Optimal Control of Controllable Switched Systems

by

Michael David Rinehart

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2005

Author .....
Department of Electrical Engineering and Computer Science
May 6, 2005

Certified by .....
Munther A. Dahleh
Professor of Electrical Engineering Computer Sciences
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Optimal Control of Controllable Switched Systems

by

Michael David Rinehart

Submitted to the Department of Electrical Engineering and Computer Science
on May 6, 2005, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

Many of the existing techniques for controlling switched systems either require the solution to a complex optimization problem or significant sacrifices to either stability or performance to offer practical controllers. In [13], it is shown that stabilizing, practical controllers with meaningful performance guarantees can be constructed for a specific class of hybrid systems by parameterizing the controller actions by a finite set. We extend this approach to the control of controllable switched systems by constraining the switching portion of the control input and fixing the feedback controller for each subsystem. We show that, under reasonable assumptions, the resulting system is guaranteed to converge to the target while providing meaningful performance. We apply our approach to the direct-injection stratified charge (DISC) engine and compare the results to that of a model predictive controller designed for the same application.

Thesis Supervisor: Munther A. Dahleh
Title: Professor of Electrical Engineering Computer Sciences

Acknowledgments

The author would like to acknowledge Dr. Illya Kolmonovsky and Dr. Davor Hrovat of Ford Research Laboratory for their collaboration on this work. This work was supported in part by Ford Research Laboratory, Ford Motor Company, Dearborn, MI 48124.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we present a formal overview of switched systems and discuss results related to the stability of such systems.

1.1 Switched Systems

Switched systems are a subclass of hybrid systems consisting of an interleaving of continuous and discrete dynamics. The discrete portion of the system is either controlled directly by a controller or a controllable event (a state intersecting some manifold, for instance), or influenced by some uncontrollable external trigger. For the purposes of this paper, we consider the first case only. A *controllable switched system* (which, for brevity, we also call a *switched system*) may be described formally in the following form

$$\begin{aligned} \dot{x}(t) &= f_{i(t)}\left(x(t), u(t)\right), t \neq t_l \\ x(t_l) &= x(t_l^-) + T_{i(t_l), i(t_l^-)} \end{aligned} \tag{1.1}$$

where $i(t) \in M = \{1, 2, \cdots, N\}$ is the *mode* of the system at time t, $x \in X_p \subset \Re^n$ is the system's state in mode p, and $u \in U_p \subset \Re^m$ is the *continuous portion* of the control input in mode p. f_p is a globally Lipshitz function, and $i(t)$ is a piecewise-constant function. The system given by $\dot{x}(t) = f_p(x(t), u(t)$ where $x \in X_p$ and $U \in U_p$ is called the p^{th} subsystem of (1.1).

We define a *switch* as a change in the value of the function $i(t)$ at some time

t ($i(t) \neq i(t^-)$), and denote the l^{th} switching instance as t_l. The *mode sequence* $(p_1, p_2, \cdots)$ induced by $i(t)$ is given by $p_l = i(t_l)$.

Notable in (1.1) is the lack of restrictions the possible switching actions of the controller (i.e. allowing for infinite switching within a finite period). Rather, the controller is formulated to prevent such undesirable behavior.

As reflected in (1.1), the state may "jump" according to the *translation vector* T_{pq} at each switching instance. For our purposes, we treat the translation vector as a means for translating the state about some equilibrium operating point, which is useful, for example, when approximating nonlinear subsystems with linear subsystems. To that end, we assume that for each subsystem of (1.1) there exists a fixed vector x_p^0 such that the *unnormalized* state $\overline{x}(t) = x(t) + x_{i(t)}^0$ is a continuous function of time. The offset vector x_p^0 is an artifact of modeling and is not chosen arbitrarily by the designer to "ensure" the continuity of $\overline{x}(t)$.

If $X_p = \Re^n$ and $U_p = \Re^m$, we call the p^{th} subsystem of (1.1) an *unconstrained* subsystem. Otherwise, if each is a compact set, we call it a *constrained* subsystem. We describe a switched system as being constrained (unconstrained) if all of the subsystems are constrained (unconstrained). Otherwise, we term it as *mixed*. If f_p is a linear function of x and u, we write it as $A_p x + B_p u$.

1.1.1 Stability of Switched Systems

We adopt the classical notion of stability in the sense of Lyapunov to switched systems: for any $\epsilon > 0$, there exists a $\delta > 0$ such that $\|\overline{x}(0)\| < \delta$ implies $\|\overline{x}(t)\| < \epsilon$ for all $t > 0$.

One approach to guaranteeing the stability of a switched system under arbitrary switching is by constructing a *common* Lyapunov function V for all modes of the system. However, this technique may be unnecessarily restrictive in that a system which maybe stabilizable under a certain class of switching laws may not be stable under all switching laws, hence not allowing for the existence of a common Lyapunov function.

Alternatively, one may construct a collection of Lyapunov functions V_p, each cor-

responding to a different operating mode of the system [8]. Although the existence of multiple Lyapunov functions does not guarantee stability under arbitrary switching sequences, there may exist switching sequences that will guarantee stability.

Let $\hat{t}_l^p$ be the l^{th} switching instance of (1.1) into mode p. If, under all allowable switching laws $i(t)$, the sequence $\left(V(\hat{t}_l^p)\right)_{l\in Z^+}$ is non-increasing, then $i(t)$ stabilizes is stable in the sense of Lyapunov. If the sequence is monotonically decreasing, then the system is asymptotically stable. Practically, one may use this condition to design a feedback control law to stabilize the system. One such approach is described in the next chapter.

1.1.2 Optimal Control of Switched Systems

The added flexibility of being able to switch between subsystems greatly increases the complexity of searching for an optimal control. Inherent in computing the optimal control law is a combinatorial problem that requires one to simultaneously determine a (potentially infinite) mode sequence as well as the switching instances and the continuous control input that controls each subsystem.

Determining the optimal control law, therefore, is not a practical endeavor, and techniques for optimally controlling such systems often try to reduce the complexity by either restricting the number of controller decisions or considering a subclass of the original optimal problem (for example, by fixing the mode sequence and optimizing over the switching instances and the continuous control input only). These techniques will be explored further in the following chapters.

1.2 Overview of this Paper

In this thesis, we analyze some existing approaches to optimally controlling switched systems and present two new approaches that overcomes several inherent defects in the existing art. We compare one such approach to another hybrid controller in an application to the DISC engine, a switched system where stability and performance are absolute requirements and where the computational and memory requirements of

the controller must be applicable to low-cost embedded hardware.

For the remainder of this paper, we denote $\|v\|_1$, $\|v\|_2$, and $\|v\|_\infty$ as the standard 1, 2, and ∞-norms whereas we denote $\|v\|_Q$ as $v'Qv$ for a symmetric, positive-definite matrix Q.

Chapter 2

Existing Approaches to Optimally Controlling Switched Systems

In this chapter, we review several existing approaches for optimally controlling both constrained and unconstrained switched systems. Although these techniques vary significantly, each either requires the solution to complex optimization problems or a sacrifice in either stability or performance to obtain a potentially implementable solution.

It should be noted that although these approaches do not treat the impact of the translation vector, and so, for the remainder of this chapter, assume $T_{pq} = 0$ in (1.1). Note that this does not imply that their formulations cannot be extended to incorporate the translation vector.

2.1 Dynamic Programming on Unconstrained Subsystems

A straight-forward application of dynamic programming to switched system control may be found in [19], where a combination of numerical optimizations is used to derive an suboptimal pair of continuous control inputs and switching instances. The mode sequence is assumed to be known so that it is not necessary to search over all

finite-length mode sequences as well.

The optimization problem posed in [19] is as follows: given a switched system of the form (1.1) and a finite mode sequence $(p_1, p_2, \cdots, p_n)$, determine the continuous input $u(t)$ and switching instances t_l so that

$$J(x,u) = \psi\left(x(t_f)\right) + \int_{t_0}^{t_f} L(x,u)dt \tag{2.1}$$

is minimized.

The authors propose adding a "false" state z_k for each of the k switching instants of the system and reparameterizing the time variable so as to reformulate the optimal control problem as fixed-switching instance, free-final state, and free-initial condition (in the false states) dynamic programming problem. Determining the optimal solution requires, for fixed values of z_k, the solutions to the standard state and co-state equations that give the continuous control input, and another set of state of equations for determining the derivative of the optimal state and input trajectories with respect to z_k. A search over z_k is then applied to find where the cost J is locally minimized.

Additional constraints on the optimization, including a restriction on the switching regions and a free-final state condition, maybe applied to this approach in a fairly straight-forward fashion.

The complexity of this approach scales both with the number of states in the original system (which determines the complexity of the numerical solutions to the state and co-state equations) and the number of switches (which governs the size of the search for minimizing J).

Though the use of dynamic programming in this fashion is guaranteed to produce a suboptimal, convergent trajectory, it is not practical to apply the approach to either fast-paced systems or systems with computational limitations. The solution requires iterations between updating the initial conditions for the false states and computing the numerical solution to a series of two-point boundary value differential algebraic equations (DAEs). For example, in a simple application to a second order system with three subsystems and two switches, the numerical optimization required

approximately 260 seconds on a high-end workstation.

Though one may consider storing the switching instances in memory (by gridding a large volume of the state and reference spaces, for example) and computing online the solution to the DAEs that give the input trajectory, the assumption that the mode sequence is fixed in the problem formulation implies that the memory requirements grow with the number of possible finite-length sequences that are considered. Furthermore, the computation of the DAE for the input trajectory may itself be too computationally intensive to apply on low-cost embedded hardware for a fast-paced system.

2.2 Dynamic Programming via Discretization and Optimal Cost Bounds

Another application of dynamic programming to the control of switched systems is presented in [15], where the authors seek to construct a function $V_p : X_p \to \Re$ for each mode p of the system so that V_p satisfies a particular set of conditions that guarantees the value of $V_{p_0}(x_0)$ acts the lower bound for the optimal cost-to-go from an initial mode and state (p_0, x_0) to a reference mode and state (p_f, x_f).

For a given optimal cost function

$$J(x_0, p_0) = \int_{t_0}^{t_f} L_{i(t)}(x, u)dt + \sum_{k=1}^{M} s\left(x(t_k), i(t_k^-), i(t_k^+)\right) \tag{2.2}$$

where M is an arbitrary integer representing the the number of switches and s is penalty of switching, the authors propose constructing the value functions V_p by discretizing the state space and explicitly computing a value for $V_p(x)$ for a set of points x in the space X_p. Of course, it is assumed that the state spaces for each of the subsystems is constrained so that the search is finite. To ensure the approximation is truly a lower bound, the search additionally accounts for the dynamics of the system between each grid point of the discretiztion.

Assuming that each V_p represents the true optimal cost-to-go, the optimal contin-

uous and discrete control laws can be computed as

$$\begin{aligned} u_p(x) &= \arg\min_{\hat{u}\in U_p} \frac{\partial V_p}{\partial x} f_p(x,\hat{u}) + L_p(x,\hat{u}) \\ i_p(x) &= \arg\min_{q} V_q(x) + s(x,p,q) \end{aligned} \tag{2.3}$$

for each approximation point x in each mode p. Essentially, u at a point x is chosen so that the optimal cost-to-go is achieved while the choice of i at x is designed to seek the minimum cost-to-go from the fixed point x across all operating modes. In this setting, the target reference is fixed.

In reality, of course, the V's are only guaranteed to act as lower-bounds for the cost-to-go. Stability and convergence are not discussed in the work, and it is not clear how fine a quantization of the space is required to guarantee convergence. For example, for a set of the trivial value functions $V_p = 0$ (which is clearly a lower-bound for the cost-to-go) for all modes p, convergence to the target is not guaranteed. Since a coarser gridding of the state space results in a poorer lower bound of the optimal cost-to-go (smaller V_p's), one should suspect that the coarseness of the state space gridding should impact convergence.

