safety constraint (FHC-2) in the FHC-D for SR-MPC provides a better method for mission risk mitigation during autonomous proximity operations. The trade off with the active safety constraint (FHC-2) in the standard mode is that the algorithm must begin the spacecraft descent sooner, resulting in slightly higher fuel consumption with SR-MPC compared to R-MPC. Figure 6.7 juxtaposes the spacecraft trajectories (without a safety event occurring) resulting from descent scenarios that use the R-MPC and SR-MPC algorithm. The upper two plots in Figure 6.7 contrast the altitude and velocity profiles from R-MPC and SR-MPC, indicating that SR-MPC descends the spacecraft sooner and has a smaller maximum velocity than R-MPC. The more-conservative velocity in SR-MPC results from the enforced safety constraint (third plot from top). By descending sooner, the spacecraft experiences higher gravity and uses slightly more fuel, as seen in the bottom plot of Figure 6.7.

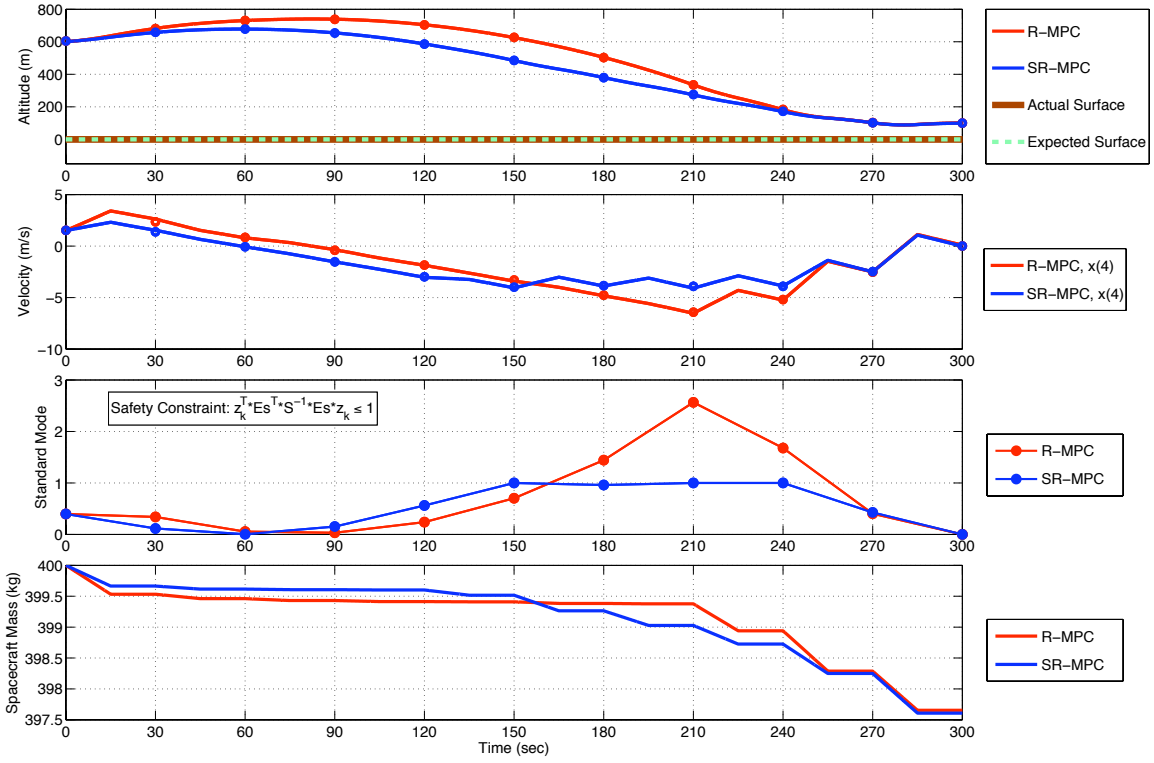


Figure 6.7: Comparison of R-MPC and SR-MPC performance.

6.3 Summary of Discrete SR-MPC Applied to Spacecraft Proximity Operations

The discrete SR-MPC algorithm has been applied to a detailed spacecraft application that incorporates a state estimator, noisy sensor measurements, external disturbances, and thrust magnitude errors. The simulations show that the algorithm maneuvers the spacecraft toward the desired target in a robust, safe, and re-solvable manner, providing some risk mitigation during proximity operations. The algorithm handles model uncertainty and disturbances, minimizes fuel or thrust-energy usage (in standard mode), and incorporates desired thruster silence times. Further, the maximum-thrust constraints are obeyed in both modes, as are the prescribed state constraints (i.e., maintain the spacecraft away from the expected surface location). As shown in the simulations, if sensor information from the altimeter invalidates the expected standard-mode state constraints (i.e., the surface is closer in proximity than expected), the SR-MPC safety mode maintains the spacecraft in a safety state that avoids impact with the surface.

Chapter 7

Conclusions

The motivation for this thesis research was the development of control algorithms suitable for online implementation in engineering applications (e.g., aerospace and mechanical vehicles) that require adherence to state and control constraints, as well as robustness to uncertainty affecting both the system dynamics and the prescribed state constraints. The research builds upon the MPC (Model Predictive Control) algorithm, which makes use of a nominal dynamics model in a finite-horizon optimization that enforces both state and control constraints and generates a feedforward control policy. The finite-horizon optimization is advantageous given the finite online computational capabilities in practical applications. Additionally, recursively re-solving the finite-horizon optimization in a compressing- or receding-horizon manner provides a form of closed-loop control by setting the nominal state at re-solve to the current actual state.

Uncertainty between the nominal model and the actual system dynamics, along with constraint uncertainty can cause feasibility issues during the re-solves and, thus, robustness problems with the MPC algorithm. The chapters of this thesis successively build upon a baseline MPC algorithm outlined in Chapter 2. The contributions, by chapter, are summarized below, followed by a discussion of further research directions that can improve upon the MPC framework and the algorithms presented in this thesis.

7.1 Summary of Algorithms

7.1.1 R-MPC Algorithm

The R-MPC (Robust and re-solvable MPC) algorithm developed in Chapter 3 for continuous-time systems enhances the baseline MPC by augmenting the framework with a separate

feedback control component that provides robustness to uncertainty and disturbances affecting the actual dynamics. The feedback component creates an invariant tube about the nominal feedforward trajectory, based on an explicit characterization of the uncertainty between the actual system and the nominal model. The feedback tube maintains the actual state in the proximity of the nominal feedforward trajectory for all time. Additionally, the online optimization for computing the feedforward control component utilizes the tube to relax the initial nominal state for each re-solve. This relaxation allows the initial nominal state to simply be within the tube cross section and not fixed to the actual state at the re-solve time, which provides robust feasibility and guaranteed re-solvability to the R-MPC algorithm. This is an advantage over the baseline MPC algorithm in Chapter 2, which requires the nominal state to be identical to the actual state at the re-solve time.