2.3 Model Predictive Control (MPC) for Constrained Linear Subsystems

In [2], it is shown how to extend the techniques of MPC (also known as receding horizon control) to arbitrary systems with dynamics that can be expressed as a combination of linear and logical dynamics. The authors show that any such system can be converted into what they term a *mixed logical dynamical* (MLD) system, and they apply a variation of MPC control to it using mixed-integer quadratic programming (MIQP), the resulting formulation called mixed-integer predictive control (MIPC). If the continuous parameters of the MIQP are bounded to polytopes, the solution to the MIPC maybe stored exactly in memory and referenced using state feedback. Of course, this application of MPC, like all applications of MPC, is limited to discrete-

time systems.

One variation of the MIQP problem formulation is as follows: given an initial state x_0 and a final state x_f, find the control sequence $u(t)$ and switching sequence $i(t)$, $0 \leq t \leq T < \infty$ so as to minimize

$$J = \sum_{t=0}^{T-1} \|x(t|0) - x_f\|_Q^2 + \|u(t) - u_e\|_R^2 + \|i(t) - i_e\|_K^2 \tag{2.4}$$

subject to $x(t|0)$ and $u(t)$ being admissable parameters, and $x(t+1|0) = A_{i(t)}x(t|0) + B_{i(t)}u(t)$ where $x(t|0)$ is the predicted value of x at time t given $x(0|0) = x_0$.

MIPC does not alleviate the issue of combinatorial complexity in choosing a mode sequence, and, at each time step of the look-ahead horizon, it must consider all possible mode sequences. An "efficient" (in the sense that it does not, on average, consider all possibilities) branch and bound algorithm for reducing expected the computational time required to compute the solution to (2.4) is given in [1]. To make the approach applicable to embedded hardware, it is assumed that the state and input spaces are polytopes so that the solution may be stored exactly in memory and simply referenced online.

In essence, MIPC attempts to reduce the complexity of hybrid control by examining only a short look-ahead horizon at each time step of the controller. One proof of the "stability" of MIPC requires that a feasible solution to the MIPC exists over the full horizon from the initial state x_0 to the terminal state x_f [2]. Of course, it is not practical to consider the "feasibility" argument for stability since the computational advantages to MIPC (and MPC in general) lie in considering a short horizon that, for many applied systems, will not include the target state.

Another method for guaranteeing the stability of an MIPC is to append (2.4) with a terminal weight of the form $x'Px$ for a symmetric, positive definite matrix P satsifying a particular set of conditions that guarantee that the cost-to-go is a Lyapunov function [3]. The existence of such a P matrix, however, is tanamount to the existence a common polyhedral Lyapunov function for the system, which is not, in general, guarenteed to exist. It is additionally remarked that a means for

synthesizing P is still an open-problem for hybrid systems.

2.4 Control Lyapunov Functions for Constrained Subsystems

In [11], control Lyapunov functions are applied to general, nonlinear input constrained switched system of the form (1.1). For each subsystem of the switched system, control Lyapunov functions are used to compute a stabilizing control law to the origin that satisfies the actuator constraints. Because it is assumed that the initial state of the system lies in the region of stability for the inital mode, the system is asymptotically stable for each mode of the system. Stability is sufficiently guaranteed under switching by applying a slight variation of the multiple Lyapunov function approach to stability: the value of the Lyapunov function when switching into a mode must be less than the value of the Lyapunov function when that mode was last switched from.

Although the authors leverage a separation in subsystem and switching control which significantly reduces the complexity of the online controller, the motivation to switch is unclear since system performance is not a factor in the problem formulation and stability is already guaranteed when no switching occur. For most applications where performance is a factor in controller design, this technique is simply unsuitable.

2.5 Switching among Autonumous Linear Systems

The work in [14] may be considered a special case of [19] with two caveats: each subsystem is stable and linear and there is no continuous control input

$$\dot{x} = A_{i(t)}x \tag{2.5}$$

(Of course, if each subsystem of a switched system is linear and stabilizable, a state-feedback law of the form $u = -K_p x$ for each mode p may be used to stablize the system so that resulting system is of the form (2.5)).

Leveraging the homogeneity of the switched system, the authors expose a pattern in the optimal switching law whereby, for a given mode and remaining mode sequence, the switching law may be expressed in a form of state-feedback where the decision to switch is made by examining the cone the state lies within. It can easily be seen that for the class of switched systems where each subsystem is homogenous and the cost function is of the form

$$J = \int_0^T \|x\|_Q^z + \|u\|_R^z dt \tag{2.6}$$

that for two switching laws $i_1(t)$ and $i_2(t)$

$$\begin{gathered} J_{i_1(t)}(x_0) < J_{i_2(t)}(x_0) \\ \Rightarrow J_{i_1(t)}(\alpha x_0) = |\alpha|^z J_{i_1(t)}(x_0) < |\alpha|^z J_{i_2(t)}(x_0) = J_{i_2(t)}(\alpha x_0) \end{gathered} \tag{2.7}$$

Indicating that the optimal switching law $i^*(t)$ is the same for all $\hat{x}$ lying in the cone αx in mode $i^*(0)$. Hence, the feedback control law has a conic structure.

The work is further extended to the case of finite-length, arbitrary mode sequences in [6] by applying the principles of dynamic programming to this switching law structure.

The determination of a continuous control input, the switching instances, and the mode sequence are well separated in this approach, and the only online computaional complexity lies in finding the conic region that contains the state. However, restricting the continuous input to linear state-feedback for each mode may be unnecessarily restrictive. Furthermore, as the length of the mode-sequence increases, the memory requirements of the switching law grows proportionally. Infinite-horizon switching laws, even if approximated by large mode sequences, are impractical to consider.

Chapter 3

Reducing the Complexity in Control of Systems with Bounded Switching Regions

The approaches presented in the previous section suffer from either significant-complexity, restrictive assumptions, or a critical sacrifice to stability. It was shown in [13] and [12] that the use of hierarchal control combined with a quantization in the controller decisions to a finite parameterization can greatly reduce the computational complexity of hybrid control, allowing for the use of planning algorithms that guarantee convergence and quality of performance. We seek to extend this approach to general switched systems that do not admit the symmetries with respect to the tracking space that is assumed in these works. In particular, we only assume that the switched system of interest has bounded switching regions, a practical assumption that includes the class of constrained switched systems.

3.1 Constructing and Applying the Static Robust Hybrid Switching Graph

The underlying idea of the Static Robust Hybrid Switching Graph (SRHSG) is as follows: restrict the ability of the system to switch to a finite set of states in each mode, called *switching states*. Applying predesigned feedback controllers in each mode, we can then find the minimum-cost path consisting of a set of switching states from the initial state to the terminal state.

In this chapter, assume that the switching regions $\overline{X}_{pq}$ are bounded between all modes p and q ($p \neq q$) of the switched system. Define the *switching states* $S_{pq} \subset \overline{X}_{pq}$ of modes p and q by

$$x_1, x_2 \in S_{pq} \text{ where } x_1 \neq x_2 \Rightarrow 0 < 2r_s < \|x_1 - x_2\|_2 \tag{3.1}$$

Essentially, the set of switching states represents those states the controller must track in order to switch from one mode to another. Once the system's state is within the switching radius r_s of one of switching states, the system may switch between these modes. The switching radius should be small but not so small that tracking it is unecessarily difficult (from the effects of noise, for example). Clearly, by the assumption that $\overline{X}_{pq}$ is bounded, the set of switching states is finite.

Our requirement that the state $\overline{x}(t)$ be continuous is reinforced by only allowing a transition from two modes to occur where the unnormalized state is admissable in both modes. For simplicity, from this point forward, we will assume without loss of generality that $T_{pq} = 0$ so that $\overline{x}(t) = x(t)$.

For each switching state, we define a *switching radius* as $SR(x) = \{x_r | \|x_r - x\|_2 < r_s\}$. The set of all switching states of mode p is given as $S_p = \bigcup_{q \in M} S_{pq}$.

We now construct the robust hybrid switching graph (SRHSG) for the system.

Define the directed SRHSG $G = (V, E)$ by

$$\begin{aligned} &1.\ v = (p, x) \in V \Leftrightarrow x \in S_p \\ &2.\ (v_1, v_2) = ((p, x_1), (q, x_2)) \in E \Leftrightarrow \\ &\quad \{p = q \text{ and } x_1, x_2 \in S_p\} \text{ or } \{p \neq q \text{ and } x_1 = x_2 \in S_{pq}\} \end{aligned} \tag{3.2}$$

The first condition simply states that all the switching states are vertices in the SRHSG and vice-versa. The second condition states there exist edges between all the switching states in the same mode, and between the same switching state in two different modes.

Assume that for each mode p of the system there exists a control law u_p such that for any initial state x_0, final state x_d, and a reference y_{ref}, there exists a time $T \geq 0$ so that $x(T) = x_d$. Furthermore, assume u_p and T are chosen so as to minimize

$$J_p = \int_0^T L\left((g(x, u) - y_{ref}), u_p\right) dt \tag{3.3}$$

where L is a positive definite function, and $g(x)$ is the output of the system in mode p. We term the optimal cost for such a given set of parameters as $J_p^*(x_0, x_d, y_{ref})$. Also, let $\hat{u}$ be the control law that minimizes (3.3) for $T = \infty$ and denote the corresponding optimal cost as $\hat{J}_p^*(x_0, x_d, y_{ref})$. Design of the subsystem controllers is discussed in section 3.5.

It is possible that a controller that minimizes the error with respect y_{ref} while tracking x_d in finite time may be too computationally intensive to use. As an alternative, one may apply a controller that optimally tracks a different reference, $\hat{y}_{ref}$, that is achievable in steady-state at x_d and optimal with respect to y_{ref}. Of course, the resulting full controller may not be optimal with respect to the traditional metrics, but the optimization may still meaningful and potentially worth the significant reduction in complexity.

By the limitation of only being able to choose from a finite set of switching states, and by having a control law that can track switching states in finite time, it is only necessary to determine the sequence of switching states to track, termed the *switching*

path, from an initial state x_0 in mode p to a reference state x_{ref} (corresponding to the reference output y_{ref}) in mode q.

The SRHSG graph G is appended with additional edges connecting the switching states of the initial and reference modes to, respectively, the initial and reference states. Let $v_0 = (p, x_0)$ and $v_f = (q, x_{ref})$, and define the directed, weighted graph $G' = (V', E')$, called the *appended SRHSG*, by

$$\begin{aligned} &1.\ V' = V \cup \{v_0, v_f\} \\ &2.\ E' = E \cup \{\bigcup_{x \in S_p} (v_0, (p, x))\} \cup \{\bigcup_{x \in S_q} ((q, x), v_f)\} \end{aligned} \tag{3.4}$$

Define the weighting function $w : E' \to \Re$ on G' as:

$$w(e) = \begin{cases} \epsilon_s, & p_1 \neq p_2 \\ J^*_{p_1}(x_1, x_2, y_{ref}), & p_1 = p_2 \text{ and } x_2 \neq x_{ref} \\ \hat{J}^*_{p_1}(x_1, x_2, y_{ref}), & otherwise \end{cases} \tag{3.5}$$

where $\epsilon_s > 0$ is the switching penalty.

Subject to the above constraints, we can now determine the optimal switching path simply by finding the "shortest" path (the path of least cost) from v_0 to v_f in G', an algorithm with $O\left(|E'| \log_2 |V'|\right)$ complexity.

Let the function $SPN : V' \to V'$ be a mapping of a node in the graph G' to the next node in the optimal switching path to v_f, and define $SPN(v_f) = v_f$. The optimal switching path starting from v_0 can be then be written as

$$P = [SPN(v_0), SPN(SPN(v_0)), \cdots] = [p_1, p_2, \cdots] \tag{3.6}$$

and the number of switches is obtained as $N = \min\{i | p_i = p_j, j > i\}$. Since the shortest-path in the appended SRHSG contains no cycles, N is guaranteed to be less than or equal to $|V'|$, meaning there are only a finite number of switches in the switching path.

3.2 Making SRHSG Applicable to High-Speed Processes

It may not be practical to compute the costs between large numbers of vertices or, perhaps, to apply a shortest-path algorithm on every time step of the system even if all of the costs are known, and so a means for storing a rough approximation to the solution of the dynamic program in memory is required. Given the granularity of the SRHSG shortest-path problem, we can assume that small shifts in the initial and reference states do not impact the switching portion (those states before the reference) of the switching path. Therefore, we can quantize the state and reference output spaces and expect a fairly good approximation to the exact solution.

Define the set $\tilde{X}_p$, the set of *approximation points* in the admissible set of the state space in operating mode p, by

$$\begin{aligned} &1.\ S_p \subset \tilde{X}_p \subset X_p \\ &2.\ x_1, x_2 \in \tilde{X}_p \Rightarrow 2r_s < \|x_1 - x_2\|_2 \\ &3.\ \tilde{X}_p \text{ is finite} \end{aligned} \tag{3.7}$$

The first condition states that the switching states of mode p act also as approximation points. The second condition ensures that a switching state is the approximation point for its switching radius, which is necessary for convergence. Define the state approximation function $APX_p : X_p \to \tilde{X}_p$ as $APX_p(x) = \arg\min_{z \in \tilde{X}_p}\{\|x - z\|_X\}$. Similarly define $\tilde{Y}_p$ and APY_p for the reference space (though the lower bound for the distance between such approximation points may be any positive constant).

Given a finite number of approximation points in both spaces, we can now store the solutions in a table. For an initial mode p and reference mode q, define the *SRHSG table* $T_{pq} = \tilde{X}_p \times \tilde{Y}_q \to M \times X$ as

$$\begin{aligned} T_{pq}(\tilde{x}, \tilde{y}) &= SPN\left((p, \tilde{x})\right) \\ (\text{with } v_0 = (p, \tilde{x}) \text{ and } v_f &= (q, \tilde{y})) \end{aligned} \tag{3.8}$$

3.3 The Full Controller

We apply the SRHSG table to the control law u_p for each mode p in an open-loop fashion. Table lookups are only performed when the mode or the reference changes, otherwise the system drives the plant's state to the next state in the switching path determined by the previous lookup. Denote the reference state, output, and mode, respectively, as x_{ref}, y_{ref}, and q, and let $x(n)$, $y(n)$, and $i(n)$ be the state, output, and operating mode of the plant at time sample n. The following algorithm uses the SRHSG table to guide the plant along the optimal switching path

$$
\begin{aligned}
&1.\ \text{if } x(n) \in SR(x) \text{ then } i(n+1) \leftarrow p_{next} \text{ else } i(n+1) \leftarrow i(n) \\
&2.\ \text{if } \textit{either the reference changes} \text{ or } i(n) \neq i(n+1) \text{ then} \\
&3.\quad \tilde{x} \leftarrow APX_{i(n+1)}\left(x(n)\right) \\
&4.\quad (p_{next}, x_d) \leftarrow T_{i(n+1)q}\left(\tilde{x}, APY_q(y_{ref})\right) \\
&5.\quad \text{if } i(n+1) = p_{next} \text{ then } x_d \leftarrow x_{ref} \\
&6.\ \text{end} \\
&7.\ u(n+1) \leftarrow \begin{cases} \hat{u}_{i(n+1)}\left(x(n), y(n), x_d, y_{ref}\right), & x_d = x_{ref} \\ u_{i(n+1)}\left(x(n), y(n), x_d, y_{ref}\right), & otherwise \end{cases}
\end{aligned}
\tag{3.9}
$$

3.4 Stability

By the properties of u_p, the finiteness of the switching portion of the switching path, and the assumption that all switching states are reachable, (3.9) is guaranteed to robustly converge to reference. By induction, it can easly be shown that since each switching point is reached in finite time, the system will eventually reach the final mode of the system where is tracks the reference state and output.

However, regardless of the feedback controllers used for each subsystem, it is not possible to guarantee stability in the sense of Lyapunov. For example, for a switched system containing two subsystems and one switching state, regardless of the intial and final states, the state must pass through the switching state in order to switch modes.

Once in the reference mode, however, the system is guaranteed to be asymptotically stable to the target state.

3.5 Subsystem Controllers

Under the switching constraints presented of the SRHSG, we want to choose the continuous control law $u(t)$ and switching times $\{T_1, T_2, \cdots, T_N\}$ so that, for a given initial state (p, x_0) and final state (q, x_f) and reference output y_{ref}, the following is minimized

$$\begin{aligned} J &= \int_0^{\infty} L\left(g_{i(t)}(x,u) - y_{ref}, u\right) dt \\ &= \int_0^{T_1} L\left(g_{i(t)}(x,u) - y_{ref}, u\right) dt + \int_{T_1}^{T_2} L\left(g_{i(t)}(x,u) - y_{ref}, u\right) dt + \cdots \\ &\quad + \int_{T_N}^{\infty} L\left(g_{i(t)}(x,u) - y_{ref}, u\right) dt \end{aligned} \tag{3.10}$$

where $i(t)$ is the mode at time t and, for simplicity, we assume a particular N-length switching path given by $[(p_1, x_2), \cdots, (p_N, \overline{x}_N), (q, x_f)]$ (i.e., $x(t_k) = x_k$ is a fixed switching point). It can be readily seen from (3.10) that, by fixing the switching state for each mode, we separate the cost function into several independant optimizations

$$\begin{aligned} J = &\int_0^{T_1} L\left(g_{i(t)}(x,u) - y_{ref}, u\right) dt + \int_0^{T_2 - T_1} L\left(g_{i(t)}(x,u) - y_{ref}, u\right) dt + \cdots \\ &+ \int_0^{\infty} L\left(g_{i(t)}(x,u) - y_{ref}, u\right) dt \end{aligned} \tag{3.11}$$

Therefore, for each mode p and any $x(0) = x_0 \in X_p$, $x_d \in X_p$, and $y_{ref} \in Y$, we must choose a final time $T < \infty$ and a continuous control input u so that $x(T) = x_d$ and the cost function

$$J = \int_0^T L(g_p(x,u) - y_{ref}, u) dt \tag{3.12}$$

is minimized.

3.5.1 Computing the Optimal Control Law

In general, minimizing (3.12) over both $u(t)$ and T is a formidable task. In this section, we consider approaches to solving this optimization problem for the special case of f_p being a linear system and L being a quadratic function of the output error and the input. We provide a means for computing the optimal, closed-loop control law for both unconstrained and constrained subsystems.

Unconstrained Systems

We adapt the deriviation of the optimal, finite-time, final-state fixed controller in [9] for linear systems with a quadratic cost function to the following optimization problem: find the continuous control law $u(t)$ or $0 \leq t \leq T$ and the optimal final time $T \geq 0$ such that the cost function

$$J = \int_0^T \|Cx + Du - y_f\|_Q^2 + \|u\|_R^2 dt \tag{3.13}$$

subject to

$$\begin{aligned} \dot{x} &= Ax + Bu \\ x(0) &= x_0 \\ x(T) &= x_d \end{aligned} \tag{3.14}$$

is minimized.

For some fixed final time T (not necessarily optimal), we append to (3.13) the boundary constraints at T

$$\overline{J} = v'x(T) + \int_0^T \|Cx + Du - y_f\|_Q^2 + \|u\|_R^2 dt \tag{3.15}$$

where v is some unknown, constant multiplier. From here, we arrive at the state,

co-state, and stationary equations for the optimization

$$\begin{aligned}
\dot{x} &= Ax + Bu \\
\dot{\lambda} &= -A'\lambda - C'Q[Cx + Du - y_f] \\
u &= -[R + D'QD]^{-1}[D'QCx - D'Qy_f + B'\lambda] = -\Gamma^{-1}[D'QCx - D'Qy_f + B'\lambda]
\end{aligned} \tag{3.16}$$

which, after some simplfication, gives

$$\begin{aligned}
\dot{x} &= A_x x + B_x \lambda + E_x y_f \\
\dot{\lambda} &= A_\lambda \lambda + B_\lambda x + E_\lambda y_f
\end{aligned} \tag{3.17}$$

where

$$\begin{aligned}
A_x &= A - B\Gamma^{-1}D'QC & A_\lambda &= -(A - B\Gamma^{-1}D'QC)' \\
B_x &= -B\Gamma^{-1}B' & B_\lambda &= -C'Q(I - D\Gamma^{-1}D'Q)C \\
E_x &= B\Gamma^{-1}D'Q & E_\lambda &= C'Q(I - D\Gamma^{-1}D'Q)
\end{aligned} \tag{3.18}$$

so that $A_\lambda = -A_x'$, $-B_x$ is symmetric and positive definite, and $-B_\lambda$ is symmetric and positive semi-definite (see Appendix B). The boundary constraints are

$$\begin{aligned}
x(0) &= x_0 \\
x(T) &= x_d \\
\lambda(T) &= v
\end{aligned} \tag{3.19}$$

We apply an extension of the *backward sweep method* presented in [9]. We assume that the boundary condition on $x(T)$ is given as a linear combination of the initial conditions x_0 and v and the reference y_f

$$x(T) = \Theta x_0 + \Sigma y_f + \Phi v \tag{3.20}$$

By linearity, the initial condition $\lambda(0) = v$ is given by a linear combination of the same parameters

$$\lambda(0) = Sx(0) + Zy_f + Wv \tag{3.21}$$

Of course, by optimality, any time $t < T$ is an "initial time" for which this is true, so the matrices Θ, Σ, Φ, S, Z, and W may be treated as time-varying matrices that maintain these conditions for all time

$$\begin{aligned} \lambda(t) &= S(t)x(t) + Z(t)y_f + W(t)v \\ x(T) = x_d &= \Theta(t)x(t) + \Sigma(t)y_f + \Phi(t)v \end{aligned} \tag{3.22}$$

For (3.22) to satisfy the boundary constraints (3.19), we arrive at the following set of sufficient boundary conditions

$$\begin{aligned} &S(T) = 0,\ Z(T) = 0,\ W(T) = I \\ &\Theta(T) = I,\ \Sigma(T) = 0,\ \Phi(T) = 0 \end{aligned} \tag{3.23}$$

Now, substituting $\lambda(t)$ from (3.22) into the co-state equation of (3.17) and relating the terms, we get

$$\begin{aligned} 0 =&[\dot{S} + SA_x + SB_xS - B_\lambda - A_\lambda S]x \\ &+ [\dot{Z} + (SB_x - A_\lambda)Z + SE_x + E_\lambda]y_f \\ &+ [\dot{W} + SB_xW - A_\lambda W]v \end{aligned} \tag{3.24}$$

where equality to zero must be true for all x and v. We therefore sufficiently satisfy (3.24) by setting each expression to zero individually

$$\begin{aligned} \dot{S} + SA_x + SB_xS - B_\lambda - A_\lambda S &= 0 \\ \dot{Z} + SB_xZ - A_\lambda Z + SE_x - E_\lambda &= 0 \\ \dot{W} + SB_xW - A_\lambda W &= 0 \end{aligned} \tag{3.25}$$

Subject to the boundary conditions in (3.23), we are left with a differential Riccati equation (DRE) in S (which must be symmetric) and two ordinary differential equations (ODEs), which can be solved numerically.

Differentiating the ψ expression from (3.22) and substituting the expression for

$\dot{x}(t)$ from (3.17), we similarly arrive at

$$\begin{aligned} \dot{\Theta} + \Theta A_x + \Theta B_x S &= 0 \\ \dot{\Sigma} + \Theta B_x Z + \Theta E_x &= 0 \\ \dot{\Phi} + \Theta B_x W &= 0 \end{aligned} \tag{3.26}$$

The relationship between A_λ and A_x yeilds $\Theta(t) = W'(t)$, and so we are simply left with two ODEs that are functions of S, Z, and W.