7.1.2 SR-MPC Algorithm

The continuous-time SR-MPC algorithm in Chapter 4 augments the R-MPC algorithm with a separate, reactive, safety control policy that provides robustness (or safety) to uncertainty/changes in static state constraints. The improved algorithm incorporates two operation modes. The primary mode, standard mode, implements a modified version of the R-MPC algorithm that incorporates a nominal state constraint that ensures the secondary mode availability at any time along the planning horizon. The secondary mode, safety mode, provides an alternate control policy, if needed, to maintain the system in an invariant set to ensure avoidance of changed state constraints. The SR-MPC safety mode provides an improvement over both the baseline MPC and R-MPC algorithms, which assume that the state constraints imposed in the control objectives are static and known perfectly ahead of the time. In cases where uncertainty exists in these constraints, avoidance of the changed constraint cannot be guaranteed in either MPC or R-MPC but can be in SR-MPC.

7.1.3 Discrete SR-MPC Algorithm

The discrete-time SR-MPC algorithm in Chapter 5 is based on the continuous-time SR-MPC algorithm. The discrete version is developed for practical applications that require a computer both to solve online the standard-mode finite-horizon optimization and to implement the standard-mode and safety-mode control policies.

The finite-horizon optimization is formulated in discrete SR-MPC as a second-order cone program that can be solved online in a computationally efficient and accurate manner by using interior-point algorithms. The optimization additionally incorporates the constraints from continuous-time R-MPC and SR-MPC that relax the initial state and ensure safety-mode availability, respectively. The discrete feedback policy, similar to that in continuous-time SR-MPC, guarantees resolvability of the discrete finite-horizon optimization problem and is formulated to ensure the actual state is within a specified set about the nominal discrete states at all discrete times. Additionally, the discrete safety control policy is formulated to guarantee the actual state remains within the prescribed safety set at all discrete times upon safety-mode activation.

7.2 Summary of Autonomous Spacecraft Application

An application of the discrete SR-MPC algorithm was demonstrated in Chapter 6 for an autonomous spacecraft descending toward the surface of an asteroid with an uncertain gravity field and an uncertain surface topography. The formulation of the discrete SR-MPC algorithm assumes perfect state knowledge. To exercise the algorithm in a more realistic implementation, noisy sensor measurements were implemented, along with process noise affecting the dynamics and thrust inputs. These noise sources were then filtered to provide state estimates.

The simulations demonstrate that the discrete SR-MPC algorithm provides thrust inputs that maneuver the spacecraft toward the desired target in a robust, safe, and re-solvable manner, providing some risk mitigation during proximity operations such as descent. Per the formulation, the algorithm handles model uncertainty and disturbances, minimizes fuel or thrust-energy usage, and incorporates desired thruster silence times as specified for the application. Additionally, the maximum-thrust constraints and the prescribed state constraints (i.e., maintain the spacecraft away from the expected surface location) imposed on the algorithm design were obeyed in both modes of the SR-MPC algorithm. As shown in the simulations, if sensor information from the altimeter invalidates the expected standard-mode state constraints (i.e., the surface is closer in proximity than expected), the SR-MPC safety mode maintains the spacecraft in a safety state that avoids impact with the surface.

7.3 Some Limitations and Potential Extensions

The theoretical development of the continuous-time R-MPC algorithm (and standard mode of the SR-MPC algorithm) is focused on proving the origin is asymptotically stable under application of the control input. As such, the origin is within the planning horizon of the FHC (Finite-Horizon optimal Control problem), whereas for many practical applications this may not be realizable. Computational limitations or limited state-constraint knowledge in practical applications can necessitate a shorter planning horizon such that the FHC does not include the desired target state during the initial feasible solution. Subsequent FHC re-solves shift the terminal set toward the desired target state, but there is no mathematical feasibility guarantee. The lack of this guarantee is well known and research inroads toward resolving this issue would be valuable to the MPC community. Regardless, applying the R-MPC and SR-MPC algorithms in this type of scenario is a natural application extension. In fact, the safety mode in SR-MPC could also be used as a backup control policy to ensure system safety if a subsequent re-solve is infeasible, which is the function of safety in the RSBK (Robust Safe But Knowledgeable) algorithm by Kuwata *et al.* [24].

The splitting of the control policy and control constraints in R-MPC and SR-MPC into the separate feedforward and feedback segments poses conservatism in the FHC feedforward optimization. The cross section of the feedback-generated invariant tube is fixed; for the specialized systems of Section 3.3, the cross section is essentially set by the worst-case bound on the model mismatch with the actual system. The capability to trade control authority between the feedforward and feedback components (allowing the tube cross section to vary) would be advantageous, especially if additional structure is known about the uncertainty or disturbances in different feasible operating states. For instance, when uncertainty or disturbances are smaller, less feedback is required to follow the feedforward trajectory. The feedforward control policy could then be more aggressive (and perhaps more optimal) if some of the feedback control authority could shift and relax the feedforward control constraints. Further work is necessary to develop mathematical frameworks that might allow this extension.

The reactive safety mode in the SR-MPC algorithm is defined through a safety set that also has a fixed cross section. This safety set essentially creates an additional tube about the feedforward trajectory generated in the SR-MPC standard mode that can be overly

conservative due to the fixed cross section. This safety tube must fit within any state corridor chosen in the FHC feedforward optimization, which can limit the choices for path planning. Allowing the safety-set cross section to vary (i.e., proactive safety based on the current state) would alleviate this limitation. For instance, in mechanical vehicles with the safety set defined by position constraints that ensure a maximum stopping distance, the SR-MPC algorithm would implicitly bound the maximum velocity. The safety tube would have a fixed cross section requiring any corridor chosen by the FHC feedforward path planning to be traversable at the maximum velocity. This is excessively restrictive, particularly if a vehicle can traverse a more optimal corridor at a slower velocity, at which the vehicle can be safely stopped within a shorter distance. The development of a proactive safety mode than can allow the safety-tube cross section to vary based on the current state (e.g., perform a trade between the feedforward velocity and the safe stopping distance) would add to the number of potential corridors.

Some of the remaining implementation issues for application of R-MPC and SR-MPC in practical engineering systems include the following: incorporating measurement uncertainty, safety from dynamically changing state constraints, restarting standard mode after safety activation, and reliability of optimization software for generating FHC solutions. The feedback and safety-mode control policies in this research assume full-state knowledge. Presumably, the feedback-invariant tubes can be extended to also incorporate knowledge uncertainty, however, this was not considered in the research. Extensions might be aided by the literature on output-feedback methods. Regarding dynamically changing state constraints, the issue is potential violation of the invariant safety set (e.g., adversarial vehicles entering the safety set). This sort of extension could consider blending additional operational modes into the SR-MPC framework. The additional modes could add various evasive maneuvers that, after execution, reconnect to the prior feedforward trajectory, thus maintaining future re-solve feasibility. These extra modes would add further conservatism to the feedforward optimization but provide some additional capabilities.

Methods for restarting the SR-MPC algorithm after activation of safety mode were also not considered in this thesis. Essentially, a new feasible solution to the feedforward optimization is required. If the prior standard-mode solution becomes feasible again (e.g., another vehicle temporarily blocked the standard-mode path), then some SR-MPC applications with mechanical systems can restart standard mode by bringing the actual state back

into the feedback invariant tube centered on the standard-mode state at safety-activation time. This is not a part of the current SR-MPC algorithm, and a general framework for this type of restart is not straight forward for all systems, primarily because the safety-set center is generally offset from the standard-mode state at safety-activation time.

The feedback and safety-mode control policies are developed offline, so only the standard-mode feedforward FHC requires significant online computational resources. To address computational demand, the discrete SR-MPC algorithm in Chapter 5 was formulated as a second-order cone program, which can be solved efficiently online with interior-point algorithms. For many practical applications, reliability of the software that implements interior-point algorithms is essential. For instance, in spacecraft applications, unreliable software poses unacceptable risk to generally expensive and one-of-a-kind missions. Thus, some practical applications would be concerned with verification and validation of the optimization software prior to implementation of R-MPC or SR-MPC.

The framework for discrete SR-MPC guarantees satisfaction of the standard- and safety-mode state constraints only at discrete times. In between the discrete time steps, a continuous-time trajectory may actually violate these constraints. This is a known problem in applying discrete-system-based control policies to continuous-time systems. A common approach is to further restrict state constraints in the discrete algorithms so that applying the discrete control policies to continuous-time systems does not violate the state constraints. This approach is used in the spacecraft example of Chapter 6 where establishing more conservative state constraints for the discrete SR-MPC algorithm was relatively straight forward. However, more complicated applications might not be as straight forward. Research developments that provide a more-constructive method for discretization with guarantees on continuous-time state-constraint satisfaction (and potentially specification of the discrete input format) would be a significant contribution.

Lastly, the discrete SR-MPC algorithm imposes a cost function only on the control input, and resulting proofs of completion of the control objectives are only provided for a compressing-horizon implementation. As such, the discrete algorithm is not as general as the continuous-time counterparts in this thesis. A more general discrete framework could be developed and provide stability guarantees in a receding-horizon manner.

Appendix A

Relevant Proofs for Continuous-Time Algorithms

A.1 Proof of Lemma 2.1: Re-Solvability of the FHC

Proof. (by induction) Suppose at t_{k-1} the FHC is feasible with T_{k-1} and provides $u_o^{k-1}(t)$ for $t \in [t_{k-1}, t_{k-1} + T_{k-1}]$. Let $z^{k-1}(t)$ be the trajectory of nominal system (2.1) for $t \in [t_{k-1}, t_{k-1} + T_{k-1}]$ corresponding to control input $u_o^k(t)$.

Let $t_k = t_{k-1} + \delta_k$ and re-solve the FHC with T_k. Since $F(\cdot)$ in nominal system (2.1) is known perfectly, the resulting nominal state $z(t)$ is exactly $z^{k-1}(t)$ from application of $u_o^{k-1}(t)$ for $t \in [t_k, t_{k-1}]$; thus, state $z^k(t_k) = z^{k-1}(t_k)$ remains the initial state of a feasible trajectory. Then, the following control input provides a feasible solution to the FHC re-solve:

$$u_o^k(t) = \begin{cases} u_o^{k-1}(t), & t \in [t_k, t_{k-1} + T_{k-1}] \\ \mathcal{L}(z(t)), & t \in [t_{k-1} + T_{k-1},\, t_k + T_k]\,, \end{cases} \tag{A.1}$$

where $T_k \geq T_{k-1} - \delta_k$, and thus $t_k + T_k \geq t_{k-1} + T_{k-1}$. Input $u_o^k(t) = u_o^{k-1}(t) \in \mathbf{U}_o$ provides $z^k(t) = z^{k-1}(t) \in \mathbf{Z}_n$ for $t \in [t_k, t_{k-1} + T_{k-1}]$. Since $z^k(t_{k-1} + T_{k-1}) = z^{k-1}(t_{k-1} + T_{k-1}) \in \mathbf{\Omega}_o$ and Condition 2.2 holds, then $\mathbf{\Omega}_o$ is invariant and $z(t) \in \mathbf{\Omega}_o$ with $u^k(t) = \mathcal{L}(z(t)) \in \mathbf{U}_o, \forall t \in [t_{k-1} + T_{k-1},\, t_k + T_k]$.

Thus, control policy (A.1) provides a feasible solution to the FHC re-solve at t_k for any $T_k \geq T_{k-1} - \delta_k$ once the FHC is feasible at t_{k-1} with T_{k-1}. □

A.2 Proof of Lemma 2.2: Shrinking Optimal Cost with Receding Horizon

Proof. Since the FHC is feasible at t_{k-1} with T_{k-1} and $u_o^{k-1}(t)$ provides the optimal cost J_{k-1}^*, then $u_o^{k-1}(t)$ can also be used to provide a feasible solution for the FHC at t_k with $T_k \in [T_{k-1} - \delta_k, T_{k-1}]$ (or any $T_k \geq t_{k-1} - \delta_k$) by using $u_o^k(t)$ in (A.1) from the proof of Lemma 2.1. So, $z^k(t) = z^{k-1}(t)$ is a feasible trajectory for $t \in [t_k, t_{k-1} + T_{k-1}]$.

From Definition 2.1 for the computation times, $t_k - t_{k-1} = \delta_k \geq \epsilon > 0$. Thus, $t_k + T_k \geq t_{k-1} + T_{k-1}$ when $T_k \geq T_{k-1} - \delta_k$. Note, $T_k \in [T_{k-1} - \delta_k, T_{k-1}]$ covers standard compressing- and receding-horizon implementations of MPC.

The cost at t_k with control input (A.1) can be written as

$$J_k = \int_{t_k}^{t_{k-1}+T_{k-1}} h\left(z^{k-1}(\tau), u_o^{k-1}(\tau)\right) d\tau \; + \int_{t_{k-1}+T_{k-1}}^{t_k+T_k} h\left(z^k(\tau), u_o^k(\tau)\right) d\tau \; + \; V\left(z^k(t_k + T_{k-1})\right).$$

Note, the second integral (with limits $t_{k-1} + T_{k-1}$ to $t_k + T_k$) is over a non-negative interval since $t_k + T_k \geq t_{k-1} + T_{k-1}$ as shown. At t_{k-1}, the optimal cost can be written as

$$J_{k-1}^* = \int_{t_{k-1}}^{t_k} h\left(z^{k-1}(\tau), u_o^{k-1}(\tau)\right) d\tau + \int_{t_k}^{t_{k-1}+T_{k-1}} h\left(z^{k-1}(\tau), u_o^{k-1}(\tau)\right) d\tau + V\left(z^{k-1}(t_{k-1} + T_{k-1})\right),$$

and thus $J_k - J_{k-1}^* =$

$$\int_{t_{k-1}+T_{k-1}}^{t_k+T_k} h\left(z^k(\tau), u_o^k(\tau)\right) d\tau - \int_{t_{k-1}}^{t_k} h\left(z^{k-1}(\tau), u_o^{k-1}(\tau)\right) d\tau + V\left(z^k(t_k + T_k)\right) - V\big(\underbrace{z^{k-1}(t_{k-1} + T_{k-1})}_{= z^k(t_{k-1} + T_{k-1})}\big).$$