Suppose we obtain the numeric solutions for each of the above matrix differential equations for a fixed final time T. By the boundary condition on $x(T)$, we use the x_d expression in (3.22) and obtain $v = \Phi^{-1}(t)\left(x_d - W'(t)x - \Sigma(t)y_f\right)$ (for a proof that Φ may be inverted at t, see Appendix B). Therefore, the feedback control law is

$$\begin{aligned} u = -\Gamma^{-1}\{&\left(D'QC + B'(S - W\Phi^{-1}W')\right)x + \left(-D'Q + B'(Z - W\Phi^{-1}\Sigma)\right)y_f \\ &+ B'W\Phi^{-1}x_d\} \end{aligned} \tag{3.27}$$

Parameterizing the Unconstrained Control Law

It is still necessary to search for the optimal value of the final time T for the optimization which depends on x_0 and y_f. Clearly, it is impractical to consider calculating the above ODEs and DRE for each potential value of T on every time step of the system to find the optimal such value.

First, we note that the differential equations (3.25) and (3.26), as well as their boundary constraints, depend completely upon the system and the final time, not x_0, x_d, or y_f. Therefore, we simply solve (3.25) and (3.26) using some very large final time T^* and denote the cooresponding solutions as S^*, W^*, Z^*, and so on. Suppose now, for some initial state x_0, we know the optimal final time is $0 < \tilde{T} < T^*$, then the solutions $\tilde{S}$, $\tilde{W}$, $\tilde{Z}$, etc. to (3.25) and (3.26) for a final time $\tilde{T}$ are simply computed as $\tilde{S}(t) = S^*(T^* - \tilde{T} + t))$, $\tilde{W}(t) = W^*(T^* - \tilde{T} + t)$, etc.

If the pair $(-B_\lambda^{1/2}, A_x)$ is detectable and the pair (A_x, B) is controllable, then, in choosing some very large time T^*, we note that the DRE that determines S (3.24)

stabilizes to the solution S_∞ of the corresponding algebraic Riccati equation (ARE)

$$S_\infty A'_x + A'_x S_\infty + S_\infty B_x S_\infty - B_\lambda = 0 \tag{3.28}$$

Therefore, for a large final time T, we can approximate $S(0)$ (since the DRE (3.24) is integrated backwards starting from T) from the solution S of (3.24) as $S(0) \approx S_\infty$.

Furthermore, we may impose some ϵ tolerance of error for this approximation by finding a final time $\tau < \infty$ such that the solution to the DRE for S with the boundary constraint at τ gives $\|S(t) - S_\infty\| < \epsilon$ for all $t \leq 0$.

By time-invariance as well as convergence of the DRE, solving the DRE for any greater final time $(\tau + T)$ yields $\|S(t) - S_\infty\| < \epsilon$ for all $t \leq T$.

In Appendix B, we show that all the matrices parameterizing the control law converge to some constantas $t \to -\infty$. Therefore, we can, with finite memory, approximate with arbitrary precision the feedback control law. By picking some very large final time T^*, the solutions to the DRE and ODEs approximate to constants for all $t < 0$.

Of course, the difficulty of determining the optimal final time for a given intial state, target state, and output reference still remains. We note that only a bounded portion of the target state space and output reference space is likely to be applied to the controller in practice, and so we can quantize this bounded subset of the space along with the state space to store, in memory, a rough approximation to the optimal time-to-go for all (finite) tuples (x_0, x_d, y_{ref}) that lie in the quantization set.

We state that the quantization set is finite despite our not bounding the initial state space because, assuming that for all $\alpha > 1$ the optimal time-to-go for αx_0 is greater than that for x_0, the optimal time-to-go will grow monototically as $\alpha \to \infty$. Regardless of whether the optimal time-to-go approaches some finite value asymptotically or grows unbounded, we can always bound the maximum final time T needed to store the paramterization of the control law. This assumption however needs to be proved formally.

Therefore, applying the assumption, a natural bound along the initial-state space

of the quantization already exists. For an initial state where the optimal time-to-go (from that state to a given target state under the influence of a given output reference) is greater that T^*, we use the constant approximations for S, W, Φ, Z, and Σ to provide an approximately-optimal control policy. Once the state lies within the region of the state space where, for the given target and reference, the optimal time-to-go is less than or equal to T^*, we apply the optimal time-to-go that is approximated by the quantization as well as the corresponding optimal control law.

Constrained Systems

For constrained systems, we design a controller only for discrete-time systems since, when expressed as a multiparametric-quadratic program, the closed-loop optimal controller may be stored in memory.

The optimization problem is similar to that of the unconstrained case: determine an optimal final time N and control input $u(n)$, $0 \leq n \leq N$, so that we minimize

$$J = \sum_{n=0}^{N} \|Cx + Du - y_f\|_Q^2 + \|u\|_Q^2 \tag{3.29}$$

subject to the boundary conditions $x(0) = x_0$ and $x(N) = x_d$ as well as the constraints $x(n) \in P_x$, $u(n) \in P_u$, and $y_f \in P_y$, where P_x, P_u, and P_y are all polytopes.

Of course, the above formulation is that of a multiparametric-quadratic program (MP-QP) in x_0, x_d, and y_f that may be solved offline to yield a solution which can be referenced online.

Clearly, for a fixed and finite set of initial states, target states, and output reference values, there exists is a maximum final time N^* such that for all final times $N > N^*$, the solution to is not optimal for any combination of values from this finite set. Because the computation of the optimal time-to-go must be performed offline and approximated online, a quantization of the space is required whereby a finite set (finite by the fact that the state space and reference space are bounded to polytopes) of initial states, target states, and output reference values is used to reference the optimal time-to-go for all points.

Applying the maximum N^* arising from such a quantization, we store the solution to the corresponding MP-QP in memory for all $0 \leq N \leq N^*$. In real-time, we can use the quanization to reference an approximately-optimal final time. Once a final time is chosen, the solution to the corresponding MP-QP is used to obtain the feedback law.

By the fact that the control input lies in the polytope, it is possible, for some initial state x_0 and some given final time N, that no admissible control input that gives $x(N) = 0$. Because we are quantizing the space and approximating the actual optimal final times by a finite set of final times, it is possible that the approximating final time makes the problem not feasible for the actual initial state. In this case, we can simply increment the final time by one until the corresponding MP-QP for the initial state has a feasible solution.

3.5.2 A Low-Memory Control Law

For both constrained and unconstrained systems, the above techniques give us a means for efficiently storing a parameterization of an otherwise complicated control law in memory for relatively fast computation of the optimal closed-loop control.

However, each of these approaches also requires a potentially large quantization of the state-reference space which may require too much memory for low-cost embedded hardware. By the separation that exists between the switching control and the subsystem controllers, each subsystem controller needs only to satisfy the condition that, in some finite time, the state of the system is within the switching radius of the switching state. In this section, we present one possible alternative for the subsystem controller that is far simpler than the controllers provided in the previous sections and which still may provide meaningful performance.

For each mode p of the system, let u_p be the optimal, infinite-horizon LQR controller that minimizes

$$J = \int_{t=0}^{\infty} \| \begin{bmatrix} x \\ Cx + Du \end{bmatrix} - \begin{bmatrix} x_d \\ \hat{y}_{ref} \end{bmatrix} \|_Q^2 + \|u - u_{ref}\|_R^2 dt \tag{3.30}$$

where $\hat{y}_{ref} = Cx_d + Du_{ref}$ is achievable at x_d and minimizes $\|\hat{y}_{ref} - y_{ref}\|_Q^2$. The solution to this LQR problem can be found for both unconstrained and constrained systems[4].

Although this approach gives a low-complexity controller for each subsystem, its performance is not optimal with respect to the traditional metrics for optimality (H-2 or H-∞, for example), and so simulation may be required to determine if the quality of performance approximately satisfies the subjective performance requirements of the designer.

Chapter 4

Reducing the Complexity in Control of Systems with Unbounded Switching Regions

In this chapter, we propose a simple scheme by which (1.1) may be controlled under the following conditions:

- all of the subsystems are homogenous (i.e. $f_i(\alpha x, \alpha u) = \alpha f_i(x, u)$)
- all of the subsystems are unconstrained
- the switching region between any pair of subsystems is unbounded

We also assume that the translation vector is zero, and that the control goal is to minimize a cost function of the form

$$J = \int_0^\infty \|x\|_Q^z + \|u\|_R^z dt \tag{4.1}$$

where $z \geq 1$.

4.1 Dynamic Switching States

In chapter 3, it was shown that the use of switching states in switched control allows for a separation between subsystem control and the switching strategy. By the fact that the number of switching states was finite, it was possible to determine the optimal switching path using dynamic programming.

For the case of unbounded switching regions, the use of a finite number of switching states does not provide a satisfactory cover of the space, and so an alternative means for applying switching states needs to be considered. For this case, we will fully leverage the assumption that each subsystem (as well as the cost function in some sense) is homogenous.

Let C be the boundary of the unit sphere in the state space X. We choose $\{v_0, v_1, v_2, \ldots, v_n\}$ as a set of states in X that satisfy $v_0 = 0$ and where the v_i's for $i > 0$ are n distinct states on C. We term these states the *base states* of the system. Also, for each base state v_i, we choose a set of scalars $\Delta_i = \{\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{im_i}\}$ where $0 < \alpha_{ij} < 1$.

We define the set of *dynamic switching states of scale* r for the switched system as

$$V_r = \{r\alpha_{ij}v_i | \forall i, j\} \bigcup \{-r\alpha_{ij}v_i | \forall i, j\} \tag{4.2}$$

For simplicity, let $v_{ij} = \alpha_{ij}v_i$. We rewrite (4.2) as

$$V_r = \{rv_{ij} | \forall i, j\} \bigcup \{-rv_{ij} | \forall i, j\} \tag{4.3}$$

Essentially, the dynamic switching states (DSSs) of scale r are positively and negatively scaled copies of the base states that are contained in the open sphere $B(0, r)$.

It is additionally noted that each base state v_i as well as $-v_i$ are contained V_r for at least one $r > 1$. Also, $0 \in V_r$ for all r.

4.2 Switching Constraints

Assume momentarily that there are no switching constraints imposed on the system. If for an initial state x_0 the control inputs that minimize (4.1) are u^* and i^* (yielding a cost-to-go J), then, under the assumptions of homogeneity, any scaled version of the same initial state βx_0 has βu^* and i^* as the corresponding optimal control (yielding a cost-to-go $|\beta|^z J$).

We would like our problem formulation to have such a scaling property since, if the optimal control law is known from any state starting on on the unit shell C, we automatically know the optimal control law from any initial state which lies on some scaled version of C (which is, of course, the entire space X).

In the spirit of the SHRSG, we restrict mode switches to the DSSs of the system. Therefore, for any initial state x_0 with unity magnitude on C, the choices in control are to either stay in the same operating mode and optimally track the origin in infinite time, or to track and switch modes at one of the nonzero DSSs in V_1 in finite time. Of course, determining the DSS to track requires knowledge of the remaining cost-to-go from each of the DSSs.

We resolve this difficulty by posing the above problem at all scales of the system. That is, if the system's state is x, we only allow the controller to either optimally track the origin in infinite time, or to track and switch modes at one of the nonzero DSSs in $V_{\|x\|}$. Intuitively, we are simply scaling our decisions.

Under the above switching constraints, knowledge of the optimal cost $J^*(x_0)$ from a state x_0 to the origin implies that the optimal cost from βx_0 is $|\beta|^z J^*(x_0)$.