Condition 2.2 implies the following with $u_o^k(t) = \mathcal{L}(z^k(t))$ on $t \in [t_{k-1} + T_{k-1}, t_k + T_k]$:

$$\int_{t_{k-1}+T_{k-1}}^{t_k+T_k} \dot{V}\left(z^k(\tau)\right) d\tau \; + \int_{t_{k-1}+T_{k-1}}^{t_k+T_k} h\left(z^k(\tau), u_o^k(\tau)\right) d\tau \leq 0$$

and

$$V\left(z^k(t_k + T_k)\right) - V\left(z^k(t_{k-1} + T_{k-1})\right) \; + \int_{t_{k-1}+T_{k-1}}^{t_k+T_k} h\left(z^k(\tau), u_o^k(\tau)\right) d\tau \leq 0.$$

This implies that

$$J_k - J_{k-1}^* \leq -\int_{t_{k-1}}^{t_k} h\left(z^{k-1}(\tau), u_o^{k-1}(\tau)\right) d\tau. \tag{A.2}$$

Given Conditions 2.1 and 2.3, if $||z|| \geq R$, then $h(z, u_o) = \rho > 0$. Since $t_k - t_{k-1} = \delta_k \geq \epsilon > 0$ by Definition 2.1 for the computation times, then

$$\int_{t_{k-1}}^{t_k} h(z_{k-1}(\tau), u_{o,k-1}(\tau)) d\tau \geq \underbrace{\rho\epsilon}_{\beta} > 0, \tag{A.3}$$

where $\beta > 0$ is independent of k.

Combining inequalities (A.2) and (A.3) shows that $J_k - J_{k-1}^* \leq -\beta < 0$, and since $J_k^* \leq J_k$, then

$$J_k^* - J_{k-1}^* \leq -\beta < 0.$$

□

A.3 Proof of Theorem 2.1: Closed-Loop Asymptotic Stability of MPC

Proof. Given the MPC algorithm and $z(t_0) \in \mathcal{R}_n$ such that the FHC is feasible with some $T = T_0$, suppose there exists $k \in \mathbb{Z}^+$ such that $z^{k-1}(t_{k-1}) \notin \mathbf{\Omega}_o$ and $z^{k-1}(t_k) \notin \mathbf{\Omega}_o$. Then, $z^{k-1}(t) \notin \mathbf{\Omega}_o$ for $t \in [t_{k-1}, t_k]$, and (2.7) holds. Consequently, if the nominal trajectory z does not enter $\mathbf{\Omega}_o$ in finite time, then there exists $k \in \mathbb{Z}^+$ such that $J_k^* < 0$, which is a contradiction. This, together with Condition 2.2, imply the existence of finite time $\tilde{t} \geq t_0$ such that $z(t) \in \mathbf{\Omega}_o,\ \forall t \geq \tilde{t}$.

Application of Step 3 in the MPC algorithm, $u_o = \mathcal{L}(z)$ for $t \geq \tilde{t}$, and use of Condition 2.2 imply

$$\lim_{t \to \infty} \|z(t)\| = 0$$

since V is a Control Lyapunov Function for nominal system (3.2) with $\dot{V}(z) < 0, \forall z$, except $\dot{V}(0) = 0$. Therefore, the closed-loop nominal system (2.1) converges asymptotically to the origin, $\forall z(t_0) \in \mathcal{R}_n$, with control input u_o given by the MPC algorithm. □

A.4 Proof of Corollary 3.1.1: Constructive R-MPC Inequalities for a Class of Systems

Proof.

Establish $\mathbf{\Omega}_o$ Invariance for System (3.10) when $x \in \mathbf{\Omega}_o$:

Pre- and post-multiplying (3.19) by $diag(Q^{-1}, I, I)$ and utilizing $K = YQ^{-1}$ from (3.25) gives

$$\begin{bmatrix} A^TQ^{-1}+Q^{-1}A+Q^{-1}BK+K^TB^TQ^{-1}+\mu Q^{-1}EE^TQ^{-1} & C^T+K^TD^T & C_q^T+K^TD_q^T \\ C+DK & -I & 0 \\ C_q+D_qK & 0 & -\mu I \end{bmatrix} \leq 0. \tag{A.4}$$

Utilizing multiple Schur complements [8] and $C^TD = 0$, this matrix inequality can be written as

$$\begin{bmatrix} \begin{pmatrix} A^TQ^{-1}+Q^{-1}A+Q^{-1}BK+K^TB^TQ^{-1}+C^TC \\ +K^TD^TDK+\frac{1}{\mu}(C_q+D_qK)^T(C_q+D_qK) \end{pmatrix} & Q^{-1}E \\ E^TQ^{-1} & -\frac{1}{\mu}I \end{bmatrix} \leq 0. \tag{A.5}$$

For $\zeta = (x^T, \phi^T)^T$, corresponding to the state x and nonlinearity ϕ for the actual system (3.10), taking $\zeta^T M \zeta$, where M is the matrix in the preceding inequality, gives

$$x^T\left(A^TQ^{-1}+Q^{-1}A\right)x + 2x^TQ^{-1}(Bu+E\phi) + \|Cx\|^2 + \|Du\|^2 + \tfrac{1}{\mu}\left(q^Tq - \phi^T\phi\right) \leq 0, \tag{A.6}$$

where $u = \mathcal{L}(x) = Kx$ from (3.25) and $q = C_qx + D_qu$ have been utilized.

Applying Lemma 3.2 with Condition 3.4, and making use of (3.17) and $\phi(0,t) = 0\ \forall t$, gives

$$q^Tq - \phi^T\phi \geq 0. \tag{A.7}$$

Further, $h(x,u) = \|Cx\|^2 + \|Du\|^2 \geq 0$ (which satisfies Condition 2.1). These inequalities imply

$$x^T\left(A^TQ^{-1}+Q^{-1}A\right)x + 2x^TQ^{-1}(Bu+E\phi) \leq 0, \tag{A.8}$$

which gives $\dot{V}(x) \leq 0$ for $V(x) = x^TQ^{-1}x$ as defined in (3.24). Thus, $V(x)$ is a Lyapunov function for the actual system (3.10) and establishes the invariance of $\mathbf{\Omega}_o = \{x : x^TQ^{-1}x \leq$

1} with control policy $u = Kx$ from (3.25). Further, inequality (3.21) ensures $\mathbf{\Omega}_o \subseteq \mathcal{Z}_\Omega$ (with $\mathcal{Z}_\Omega \subseteq \mathbf{Z}_n$ as defined in Condition 3.6) [8].

Establish Bounded Input $u \in \mathbf{U}_o$ when $x \in \mathbf{\Omega}_o$:

Pre- and post-multiplying the second LMI in (3.20) by $diag(Q^{-1}, I)$ and utilizing a Schur complement gives $Q^{-1} - K^T \Pi_o K \geq 0$, which is equivalent to

$$u^T \Pi_o u \leq x^T Q^{-1} x \tag{A.9}$$

after pre- and post-multiplying with x^T and x, respectively, and utilizing the feedback policy $u = Kx$ from (3.25). Thus, for $x \in \mathbf{\Omega}_o$, $u^T \Pi_o u \leq 1$, so $u \in \mathbf{U}_o$ from Condition 3.6. Thus, Corollary 3.1.1 establishes a framework and feedback policy satisfying Conditions 2.1 and 3.2 for system (3.10).

Establish $\mathbf{\Omega}_o$ Invariance for System (3.11) and Bounded Input $u_o \in \mathbf{U}_o$ when $z \in \mathbf{\Omega}_o$:

The above results apply identically to the nominal system (3.11) when $z \in \mathbf{\Omega}_o$ through the substitution of $\zeta = (z^T, \psi^T)^T$, with ψ corresponding to the nonlinearity for the nominal system (3.11), $u_o = \mathcal{L}(z) = Kz$, and $q_o = C_q z + D_q u_o$. Thus, Corollary 3.1.1 establishes a framework and feedback policy satisfying Conditions 2.1 and 2.2 for system (3.11).

Establish $\mathbf{X}_f$ invariance:

Let positive-definite function $V(\eta) = \eta^T P^{-1} \eta$ be a Lyapunov function candidate. Pre- and post-multiplying (3.18) by $diag(P^{-1}, I)$ and utilizing $K_f = LP^{-1}$ from (3.26) gives

$$\begin{bmatrix} A^T P^{-1} + P^{-1} A + P^{-1} B K_f + K_f^T B^T P^{-1} + P^{-1}/\lambda + (\beta + \lambda\gamma^2) P^{-1} E E^T P^{-1} & C_q^T + K_f^T D_q^T \\ C_q + D_q K_f & -\beta I \end{bmatrix} \leq 0. \tag{A.10}$$

Using multiple Schur complements, this matrix inequality can be written as

$$\begin{bmatrix} \begin{pmatrix} A^T P^{-1} + P^{-1} A + P^{-1} B K_f + K_f^T B^T P^{-1} \\ + \frac{1}{\beta} (C_q + D_q K_f)^T (C_q + D_q K_f) + P^{-1}/\lambda \end{pmatrix} & P^{-1} E & P^{-1} E \\ E^T P^{-1} & -\frac{1}{\beta} I & 0 \\ E^T P^{-1} & 0 & -\frac{1}{\lambda} \frac{1}{\gamma^2} I \end{bmatrix} \leq 0. \tag{A.11}$$

For $\zeta = (\eta^T, \pi^T, w^T)^T$, taking $\zeta^T M \zeta$, where M is the matrix in the preceding inequality, gives

$$\eta^T(A^TP^{-1} + P^{-1}A)\eta + 2\eta^TP^{-1}(Bu_f + E\pi + Ew) \\ + \tfrac{1}{\beta}\left((C_q\eta + D_qu_f)^T(C_q\eta + D_qu_f) - \pi^T\pi\right) + \tfrac{1}{\lambda}\left(\eta^TP^{-1}\eta - \tfrac{1}{\gamma^2}w^Tw\right) \leq 0, \quad \text{(A.12)}$$

where $u_f = K_f\eta$ from (3.26) has been utilized for the *error* dynamics in system (3.15).

Making use of relationship (3.16) for π and the properties of θ in (3.17) gives

$$(C_q\eta + D_qu_f)^T(C_q\eta + D_qu_f) - \pi^T\pi \geq 0, \quad \text{(A.13)}$$

thus

$$\eta^T(A^TP^{-1} + P^{-1}A)\eta + 2\eta^TP^{-1}(Bu_f + E\pi + Ew) + \tfrac{1}{\lambda}\left(\eta^TP^{-1}\eta - \tfrac{1}{\gamma^2}w^Tw\right) \leq 0. \quad \text{(A.14)}$$

From (3.13), $\frac{1}{\gamma^2}w^Tw \leq 1$, thus

$$\tfrac{1}{\gamma^2}w^Tw \leq \eta^TP^{-1}\eta \text{ when } \eta^TP^{-1}\eta \geq 1, \quad \text{(A.15)}$$

which implies

$$\eta^T(A^TP^{-1} + P^{-1}A)\eta + 2\eta^TP^{-1}(Bu_f + E\pi + Ew) \leq 0 \text{ when } \eta^TP^{-1}\eta \geq 1, \quad \text{(A.16)}$$

and thus $\dot{V}(\eta) \leq 0$ when $\eta^TP^{-1}\eta \geq 1$. Thus, $\mathbf{X}_f = \{\eta : \eta^TP^{-1}\eta \leq 1\}$ is an invariant set for η [2]. Additionally, inequality (3.22) ensures $\mathbf{X}_f \subseteq \mathcal{X}_f$ (with $\mathcal{X}_f$ defined as in Condition 3.6) [8].

Establish Bounded Feedback $u_f \in \mathbf{U}_f$ when $\eta \in \mathbf{X}_f$:

Pre- and post-multiplying the first LMI in (3.20) by $diag(P^{-1}, I)$ and utilizing a Schur complement gives $P^{-1} - K_f^T\Pi_fK_f \geq 0$, which is equivalent to

$$u_f^T\Pi_fu_f \leq \eta^TP^{-1}\eta \quad \text{(A.17)}$$

after pre- and post-multiplying with η^T and η, respectively, and utilizing the feedback policy

$u_f = K_f \eta$ from (3.26). Thus, for $\eta \in \mathbf{X}_f$, $u_f^T \Pi_f u_f \leq 1$, so $u_f \in \mathbf{U}_f$ from Condition 3.6.

Thus, Corollary 3.1.1 establishes a framework and feedback policy satisfying Condition 3.1 for the *error* dynamics in system (3.15).

Establish $\mathbf{X}_f$ Contained in $\mathbf{\Omega}_o$:

Inequality (3.23) implies $P^{-1} \geq \frac{1}{c_2} I > \frac{1}{c_1} I \geq Q^{-1}$. Then, for $x \in \mathbf{X}_f$,

$$1 \geq x^T P^{-1} x \geq \tfrac{1}{c_2} x^T x > \tfrac{1}{c_1} x^T x \geq x^T Q^{-1} x. \tag{A.18}$$

Thus, an $x \in \mathbf{X}_f$ is also $x \in \mathbf{B}_r$, $x \in \mathbf{B}_R$, and $x \in \mathbf{\Omega}_o$ with $r = \sqrt{c_2}$ and $R = \sqrt{c_1}$ for $\mathbf{B}_r$ and $\mathbf{B}_R$, respectively. The inequality establishes Condition 3.3 such that $\mathbf{X}_f \subset \mathbf{\Omega}_o$. □

Appendix B

Relevant Proofs for Discrete-Time Algorithms

B.1 Proof of Theorem 5.1: Feedback Policy for Discrete SR-MPC

The following Lemma is useful in the proof of Theorem 5.1.