4.3 Constructing the Dynamic Robust Hybrid Switching Graph

4.3.1 Computing the Optimal Cost-to-Go for the DSSs

As with the SRHSG setup in Chapter 3, there is a nice separation in the continuous and switching portions of the control input that allow us to design custom controllers in each mode that are parameterized independently of the switching control law. For the cost function (4.1), we arrive at the following subsystem control objective: for a given initial state x_0 and target state x_d in mode p, choose a final time T and a control input u so as to minimize

$$J_p(x_0, x_d) = \int_0^T \|x\|_Q^z + \|u\|_R^z dt \tag{4.4}$$

subject to $x(T) = x_d$. We denote the resulting optimal cost $J_p^*(x_0, x_d)$. For $x_d = 0$, we automatically set $T = \infty$ so that the control law is the infinite-horizon regulator.

Of course, for a fixed initial state x_0 and target state x_d, the choice of operating mode will also have an impact on the cost. Let $p^* = p^*(x_0, x_d)$ be the operating mode (which may not be unique) that minimizes the cost from x_0 to x_d in (4.4) (i.e., $p^* = \arg\min_p J_p^*(x_0, x_d)$). Then $J_{p^*}^*(x_0, x_d)$ is the optimal cost over all operating modes (subscript "p^*") as well as control inputs and final tracking times.

Let $J^*(x)$ denote the optimal cost from a state x to the origin under the imposed switching constraints. We then have $J^*(\pm r v_{ij}) = J^*(\pm r \alpha_{ij} v_i) = |r\alpha_{ij}|^z J^*(v_i)$.

Therefore, the optimal cost $J^*(v_i)$ from any base state v_i must be equal to either

- the optimal infinite-horizon cost — $J_{p^*}^*(v_i, v_0)$ where $v_0 = 0$
- the sum of the cost to track some DSS $v \in V_1$ plus that DSS's optimal cost — $J_{p^*}^*(v_i, v) + J^*(v)$

Of course, the application of the DSS v_0 in the latter case is the same as in the former, and hence we no longer need to distinguish between the two cases.

Formally stated, for each non-zero base state v_i, there exists a DSS $v \in V_1$, where $v = \alpha v_j$ for some base state $v_j \neq v_i$, such that

$$\begin{aligned} J^*(v_i) &= J^*_{p^*}(v_i, v) + J^*(v) \\ &= J^*_{p^*}(v_i, \alpha v_j) + J^*(\alpha v_j) \\ &= J^*_{p^*}(v_i, \alpha v_j) + |\alpha|^z J^*(v_j) \end{aligned} \tag{4.5}$$

The above expression represents a simple linear relationship between the optimal costs of two base states v_i and v_j. We can leverage optimality further and express the same equality as the minimum cost obtained over all DSSs in V_1

$$\begin{aligned} J^*(v_i) &= \min_{v \in V_1} \{J^*_{p^*}(v_i, v) + J^*(v)\} \\ &= \min_{j \neq i; k} \{J^*_{p^*}(v_i, +\alpha_{jk} v_j) + J^*(+\alpha_{jk} v_j), J^*_{p^*}(v_i, -\alpha_{jk} v_j) + J^*(-\alpha_{jk} v_j)\} \\ &= \min_{j \neq i; k} \{J^*_{p^*}(v_i, +\alpha_{jk} v_j) + |\alpha_{jk}|^z J^*(v_j), J^*_{p^*}(v_i, -\alpha_{jk} v_j) + |\alpha_{jk}|^z J^*(v_j)\} \end{aligned} \tag{4.6}$$

where the second equality stems from the inclusion of both positively and negatively scaled copies of v_j in V_1. Of course, we can determine which of the two copies $+v_{jk}$ or $-v_{jk}$ is the optimal to apply without knowing the optimal cost from each because the optimal costs are the same ($J^*(v_j) = J^*(-v_j)$). Without loss of generality, we can simplify notation by assuming that the optimal choice is always the positively scaled DSS v_{jk}.

Because this is true for all i, we can solve the optimal cost $J^*(v_i)$ for all v_i by solving the following linear program

$$
\begin{aligned}
&\min \sum_i J^*(v_i) \text{ subject to} \\
&\qquad J^*(v_0) = 0 \\
&0 \le J^*(v_1) = \min_{j \neq i;k} \{J(v_i, \alpha_{jk} v_j) + |\alpha_{jk}|^z J^*(v_j)\} \\
&0 \le J^*(v_2) = \min_{j \neq i;k} \{J(v_i, \alpha_{jk} v_j) + |\alpha_{jk}|^z J^*(v_j)\} \\
&\qquad\qquad \vdots \\
&0 \le J^*(v_n) = \min_{j \neq i;k} \{J(v_i, \alpha_{jk} v_j) + |\alpha_{jk}|^z J^*(v_j)\}
\end{aligned}
\tag{4.7}
$$

With $J^*(v_i)$ known for all v_i, the optimal DSS to track from v_i is determined by searching over all DSSs $v \in V_1$, using the fact that $J^*(v) = |\alpha|^z J^*(v_j)$ for some base state v_j.

We denote the optimal DSS to track from v_i as $\chi_{p^*}(v_i)$. We use the subscript "p^*" to denote that we are applying the optimal mode during the tracking as well. By homogeneity, the optimal DSS to track from βv_i is simply given as $\chi_{p^*}(\beta v_i) = \beta \chi_{p^*}(v_i)$.

We define the *Dynamic Robust Hybrid Switching Graph* (DRHSG) $G = (V, E)$ by

$$
\begin{aligned}
&1.\ V = V_1 \bigcup \{v_0, v_1, \ldots, v_n\} \bigcup \{-v_0, -v_1, \ldots, -v_n\} \\
&2.\ (x_1, x_2) \in E \Leftrightarrow x_1 \text{ is a base state }, x_2 \in V_1, \text{ and} \\
&\qquad x_2 \neq \beta x_1 \text{ for any } \beta \neq 0 \\
&3.\ w\left((x_1, x_2)\right) = J^*_{p^*}(x_1, x_2)
\end{aligned}
\tag{4.8}
$$

4.3.2 Applying DRHSG for all States

The optimal cost for any non-base state $x \in C$ in mode p is simply given by

$$
J^*_p(x) = \min_{v \in V_1} J^*_p(x, v) + J^*(v) \tag{4.9}
$$

The optimal DSS to track is denoted as $\chi_p(x)$. By homogeneity, the DSS state to track for any non-DSS state x in mode p is $\|x\| \chi_p(\frac{x}{\|x\|})$. The appended DRHSG, just

as the appended SRHSG, is defined as including this new initial state, though we will not define it here formally.

Of course, it is impractical to consider performing the above computation online, and so a quantization of the state space is required to store an approximation of the control law in memory. As shown is Chapter 3 for the case z=2, the continuous portion of the control law may be parameterized fairly efficiently in memory if the subsystems are all linear. Therefore, if we quantize C to a finite set of approximation states $\tilde{X}_p$ for each mode p in the system, then, for each $x \in \tilde{X}_p$, the next state to track $\chi_p(x)$ as well as the optimal time in which to track that state may be stored in memory. If x is a DSS, the optimal mode to apply p^* may be stored as well since mode switches can occur at DSSs.

4.4 Stability

We now prove that a DRHSG-based controller asymptotically stabilizes a switched system. The proof relies on the assumption that optimal state trajectory $x(t)$ resulting from minimizing (4.4) is continuous as a function of the initial state x_0. This assumption needs to be proven.

For some DSS $v \in V_1$, $\chi_p^{-1}(v)$ is the set of points on C for which v is the optimal DSS to track in mode p (a base state will belong to one of these sets for some v and p). Let $P_{pv} = \overline{\chi_p^{-1}(v)}$ be the closure of this set, which is compact in R^n.

For an initial state $x_0 \in P_{pv}$, the closed-loop system will track v in some finite time. Let $x^*(x_0, t)$ denote the optimal state trajectory according to (4.4) from x_0 to v, and let T be the cooresponding optimal final time. Then, over the compact interval $t \in [0, T]$, $x^*(x_0, t)$ varies continuously as a function of t and so there is a maximum deviation $\|x^*(x_0, t^*)\| < \infty$ of the trajectory at some time $0 \leq t^* < T$.

If the optimal control trajectory $x^*(x_0, t)$ also varies continuously in the initial state x_0, then $x^*(x_0, t)$ is continuous over the compact set P_{pv}, and so there is a maximum deviation η_{pv} such that $\|x^*(x_0, t)\| < \eta_{pv} < \infty$ over all $x_0 \in P_{pv}$ and t.

Because the number of modes p and DSSs v in V_1 are finite, a finite number of

sets P_{pv} cover the unit shell C. If we let η be the maximum deviation over all η_{pv}, then the maximum deviation of any trajectory with an initial condition on C is η.

Since the next switching state is contained in $B(0,1)$, it can be shown by induction and homogeneity that the optimal trajectory $x(t)$ starting from any x_0 on C is contained in the ball $B(0,\eta)$.

Therefore, given any $\epsilon > 0$, choosing $\|x_0\| < \epsilon/\eta$ guarantees $\|x(t)\| < \epsilon$ for all t. Since $x(t) \to 0$ as $t \to \infty$, the system is asymtotically stable.

4.5 Principle of Optimality

It is clear that if the closed-loop system controlled by a DRHSG controller is sampled at the switching times, it obeys Bellman's Principle of Optimality. However, it does not possess this property when examined along the full state trajectory.

Consider the following example. Suppose that for each base state, there are only two cooresponding DSSs — a positive and negatively scaled version. If the initial condition is one of the base states $x(0) = v_i$, then, in some finite time T, we have $x(T) = \chi_{p^*}(v_i) \in V_1$.

Assume $x(T) \neq 0$ and take some state $x(t_1)$ along the state trajectory. If the system were to treat $x(t_1)$ as the "initial condition", then the system would not drive to the DSS $x(T)$ because $x(T) \notin V_{\|x(t_1)\|}$.

Because the principle of optimality does not apply, a "true" closed-loop controller cannot be used to stabilize the system. The system may only use a closed-loop controller to track an open-loop switching path.

It is believed that if switching *lines* rather than switching *states on lines* are used, the principle of optimality would apply because the same problem is always posed for all states of the system. However, it also may be for more computationally intensive to solve for the optimal trajectory.

Chapter 5

Application of SRHSG to the DISC Engine

We compare the SRHSG control methodology to a MIPC controller designed for the same application: the direct-injection stratified charge (DISC) engine. This application is particularly relevant because the DISC engine is a constrained system with demanding resource and performance requirements.

5.1 Overview of the DISC Engine

The DISC engine can be operated in two combustion regimes: homogeneous and stratified. In homogeneous operation, fuel is injected during the intake stroke, providing an approximately even air-fuel mixture throughout the cylinder. The characteristics of the engine are similar to that of the typical port-fuel injection (PFI) engine in terms of performance and emissions, and the air-to-fuel ratio (AFR) operates about the stoichiometric value of 14.62:1.

In the stratified operation, fuel is injected late into the compression stroke, forcing the fuel, under the influence of a specialized piston head, to be concentrated about the spark plug. The typical AFR for this mode of operation is about 35:1, significantly higher than that of the PFI engine. However, this operation regime also generates a significant amount of NOx byproduct that must be accounted for by a specialized NOx

catalyst in order for the engine to meet strict air-quality standards. These catalysts lose their effectiveness over time and must be purged by switching the engine for some time back into the homogeneous regime and running the AFR slightly rich of stoichiometry.

The choice of the operating mode depends upon the amount of brake torque demanded by the driver and the current state of the NOx catalyst. When the demanded torque is high (such as at high speeds or for fast acceleration), the engine should be operated in homogeneous operation. When the demanded torque is low to moderate, and the catalyst is operating effectively, stratified operation may be used to greatly increase the fuel economy. As the catalyst becomes saturated, the mode should be temporarily switched to the homogeneous regime until it is clean (termed a purge operation). Based upon the torque demanded by the driver and the catalyst state, a high-level algorithm determines the appropriate combustion mode to apply and computes a set point for the intake-manifold pressure, AFR, and the torque. The purpose of the DISC engine controller is to track these references in such a way as to guarantee convergence and quality of performance.