Lemma B.1. *Consider the discrete time system (5.22) satisfying Condition 5.4 and Condition 5.5. Suppose that there exists some $P = P^T > 0$ and associated Lyapunov function $V_k = \eta_k^T P^{-1} \eta_k$ such that*

$$V_{k+1} - V_k + \lambda \left(V_k - \frac{w_k^T w_k}{\sigma^2} \right) + \beta(\gamma^2 \eta_k^T C_q^T C_q \eta_k - \pi_k^T \pi_k) \leq 0\,, \quad \forall\, \eta_k,\ \forall\, w_k,\ \forall\, \pi_k, \quad \text{(B.1)}$$

with some $\lambda \in (0\,,\ 1)$, $\sigma > 0$, and $\beta > 0$. Then $\mathbf{X}_f = \{\eta\ :\ \eta^T P^{-1} \eta \leq 1\}$ is an invariant set for (5.22), i.e., if $\eta_0 \in \mathbf{X}_f$, then $\eta_k \in \mathbf{X}_f, \forall k \in \mathbb{Z}^+$. ◇

Proof. Since $\mathcal{G}$ is a closed and convex set (Condition 5.4), and making use of Lemma 3.2, then for any π there exists η and $G \in \mathcal{G}$ such that

$$\pi_k = G C_q \eta_k.$$

This implies that

$$\pi_k^T \pi_k = \eta_k^T C_q^T G^T G C_q \eta_k \leq \gamma^2 \eta_k^T C_q^T C_q \eta_k$$

by use of (5.24). Set $\mathbf{X}_f$ is an invariant set for (5.22) if

$$V_{k+1} \leq 1, \quad \forall V_k \leq 1, \forall w_k^T w_k \leq \sigma^2, \text{ and } \forall \gamma^2 \eta_k^T C_q^T C_q \eta_k \geq \pi_k^T \pi_k. \tag{B.2}$$

Note, Condition 5.5 implies $w_k^T w_k \leq \sigma^2, \forall k$. Making use of the S-procedure [8], the existence of positive scalars c_1, c_2, and c_3 satisfying the following inequality is sufficient for the satisfaction of the inequalities in (B.2):

$$V_{k+1} - 1 + c_1(1 - V_k) + c_2(\sigma^2 - w_k^T w_k) + c_3(\gamma^2 \eta_k^T C_q^T C_q \eta_k - \pi_k^T \pi_k) \leq 0, \quad \forall \eta_k, \forall w_k, \forall \pi_k.$$

This inequality can be rewritten as

$$V_{k+1} - c_1 V_k - c_2 w_k^T w_k - (1 - c_1 - c_2\sigma^2) + c_3(\gamma^2 \eta_k^T C_q^T C_q \eta_k - \pi_k^T \pi_k) \leq 0\,, \quad \forall\, \eta_k,\ \forall\, w_k,\ \forall\, \pi_k\,. \tag{B.3}$$

Similarly, inequality (B.1) can be rewritten as

$$V_{k+1} - (1 - \lambda) V_k - \frac{\lambda}{\sigma^2} w_k^T w_k + \beta(\gamma^2 \eta_k C_q^T C_q \eta_k - \pi_k^T \pi_k) \leq 0\,, \quad \forall\, \eta_k,\ \forall\, w_k,\ \forall\, \pi_k\,. \tag{B.4}$$

Since $\lambda \in (0, 1)$, letting $c_1 = 1 - \lambda$, $c_2 = \frac{\lambda}{\sigma^2}$, and $c_3 = \beta$ provides positive scalars that establish the equivalence between inequalities (B.3) and (B.4), which completes the proof. □

Proof of Theorem 5.1

Proof.

Establish $\mathbf{X}_f$ Invariance for System (5.22) when $\eta_k \in \mathbf{X}_f$:

Let positive-definite function $V_k = \eta_k^T P^{-1} \eta_k$ be a Lyapunov function candidate. Pre-

and post-multiply (5.27) by $diag\left(\left[P^{-1}, \frac{1}{\alpha\gamma}I, \frac{1}{\sigma}I, I, I\right]\right)$ and let $K_f = LP^{-1}$:

$$\begin{bmatrix} -(1-\lambda)P^{-1} & 0 & 0 & A_d^T + K_f^T B_d^T & C_q^T \\ 0 & -\frac{1}{\alpha\gamma^2}I & 0 & E_d^T & 0 \\ 0 & 0 & -\frac{\lambda}{\sigma^2}I & E_d^T & 0 \\ A_d + B_d K_f & E_d & E_d & -P & 0 \\ C_q & 0 & 0 & 0 & -\alpha I \end{bmatrix} \leq 0. \tag{B.5}$$

Utilizing two Schur complements [8] on the above inequality, and letting $\beta = \frac{1}{\alpha\gamma^2}$, gives

$$\begin{bmatrix} -(1-\lambda)P^{-1} + \beta\gamma^2 C_q^T C_q & 0 & 0 \\ 0 & -\beta I & 0 \\ 0 & 0 & -\frac{\lambda}{\sigma^2}I \end{bmatrix} + \begin{bmatrix} A_d^T + K_f^T B_d^T \\ E_d^T \\ E_d^T \end{bmatrix} P^{-1} \begin{bmatrix} A_d + B_d K_f & E_d & E_d \end{bmatrix} \leq 0. \tag{B.6}$$

Pre- and post-multiply this inequality by ζ^T and ζ, respectively, where $\zeta = \left(\eta_k^T, \pi_k^T, w_k^T\right)^T$:

$$\begin{aligned} &-(1-\lambda)V_k - \frac{\lambda}{\sigma^2} w_k^T w_k + \beta(\gamma^2 \eta_k^T C_q^T C_q \eta_k - \pi_k^T \pi_k) \\ &\quad + (\eta_k^T A_d^T + u_{f,k}^T B_d^T + \pi_k^T E_d^T + w_k^T E_d^T) P^{-1} (A_d \eta_k + B_d u_{f,k} + E_d \pi_k + E_d w_k) \leq 0, \end{aligned} \tag{B.7}$$

where $u_{f,k} = K_f \eta_k$ and $V_k = \eta_k^T P^{-1} \eta_k$ are utilized. From the relationship for η_{k+1} in (5.22) and $V_{k+1} = \eta_{k+1}^T P^{-1} \eta_{k+1}$, the above inequality is equivalent to (B.1). Thus, by Lemma B.1, V_k is a Lyapunov function and $\mathbf{X}_f = \{\eta : \eta^T P^{-1} \eta \leq 1\}$ is an invariant set for (5.22).

Establish $\mathbf{X}_f$ Contained in Geometric Constraints of (5.11):

The ellipsoid $\mathbf{X}_f$ is contained in the polytope described by (5.12), if and only if (5.28) holds [8]. Further, $\mathbf{X}_f$ is also contained within the region described by (5.13): pre- and post-multiply (5.29) by $diag\left(\left[P^{-1}, I\right]\right)$, use a Schur complement, and pre- and post-multiply by η_k^T and η_k, respectively, to obtain

$$\eta_k^T Y_j \eta_k \leq \eta_k^T P^{-1} \eta_k, \tag{B.8}$$

which implies $\mathbf{X}_f \subseteq \Lambda_j$ since $\mathbf{X}_f = \{\eta : \eta^T P^{-1} \eta \leq 1\}$. Thus, satisfaction of inequalities (5.28) and (5.29) ensures satisfaction of (5.11) in Definition 5.1.

Establish Bounded Feedback $u_{f,k}$ when $\eta_k \in \mathbf{X}_f$:

Pre- and post-multiply (5.30) by $diag\left(\left[P^{-1}, I\right]\right)$, use a Schur complement, and pre- and post-multiply by η_k^T and η_k, respectively, to obtain

$$u_{f,k}^T u_{f,k} \leq V_{max}^2 \eta_k^T P^{-1} \eta_k, \tag{B.9}$$

where $u_{f,k} = K_f \eta_k$ with $K_f = LP^{-1}$. This inequality implies $||u_{f,k}|| \leq V_{max}$, and thus $u_{f,k} \in \mathbf{U}_f$, when $\eta_k \in \mathbf{X}_f$.

□

B.2 Proof of Theorem 5.2: Safety-Mode Feedforward Policy for Discrete SR-MPC

The following Lemma is useful in the proof of Theorem 5.2.

Lemma B.2. *Consider the discrete time system (5.39) satisfying Condition 5.7. Suppose that there exists some $S = S^T > 0$ and associated Lyapunov function[21] $V_l = \tilde{z}_l^T S^{-1} \tilde{z}_l$ such that*

$$V_{l+1} - V_l + \lambda \left(V_l - \frac{\bar{g}_l^T \bar{g}_l}{\delta^2} \right) \leq 0\,, \quad \forall\, \bar{g}_l, \tag{B.10}$$

with some $\lambda \in (0\,,\ 1)$ and $\delta > 0$. Then $\mathbf{Z}_s = \{\tilde{z} : \tilde{z}^T S^{-1} \tilde{z} \leq 1\}$ is an invariant set for (5.39), i.e., if $\tilde{z}_0 \in \mathbf{Z}_s$, then $\tilde{z}_l \in \mathbf{Z}_s, \forall l \in \mathbb{Z}^+$. ◇

Proof. Lemma B.1 is more general and contains this Lemma as a special case. The proof follows similarly and makes use of Condition 5.7, which implies $\bar{g}_l^T \bar{g}_l \leq \delta^2, \forall l$. □

Proof of Theorem 5.2

Proof.

Establish $\mathbf{Z}_s$ Invariance for System (5.39) when $\tilde{z}_l \in \mathbf{Z}_s$:

Let positive-definite function $V_l = \tilde{z}_l^T S^{-1} \tilde{z}_l$ be a Lyapunov function candidate. Pre- and

post-multiply (5.41) by $diag\left(\left[S^{-1}, \frac{1}{\delta}I, I\right]\right)$ and let $K_s = RS^{-1}$:

$$\begin{bmatrix} -(1-\lambda)S^{-1} & 0 & A_d^T + K_s^T B_d^T \\ 0 & -\frac{\lambda}{\delta^2}I & E_d^T \\ A_d + B_d K_s & E_d & -S \end{bmatrix} \leq 0. \tag{B.11}$$

Utilizing a Schur complement [8] on the above inequality gives

$$\begin{bmatrix} -(1-\lambda)S^{-1} & 0 \\ 0 & -\frac{\lambda}{\delta^2}I \end{bmatrix} + \begin{bmatrix} A_d^T + K_s^T B_d^T \\ E_d^T \end{bmatrix} S^{-1} \begin{bmatrix} A_d + B_d K_s & E_d \end{bmatrix} \leq 0. \tag{B.12}$$

Pre- and post-multiply this inequality by ζ^T and ζ, respectively, where $\zeta = \left(\tilde{z}_l^T, \bar{g}_l^T\right)^T$:

$$-(1-\lambda)V_l - \frac{\lambda}{\delta^2}\bar{g}_l^T\bar{g}_l + (\tilde{z}_l^T A_d^T + u_{os,l}^T B_d^T + \bar{g}_l^T E_d^T)S^{-1}(A_d\tilde{z}_l + B_d u_{os,l} + E_d\bar{g}_l) \leq 0, \tag{B.13}$$

where $u_{os,l} = K_s\tilde{z}_l$ and $V_l = \tilde{z}_l^T S^{-1}\tilde{z}_l$ are utilized. From the relationship for $\tilde{z}_{l+1}$ in (5.39) and $V_{l+1} = \tilde{z}_{l+1}^T S^{-1}\tilde{z}_{l+1}$, the above inequality is equivalent to (B.10). Thus, by Lemma B.2, V_l is a Lyapunov function and $\mathbf{Z}_s = \{\tilde{z} : \tilde{z}^T S^{-1}\tilde{z} \leq 1\}$ is an invariant set for (5.39).

Establish $\mathbf{Z}_s$ Contained in Geometric Constraints of (5.14):

The ellipsoid $\mathbf{Z}_s$ is contained in the polytope described by (5.15), if and only if (5.42) holds [8]. Further, $\mathbf{Z}_s$ is also contained within the region described by (5.16): pre- and post-multiply (5.43) by $diag\left(\left[S^{-1}, I\right]\right)$, use a Schur complement, and pre- and post-multiply by $\tilde{z}_l^T$ and $\tilde{z}_l$, respectively, to obtain

$$\tilde{z}_l^T C_s^T \Pi_j C_s \tilde{z}_l \leq \tilde{z}_l^T S^{-1}\tilde{z}_l, \tag{B.14}$$

which implies $\mathbf{Z}_s \subseteq \Upsilon_j$ since $\mathbf{Z}_s = \{\tilde{z} : \tilde{z}^T S^{-1}\tilde{z} \leq 1\}$. Thus, satisfaction of inequalities (5.42) and (5.43) ensures satisfaction of (5.14) in Definition 5.2.

Establish Bounded Feedforward $u_{os,l}$ when $\tilde{z}_l \in \mathbf{Z}_s$:

Pre- and post-multiply (5.44) by $diag\left(\left[S^{-1}, I\right]\right)$, use a Schur complement, and pre- and

post-multiply by $\tilde{z}_l^T$ and $\tilde{z}_l$, respectively, to obtain

$$u_{os,l}^T u_{os,l} \leq U_{max}^2 \tilde{z}_l^T S^{-1} \tilde{z}_l, \tag{B.15}$$

where $u_{os,l} = K_s \tilde{z}_l$ with $K_s = RS^{-1}$. This inequality implies $||u_{os,l}|| \leq U_{max}$, and thus $u_{os,l} \in \mathbf{U}_o$, when $\tilde{z}_l \in \mathbf{Z}_s$. □

Bibliography

[1] A.B. Açıkmeşe and J.M. Carson. A nonlinear model predictive control algorithm with proven robustness and resolvability. In *Proceedings of the American Control Conference*, pages 887–893, June 2006.

[2] A.B. Açıkmeşe and M. Corless. Robust tracking and disturbance rejection of bounded rate signals for uncertain/non-linear systems. *International Journal of Control*, 76(11):1129–1141, 2003.

[3] B. Açıkmeşe and S.R. Ploen. Convex programming approach to powered descent guidance for Mars landing. *Journal of Guidance, Control, and Dynamics*, 30(5):1353–1366, 2007.

[4] B. Açıkmeşe, D.P. Scharf, E.A. Murray, and F.Y. Hadaegh. A convex guidance algorithm for formation reconfiguration. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2006. AIAA-2006-6070.