5.1.1 DISC Engine Model

We treat the simplified model of the DISC engine presented in [5], a derivative of the mean value model proposed and verified in [18]. In this section, we present a brief overview of the DISC engine parameters and present the form of the linearized, discrete-time DISC engine model.

Model Parameters

The tracking parameters (outputs) of the DISC engine are:

- Intake-manifold pressure p_m which governs the mass-flow rate of air into the cylinder and impacts the air-to-fuel ratio (AFR).
- Air-to-fuel ratio λ, the ratio of the mass flow rates of air and fuel into the cylinder and serves as the measure for fuel-economy.

- Brake torque τ, the torque provided to the driver which is the sum of the indicated torque (torque generated through combustion) with frictional and pumping losses.

- Combustion regime ρ where $\rho = 1$ designates stratified operation and $\rho = 2$ designates homogeneous operation. The DISC engine controller controls this parameter directly in order to track the reference operation mode.

The following parameters are treated as inputs to the engine:

- Mass flow rate of air through the throttle W_{th} used for controlling p_m.

- Fueling rate W_f that affects both the AFR and the brake torque.

- Spark timing δ that impacts the amount of torque generated through combustion.

There is a particular spark timing, termed the maximum-brake torque timing (δ_{mbt}), such that the amount of brake torque generated through combustion is maximized when δ attains this value. δ_{mbt} is a function of λ, and the brake torque quadratically depends upon the amount of deviation between δ and δ_{mbt}.

Constraints

In both operating modes, there exist actuator saturations and other practical limitations that are treated as hard constraints on the ranges of W_{th} , W_f, and δ. By convention, we restrict the spark timing to the interval $[0, \delta_{mbt}]$, where the upper limit (δ_{mbt}) is variable. In general, we denote the minimum admissible value for a parameter v as v_{min} and the maximum admissible value as v_{max}.

To avoid misfirings and excessive emissions caused by either too rich or too lean an air-fuel mixture, the AFR λ is specially bounded to a range that depends only upon the combustion regime. The output parameters p_m and τ are naturally bounded by all of the above limitations.

We assume that all references to the system are *achievable*; that is, there is an input to the system that will track that reference is steady-state. Hence, all references are bounded as well.

Linear and Discretized Model

The general form of the linear, discrete-time model of the DISC engine is presented below (the dependence on ρ is implicit):

$$\begin{aligned} \hat{p}_m(n+1) &= a\hat{p}_m(n) + \begin{bmatrix} b & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{W}_{th}(n) \\ \hat{W}_f(n) \\ \hat{\delta}(n) \end{bmatrix} \\ \begin{bmatrix} \hat{p}_m(n) \\ \hat{\lambda}(n) \\ \hat{\tau}(n) \end{bmatrix} &= \begin{bmatrix} 1 \\ c_2 \\ c_3 \end{bmatrix} \hat{p}_m(n) + \begin{bmatrix} 0 & 0 & 0 \\ 0 & d_{22} & d_{23} \\ 0 & d_{32} & d_{33} \end{bmatrix} \begin{bmatrix} \hat{W}_{th}(n) \\ \hat{W}_f(n) \\ \hat{\delta}(n) \end{bmatrix} \end{aligned} \tag{5.1}$$

where $\hat{v}$ denotes the normalized value of the parameter v to its operating point about which the approximating linearization is performed.

5.2 MPC Control of the DISC Engine

In this section, we give a very brief, qualitative description of the MIPC controller presented in [5] for controlling the DISC engine.

The MIPC problem formulation is as follows: given the reference y_{ref} and a nominal input u_{ref} (an "equilibrium" input for the system corresponding to the "equilibrium" output y_{ref} according to the nonlinear DISC engine model), find an N-step input horizon for the input $u(n)$ and the combustion mode $\rho(n)$ so as the minimize

$$J = \sum_{n=0}^{\infty} \|Q \begin{bmatrix} y_{ref} - y(n) \\ \epsilon_\tau(n) \\ \epsilon_\lambda(n) \end{bmatrix} \|_1 + \|R \begin{bmatrix} u_{ref} - u(n) \\ \rho(n) - \rho_{ref} \\ s(n) \end{bmatrix} \|_1 \tag{5.2}$$

where

$$\begin{aligned} \epsilon_\tau(n+1) &= \epsilon_\tau(n) + T(\tau(n) - \tau_{ref}) \\ \epsilon_\lambda(n+1) &= \epsilon_\lambda(n) + T(\lambda(n) - \lambda_{ref}) \end{aligned} \tag{5.3}$$

subject to $x(n)$, $y(n)$, and $u(n)$ are all admissible for each of the N future time samples. The positive definite matrices Q and R individually weight the errors in the output and the input, the magnitude of the parameter $s(n)$, and the integrated error between the torque and the AFR which is meant to provide zero-offset tracking of these two parameters.

To guarantee that the above QP is feasible, the admissible set of parameters is no longer restricted to polytopic sets but to the entire space. The *slack* variable $s(n)$ is introduced into the problem formulation to relax the original hard constraints to a set of soft constraints that heavily penalizes though allows for constrained parameters to admit values in the prediction horizon that violate their original polytopic constraints.

The stability, convergence, and performance properties of the above controller are not discussed in [5]. Though the problem seems to be formulated as an optimal control problem, in fact, optimizing along a short-look ahead horizon does not guarantee performance nor convergence to the reference. Though the controller is designed to drive the system to the reference by "forcing" both the output and the input to attain pre-specified, nominal values, formal results proving this do not exist. Furthermore, by allowing physically constrained parameters to be modeled as unconstrained parameters in the QP, it is possible that the system could potentially suffer from integrator windup in the parameters ϵ_τ and ϵ_λ, which may negatively impact stability and performance.

5.3 SRHSG Control of the DISC Engine

5.3.1 Subsystem Controllers

Before the switching strategy for the DISC engine can be computed, controllers for each subsystem of the DISC engine need to be constructed. Therefore, we consider

the subproblem of tracking a particular reference in the same mode subject to the cost function

$$J = \sum_{k=0}^{T} \|Cx + Du - y_{ref}\|_Q^2 \tag{5.4}$$

From (5.1), we see that there exists a nice separation between the input that affects the state and the inputs that affect the output in that, under the assumption that all references are achievable, allows us to optimally track a reference with respect to simply by tracking the IMP in minimum time and minimizing the error of the AFR and the torque at every time step.

To optimally determine values for the inputs $W_f(n)$ and $\delta(n)$ so as to minimize the errors in λ and τ while satisfying hard constraints, we use a quadratic program (QP) that depends upon the plant's state and the reference. To track the state in minimum time, we implement a basic discrete-time bang-bang controller for determining $W_{th}(n)$

$$\hat{W}_{th}(n+1) = \text{sat}\left(\frac{\begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\hat{y}_{ref} - a(\rho)\hat{x}(n)}{b(\rho)}\right) \tag{5.5}$$

The controller for determining $W_f(n)$ and $\delta(n)$ is a QP that seeks to minimize the error in λ and τ while meeting hard constraints on the inputs and the outputs. Given a reference output y_{ref} and combustion mode ρ, let

$$\begin{aligned} y_{23} &= \begin{bmatrix} \lambda & \tau \end{bmatrix} & u_{23} &= \begin{bmatrix} W_f & \delta \end{bmatrix} \\ Q_{23} &= PQP^T & D_{23}(\rho) &= PD(\rho)P^T \\ y_{23}^{ref} &= Py_{ref} & C_{23}(\rho) &= PC(\rho)P^T \end{aligned} \tag{5.6}$$

where $P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

The optimizing quadratic program is

$$
\begin{aligned}
& y_{23} = \arg\min_{\tilde{y}_{23}} \|y_{23}^{ref} - \tilde{y}_{23}\|_{Q_{23}}^2 \\
& \text{subject to} \\
& \begin{bmatrix} \hat{\lambda}_{min} \\ \hat{\tau}_{min} \end{bmatrix} \leq \tilde{y}_{23} \leq \begin{bmatrix} \hat{\lambda}_{max} \\ \hat{\tau}_{max} \end{bmatrix} \\
& \begin{bmatrix} \hat{W}_{f,min} \\ \hat{\delta}_{min} \end{bmatrix} \leq D_{23}^{-1}(\rho)(\hat{\tilde{y}}_{23} - C_{23}(\rho)\hat{x}) \leq \begin{bmatrix} \hat{W}_{f,max} \\ \hat{\delta}_{mbt} \end{bmatrix}
\end{aligned}
\tag{5.7}
$$

where δ_{mbt} is an affine function of the AFR and the combustion regime.

The above QP determines the optimal value for y_{23}, which can then be used to compute the control input u_{23}. However, solving (A.2) at every time sample of a fast-paced system such as the DISC engine is not practical, and so we employ the multi-parametric quadratic programming (MPQP) techniques detailed in [4]. Approximately 20 partitions in the parameter space are required to store the exact solution (A.2) in memory.

Of course, even if multiple linearizations are utilized, the controller is still open-loop, so zero-offset tracking cannot be guaranteed. Integral action is applied to (A.2) to obtain a closed-loop controller for λ and τ.

$$
\begin{aligned}
& 1.\ e \leftarrow (1-\alpha)\left(y_{23}^{ref} - y_{23}(n)\right) | e_I(n-1) \\
& 2.\ \text{Let } y'_{23} \text{ be the solution to (A.2) applying the reference } \tilde{y}_{23}^{ref} = y_{23}^{ref} + e \\
& 3.\ e_I(n) = y'_{23} - y_{23}^{ref} \\
& 4.\ \text{Compute } u_{23} \text{ accordingly from } y'_{23}
\end{aligned}
\tag{5.8}
$$

where the constant $0 < \alpha < 1$ determines the trade off between the convergence rate and sensitivity to noise. It can be shown that (5.8) asymptotically stabilizes λ and τ to their optimal values with respect to a fixed mode of operation and system state. We refer the reader to Appendix A for the proof of this result as well as a comparison of (5.8) to another controller designed for the same purpose.

Approximating via Muliple Linearizations

It is shown in Appendix A how to adapt the torque and AFR controller for use with multiple linearizations of the nonlinear DISC engine model. For brevity, we do not discuss these approaches here as they are not relevant, but we note that the simulations in this chapter make use of such multiple linearizations.

5.3.2 The Full Controller

We apply (3.9) using the control law constructed from (5.5) and (5.8) to build the full controller for the DISC engine. We modify (5.5) so that the tracking state x_d provided by the switching path is tracked

$$\hat{W}_{th}(n+1) = \text{sat}\left(\frac{x_d - a(\rho)\hat{x}(n)}{b(\rho)}\right) \tag{5.9}$$

Algorithm (5.8) remains unchanged for determining the values of $W_f(n)$ and $\delta(n)$ as it minimizes the error between the reference and the output AFR and torque.

Although usable SRHSG tables of practical sizes can easily be generated for the DISC engine, we briefly consider in this section a means for significantly reducing the table's memory requirements for systems possessing scalar states. Consider the rules below

- *Rule A*: If, for a given state x_0 and reference y_{ref}, the tracking state is x_d, then the same tracking state should be used for all states in between x_0 and x_d.

- *Rule B*: The operating mode may change only once when tracking a reference.

Applying these assumptions, the SRHSG table format may be modified. For each reference approximation point $\tilde{y}_{ref} \in \tilde{Y}_\rho$ in reference mode ρ, the following data structure may be used

$$T_D\left(\tilde{y}_{ref}, \rho\right) = \begin{bmatrix} -\infty & x_{l2} & x_{l3} & \cdots & x_{ln} \\ x_{s1} & x_{s2} & x_{s3} & \cdots & x_{sn} \end{bmatrix} \tag{5.10}$$

By Rule B, it is redundant to store a table for the case that the state and reference are in the same mode, so table (5.10) assumes that the system is in a different mode than the reference. The first row specifies the range of points that drive to a switching state, the second row is the switching states to use. The algorithm for getting the next switching state is

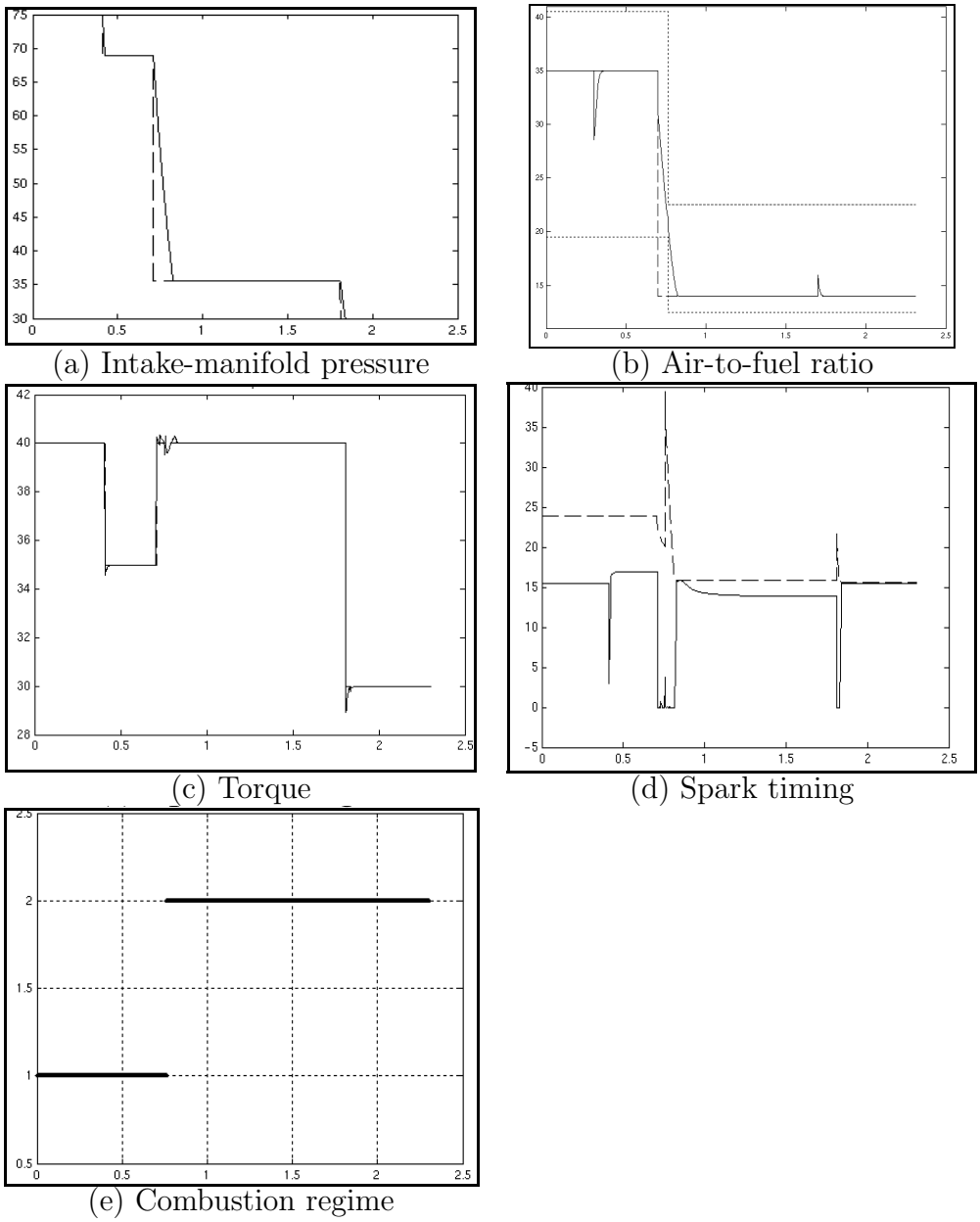
$$\begin{aligned} &1.\ T \leftarrow T_D\left(APY_\rho(y_{ref}), \rho\right) \\ &2.\ i \leftarrow \max\left\{j | x(n) \geq [T]_{1,j}\right\} \\ &3.\ x_d \leftarrow [T]_{2,i} \end{aligned} \tag{5.11}$$

5.4 Results and Comparison

The simulation results for the SRHSG-controlled system were obtained using a 22KB table with the following attributes: 106 switching points along in the IMP, 49 and 36 approximation points along the IMP in the stratified and homogenous modes respectively, and 48 and 29 approximation points in the reference output space in the stratified and homogenous modes respectively. It took roughly 2 hours to generate the table using a 1GHz PC.

The left column of Figure 5-1 shows the system's responses under SRHSG control, and the right column shows some of the the control inputs. The system is initialized in the stratified regime, and tracks the IMP, AFR, and torque references. The use of MP-QP allowed for a very fast tracking of the latter two parameters while satisfying the hard constraints on the inputs. When the reference indicates switching to the homogenous regime (at the time when the reference AFR drops to roughly 14:1), the controller examines the SRHSG table and selects a switching point along the IMP (in thiscase, the switching point is at 52 kPa). When the pressure reaches this point, the system switches modes, and the controller drives to the reference IMP. The mode shift causes a change in the system dynamics, and, consequently, a perturbation in the torque and AFR. This disturbance, however, is relatively insignificant, meaning the SRHSG shortest-path algorithm chose a good switching point. The response may be compared to the response given by the MIPC controller shown in Figure 5-2 where

convergence to the target is not guaranteed.



(a) Intake-manifold pressure

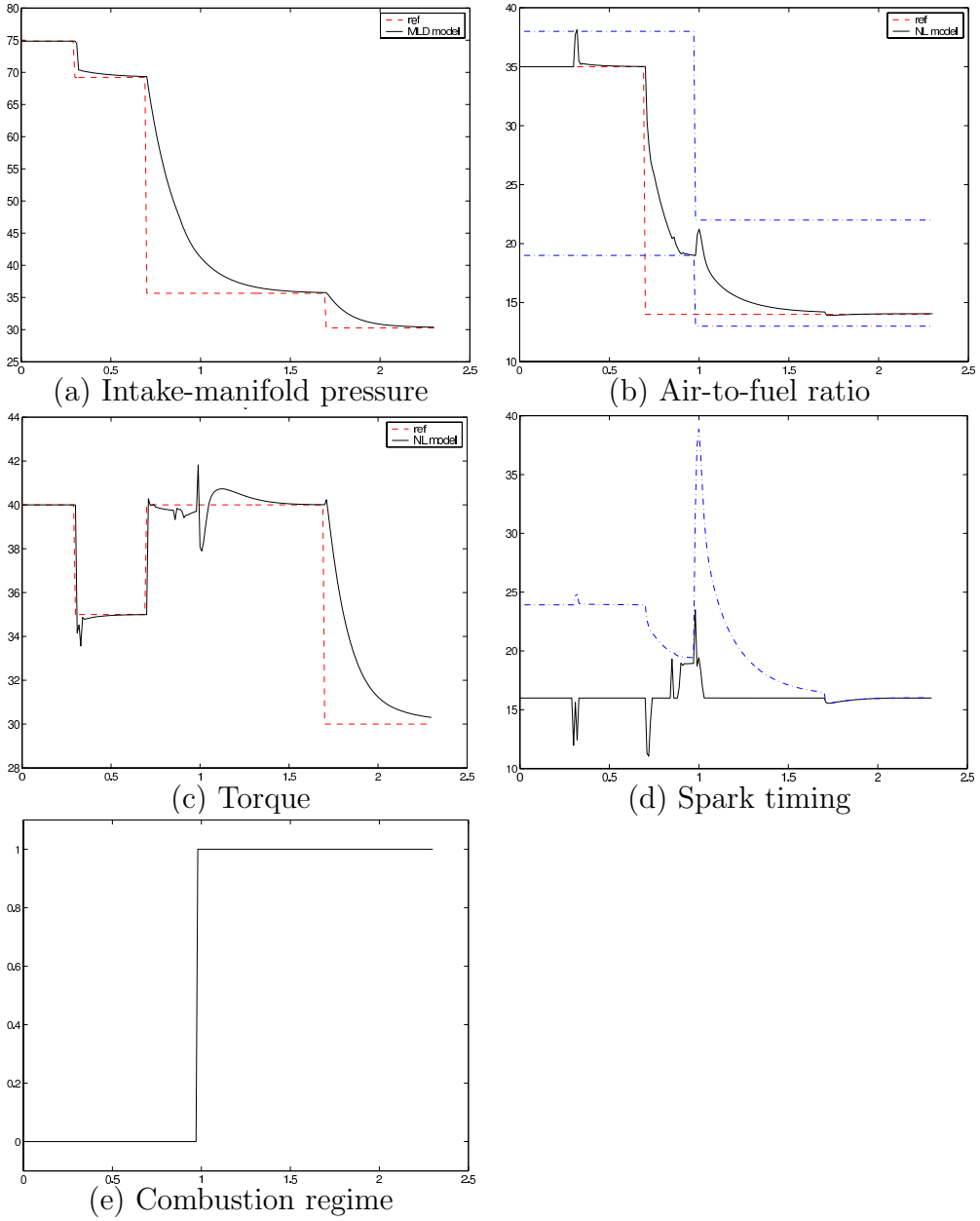
(b) Air-to-fuel ratio

(c) Torque

(d) Spark timing

(e) Combustion regime

Figure 5-1: SRHSG simulation results for the full controller applied to the nonlinear DISC engine. Solid line — response of the system; Dashed line reference (except for spark timing where dashed line is MBT spark timing. The fine-dotted lines in AFR represent the AFR boundaries).



(a) Intake-manifold pressure

(b) Air-to-fuel ratio

(c) Torque

(d) Spark timing

(e) Combustion regime

Figure 5-2: MIPC simulation results for the full controller applied to the nonlinear DISC engine. Solid line — response of the system; Dashed line reference (except for spark timing where dashed line is MBT spark timing. The fine-dotted lines in AFR represent the AFR boundaries).

Chapter 6

Conclusions and Future Work

In this paper, we presented two new approaches to controlling switched systems with both bounded and unbounded switching regions that reduced the complexity of finding both an optimal control input and mode sequence to a dynamic programming problem in tracking an optimal switching path, using pre-designed feedback controllers for each subsystem of the switched system. The switching path is completely determined by finding the minimum-cost sequence of switching states to track in each mode.

Unlike other approaches, our both static and dynamic RHSG

- do not require a pre-determined, finite length mode sequence
- are inherently immune to the Zeno effect
- are guaranteed, under reasonable assumptions, to converge to the target state.

Furthermore, by separating the subsystem controllers from the switching path, the subsystem controllers are parameterized separately from the switching law. A further reduction in complexity may be obtained by reducing the complexity of the subsystem controllers by considering controllers which do not provide optimal performance in the traditional p-norm sense, though the designer would need to determine whether such a sacrifice in performance is worth the benefits in simplicity. Our work does not provide a quantitative measure of the resulting performance in this case.

Future work in the control of constrained switched systems should include examining other approaches to reducing controller decisions to obtain simple and effective controllers, simplifying the individual subsystem controllers, and providing some measure of the system's stability (not simply convergence to the target) for SRHSG.

For unconstrained switched system, work into applying the concept of switching lines rather than states should offer a control scheme for which the principle of optimality applies, allowing for the use of closed-loop controllers.

Appendix A

Proof of Stability of the Torque and AFR Subsystem Controller

In this appendix, we prove the stability properties of the torque and AFR subsystem controller. For simplicity in presentation, we use different notation from that used in Chapter 5. Note, in particular, the following changes

- x is the state of the closed-loop integrator system, not the intake-manifold pressure
- y_{ref}, not y_{23}^{ref} is the torque and AFR reference
- u, not u_{23}, is the spark timing and fueling rate inputs
- D and C, not D_{23} and C_{23} are the corresponding system matrices for the subsystem.