[5] R.G. Bartle. *The Elements of Real Analysis*. Wiley, 1964.

[6] A. Bemporad, F. Borelli, and M. Morari. Model predictive control based on linear programming — the explicit solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985, 2002.

[7] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[8] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM, 1994.

[9] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[10] J.M. Carson and B. Açıkmeşe. A model predictive control technique with guaranteed resolvability and required thruster silent times for small-body proximity operations. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2006. AIAA-2006-6780.

[11] J.M. Carson, B. Açıkmeşe, R.M. Murray, and D.G. MacMynowski. A robust model predictive control algorithm with a reactive safety mode. In *Proceedings of the International Federation of Automatic Control Conference*, July 2008.

[12] C.C. Chen and L. Shaw. On receding horizon feedback control. *Automatica*, 18(3):349–352, 1982.

[13] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

[14] H. Chen, C. Scherer, and F. Allgöwer. A game theoretic approach to nonlinear robust receding horizon control of constrained systems. In *Proceedings of the American Control Conference*, 1997.

[15] F. Fahroo and I.M. Ross. Direct trajectory optimization by a Chebyshev pseudospectral method. *Journal of Guidance, Control, and Dynamics*, 25(1):160–166, 2002.

[16] R. A. Freeman and J. A. Primbs. Control Lyapunov functions: New ideas from an old source. In *Proceedings of the 35th IEEE Conference on Decision and Control*, pages 3926–3931, December 1996.

[17] C.E. Garcia and M. Morari. Model predictive control: theory and practice — a survey. *Automatica*, 25(3):335–348, 1989.

[18] A. Jadbabaie. *Receding Horizon Control of Nonlinear Systems: A Control Lyapunov Function Approach*. PhD thesis, California Institute of Technology, 2000.

[19] A.A. Jalali and V. Nadimi. A survey on robust model predictive control from 1999-2006. In *Proceedings of the International Conference on CIMCA-IAWTIC*, pages 207–212, November 2006.

[20] S.S. Keerthi and E.G. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of Optimization Theory and Applications*, 57(2):265–293, 1988.

[21] H.K. Khalil. *Nonlinear Systems, Third Edition*. Prentice Hall, 2001.

[22] D.E. Kirk. *Optimal Control Theory, An Introduction*. Prentice Hall, 1970.

[23] M.V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10):1361–1379, 1996.

[24] Y. Kuwata, T. Schouwenaars, A. Richards, and J. How. Robust constrained receding horizon control for trajectory planning. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2005. AIAA-2005-6079.

[25] W.H. Kwon, A.M. Bruckstein, and T. Kailath. Stabilizing state-feedback design via the moving horizon method. *International Journal of Control*, 37(3):631–643, 1983.

[26] W.H. Kwon and A.E. Pearson. A modified quadratic cost problem and feedback stabilization of a linear system. *IEEE Transactions on Automatic Control*, 22(5):838–842, October 1977.

[27] W. Langson, I. Chryssochoos, S.V. Raković, and D.Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40(1):125–133, 2004.

[28] M. Magni, H. Nijmeijer, and A. Van Der Schaft. A receding horizon approach to the nonlinear H_∞ control problem. *Automatica*, 37(3):429–435, 2001.

[29] D.Q. Mayne and W. Langson. Robustifying model predictive control of constrained linear systems. *Electronics Letters*, 37(23):1422–1423, 2001.

[30] D.Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, July 1990.

[31] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[32] D.Q. Mayne, M.M. Seron, and S.V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[33] H. Michalska and D.Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993.

[34] M.B. Milam. *Real Time Optimal Trajectory Generation for Constrained Dynamical Systems*. PhD thesis, California Institute of Technology, 2002.

[35] M.B. Milam, R. Franz, and R.M. Murray. Real-time constrained trajectory generation applied to a flight control experiment. *IFAC World Congress*, 2002.

[36] Y. Nesterov and A. Nemirovsky. *Interior-point Polynomial Methods in Convex Programming*. SIAM, 1994.

[37] J.A. Primbs. *Nonlinear Optimal Control: A Receding Horizon Approach*. PhD thesis, California Institute of Technology, 1999.

[38] S.J. Qin and T.A. Badgwell. An overview of industrial model predictive control technology. In *Chemical Process Control — V, AIChe Symposium Series*, volume 93, number 316, pages 232–256. CACHE, 1997.

[39] S.V. Raković, A.R.Teel, D.Q. Mayne, and A. Astolfi. Simple robust control invariant tubes for some classes of nonlinear discrete time systems. *Proceedings of the 45^{th} IEEE Conference on Decision & Control*, pages 6397–6402, 2006.

[40] S.V. Raković and D.Q. Mayne. A simple tube controller for efficient robust model predictive control of constrained linear discrete time systems subject to bounded disturbances. *Proceedings of the 16^{th} IFAC World Congress*, July 2005.

[41] J.B. Rawlings and K.R. Muske. The stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38(10):1512–1516, 1993.

[42] J. Richalet. Industrial applications of model based predictive control. *Automatica*, 29(5):1251–1274, 1993.

[43] T. Schouwenaars, J. How, and E. Feron. Receding horizon path planning with implicit safety guarantees. In *Proceedings of the American Control Conference*, pages 5576–5581, June 2004.

[44] P.O.M. Scokaert and D.Q. Mayne. Min-max feedback model predictive control for constrained linear sytems. *IEEE Transactions on Automatic Control*, 43(8):1136–1142, 1998.

[45] R.S. Smith. Robust model predictive control of constrained linear systems. In *Proceedings of the American Control Conference*, pages 245–250, June 2004.

[46] E. D. Sontag. A Lyapunov-like characterization of asymptotic controllability. *SIAM Journal on Control and Optimization*, 21(3):462–471, 1983.

[47] E. D. Sontag. A 'universal' construction of Artstein's theorem on nonlinear stabilization. *Systems & Control Letters*, 13(2):117–123, 1989.

[48] J.F. Sturm. Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11(1):625–653, 1999.

[49] Y.A. Thomas. Linear quadratic optimal estimation and control with receding horizon. *Electronics Letters*, 11(1):19–21, January 1975.

[50] K.C. Toh, M.J. Todd, and R.H. Tutuncu. SDPT3 — a Matlab software package for semidefinite programming. *Optimization Methods and Software*, 11(1):545–581, 1999.

[51] J. Vlassenbroeck and R.V. Dooren. A Chebyshev technique for solving nonlinear optimal control problems. *IEEE Transactions on Automatic Control*, 33(4):333–340, 1988.

[52] D.K. Yeomans, P.G. Antreasian, J.-P. Barriot, S.R. Chesley, D.W. Dunham, R.W. Farquhar, J.D. Giorgini, C.E. Helfrich, A.S. Konopliv, J.V. McAdams, J.K. Miller, W.M. Owen Jr., D.J. Scheeres, P.C. Thomas, J. Veverka, and B.G. Williams. Radio science results during the NEAR-Shoemaker spacecraft rendezvous with Eros. *Science*, 289:2085–2088, September 2000.