A.1 Linear and Disceretized Model

The subsystem of interest in this appendix is that which controls only the latter two output parameters of the DISC engine model. As can be seen in (5.1), there is a nice separation between the input that regulates the state (p_m), and the inputs that regulate the remaining outputs (λ and τ). We assume that the state is controlled

optimally (by a bang-bang controller, for instance) and so in further assuming the target reference is achievable, we can optimally control λ and τ independently of p_m.

Presented below is the sub-system of interest

$$y(n) = \begin{bmatrix} \hat{\lambda}(n) \\ \hat{\tau}(n) \end{bmatrix} = C\hat{p}_m(n) + Du(n) \tag{A.1}$$

where, for simplicity, we substituted the output and input vectors with y and u respectively. D is invertible.

A.2 Optimal Torque and AFR Control

For a given reference $y_{ref} = \begin{bmatrix} \lambda_{ref} & \tau_{ref} \end{bmatrix}^T$ and intake-manifold pressure p_m, it is the responsibility of the torque and AFR controller to determine admissible values for W_f and δ so as to make the outputs λ and τ "optimally close" to their respective set-points. For this, we seek to minimize the standard 2-norm distance of the output and reference. Thus, the difficulty of computing the appropriate inputs is reduced to a quadratic program (QP).

However, solving a QP at every step of the controller for a fast-paced system like the DISC engine is clearly impractical. We leverage the fact that the constraints on y_{ref} as well as on y, u, and p_m are all polytopic, reforming the problem to that of a multiparametric-quadratic program (MP-QP). The solution to a MP-QP, in this case, is an explicit piecewise affine function of y_{ref} and p_m that can be evaluated practically in real-time. We do not discuss MP-QPs in detail in this paper as it is only the QP portion of the problem that is important for the controller analysis.

We denote the quadratic program used to determine the control input u as Q_p :

$\Re^2 \rightarrow \Re^2$. It is presented below

$$
\begin{aligned}
&\arg\min_{\tilde{y}}(y_{ref} - \tilde{y})' Q (y_{ref} - \tilde{y}) \\
&\text{subject to} \\
&\begin{bmatrix} \hat{\lambda}_{min} \\ \hat{\tau}_{min} \end{bmatrix} \leq \tilde{y} \leq \begin{bmatrix} \hat{\lambda}_{max} \\ \hat{\tau}_{max} \end{bmatrix} \\
&\begin{bmatrix} \hat{W}_{f,min} \\ \hat{\delta}_{min} \end{bmatrix} \leq D^{-1}(\tilde{y} - C\hat{p}_m) \leq \begin{bmatrix} \hat{W}_{f,max} \\ \hat{\delta}_{mbt} \end{bmatrix}
\end{aligned}
\tag{A.2}
$$

where Q is a positive definite, symmetric matrix. We implicitly treat the dependence of Q_p on p_m and so it is not a parameter of the function.

Q_p maps the reference y_{ref} to the optimal achievable output y', from which the control signal to apply is determined as $u = D^{-1}(y' - C\hat{p}_m)$. Due to the invertibility of D, it is only necessary to formulate the problem in terms of one of these parameters, allowing for all the constraints on both y and u to be treated as just a single set of constraints on y. For simplicity, we denote the polytope bounding y as P.

A.2.1 Closed-Loop Controller

The algorithm for computing the optimal W_f and δ for a given reference and current plant output is presented below

$$
\begin{aligned}
&1.\ x(n+1) = (1-\alpha)(y_{ref} - y(n)) + e_I(n) \\
&2.\ y'(n+1) = Q_p(y_{ref} + x(n+1)) \\
&3.\ e_I(n+1) = y'(n+1) - y_{ref} \\
&4.\ u(n+1) = D^{-1}(y'(n+1) - C\hat{p}_m)
\end{aligned}
\tag{A.3}
$$

where:

- $0 < \alpha < 1$ is a constant that impacts the speed of convergence (smaller values give faster convergence while larger values decrease the closed-loop sensitivity to noise)

- x is the "true" integrated error between the reference and the output

- y' is the optimal point in P with respect to $y_{ref} + x$ to which the system should strive to attain

- e_I is the integrated error after anti-windup is applied (a "corrected" version of x).

(It should be noted that it is assumed the solution to Q_p is an MP-QP that assumes the reference is bounded to a particular polytope. This may be accounted for by saturating x to some arbitrary polytope in (A.3). This correction is of little importance, however, and so we ignore it.)

To simplify analysis, we let Q be the identity matrix. This is done without a loss in generality since we can always apply an invertible transformation of coordinates to the system in the form of $\sqrt{Q}$, for which an identical stability analysis may be performed. We also assume, without loss of generality, that the output of the plant $y(n)$ is exactly equal to $y'(n)$ (if it is not, we could apply a translation of coordinates and perform the same analysis). Thus, we can represent the entire closed-loop system (plant and controller) in the following state-space form

$$\begin{aligned} x(n+1) &= \alpha \left[Q_p(y_{ref} + x(n)) - y_{ref} \right] \\ y(n) &= Q_p(y_{ref} + x(n)) \end{aligned} \tag{A.4}$$

Of course, y_{ref} may not be an achievable reference since it is assumed only that the reference $\begin{bmatrix} \hat{p}_{m,ref} & {y_{ref}}^T \end{bmatrix}^T$ is achievable. If p_m is not at its target set-point $p_{m,ref}$, there is no guarantee that y_{ref} may be reached by an admissible u. In this case it is desirable to drive the system to the optimal admissible output value given by $y_{opt} = Q_p(y_{ref})$.

A.3 Stability of the Closed-Loop System

A.3.1 Equilibria

Before determining an equilibrium point for (A.4), we present some basic results. In all, we let $F : \Re^n \to \Re^n$ be the quadratic program given by $F(y) = \arg\min_{\hat{y} \in M} \|y - \hat{y}\|$ for some polytope M.

Lemma A.1. *For some $y \notin M$, there exists a hyperplane W that separates the space into two disjoint regions R_1 and R_2 such that y is contained in R_1 and M is contained in R_2. Furthermore, the vector $(y - y^*)$ is perpendicular to W.*

Proof. Let W be the hyperplane containing the point $y^* = F(y)$ and perpendicular to the vector $(y - y^*)$. We prove the claim by contradiction. Suppose some point $p \in M$ is contained in R_1. By the convexity of M, the line segment $l = \{(1-\beta)y^* + \beta p | 0 \leq \beta \leq 1\}$ is contained in M, and, furthermore, it intersects W at only y^*.

Let B_y be the closed ball of radius $\|y - y^*\|$ centered at y. Then l intersects B_y at y^* though it is not tangential to it. Therefore, there is some point in l (also in M) contained in the interior of B_y which is a contradiction of the optimality of y^*. □

Corollary A.2. *If $F(y) = y^*$ and $y \notin M$, then $F(\beta y + (1-\beta)y^*) = y^*$ for all $\beta \geq 0$.*

We can now easily show that $x_e = \alpha(y_{opt} - y_{ref})$ is an equilibrium for the closed-loop system (A.4).

$$\begin{aligned}\alpha\{Q_p(y_{ref} + x_e) - y_{ref}\} =& \alpha\{Q_p(\alpha y_{opt} + (1-\alpha)y_{ref}) - y_{ref}\} \\ =& \alpha(y_{opt} - y_{ref}) \\ =& x_e\end{aligned}$$

At this equilibrium state, u, which is uniquely determined by x and y_{ref} drives the plant output to y_{opt}. Without loss of generality, we will assume that $y_{opt} = 0$, which is easily obtained by simply applying a translation of coordinates. Letting $z_e = x - x_e$,

we rewrite (A.4) as

$$\begin{aligned} z(n+1) &= \alpha\left[Q_p(q+z(n)) - y_{opt}\right] = \alpha\left[Q_p(q+z(n))\right] \\ y(n) &= Q_p(q+z(n)) \end{aligned} \tag{A.5}$$

where $q = \alpha y_{opt} + (1-\alpha)y_{ref} = (1-\alpha)y_{ref}$. An equilibrium state for this system is clearly $z_e = 0$.

A.3.2 Asymptotic Stability

It remains to show that the equilibrium point $z_e = 0$ of (A.5) is globally asymptotically stable. Before proceeding, we establish the contraction property of the quadratic program F. It is assumed $0 \in M$.

Lemma A.3. *$\|F(y)\| \leq \|y\|$ for all y.*

Proof. Let B_0 be the closed ball with radius $\|y\|$ centered at 0, and let x be a point on the surface of B_0 that intersects the ray $R = \{\beta F(y) | \beta \geq 0\}$.

Consider the existence of a positive real constant γ such that the point $y^* = \gamma x$ possesses the property that $(y - y^*) \perp y^*$ (that is, y^* is the point on R minimally distant to y). Since $(y^* - y)'x = 0$, we have that $\gamma\|y\|^2 = \gamma x'x = y'x \leq \|y\|^2$. Therefore, $0 \leq \gamma \leq 1$ so that if such a y^* exists, it lies in B_0 and $\|F(y)\| = \|y^*\| \leq \|y\|$.

In the case that there is no point y^* on R such that $y^* \perp (y^* - y)$, it can easily be shown that $F(y) = 0$ since the set $-R$ must contain such a point. □

To prove that the origin is an asymptotically stable equilibrium point of (A.5) we apply Lyapunov's direct method. Let $V(z) = \|z\|^2$. We prove that $\|z(n+1)\| < \|z(n)\|$ for all $z(n) \neq 0$ by proving that $\|z(n)\| \geq \|F(y+z)\| > \alpha\|F(y+z)\| = \|z(n+1)\|$.

Theorem A.4. *Equation (A.5) has a globally asymptotically stable equilibrium point at $z = 0$.*

Proof. We prove the claim by contradiction. Assume that there exists some $z \neq 0$ such that $\|z\| < \|F(y+z)\|$ where $F(y) = 0$ (the proof is obvious for $z = 0$).

Let $z = \gamma y + y^{\perp}$ where γ is some real constant and $y^{\perp}$ is perpendicular to y. By Lemma A.3, we know that $\|z\| < \|F(y+z)\| \leq \|y+z\|$, and so we obtain that $\gamma > -1$ and $y^{\perp}$ is arbitrary.

Let W be the hyperplane separating y and M as described in Lemma A.1 which separates the space into the two regions R_1 and R_2. Assume $y \in R_1$; by the bound on γ, we have that $(y + z)$ must lie in the closure of R_1.

For any ray $R = \{\beta p \| \beta \geq 0\}$ where $p \in R_2$, the point x on R with minimum distance to $(y + z)$ is such that $x = 0$ or $(x - (y + z)) \perp x$ (this extends from the proof of Lemma A.3). If $x = 0$, then $\|z\| > F(y + z)$ and the proof is complete, so assume the latter.

It can be derived using the proof of Lemma A.1 that $x'x = (y+z)'x \leq (y^{\perp})'x$ (the proof is straight-forward in $\Re^2$, and it can be easily extended to $\Re^n$ because $(y + z)$ and R lie in a plane). Hence, $\|F(y + z)\| \leq \|y^{\perp}\| \leq \|z\|$, which is a contradiction.

The claim that $\|z\| \geq \|F(y + z)\|$ for all z is true. Therefore, for (A.5), $\|z(n + 1)\| \leq \alpha\|z(n)\| < \|z(n)\|$ for all $z(n) \neq 0$, proving that the system is asymptotically stable. □

Of course, at the equilibrium ($z_e = 0$), the output of (A.4) is $y_{opt} = Q_p(y)$, which is precisely what is desired.

A.4 Using Multiple Linearizations

The brake torque τ is a function of what is termed the indicated torque τ_{ind} (torque generated through combustion) summed with frictional and other losses.

$$\tau_{ind} = \left(\theta_a + \theta_b(\delta - \delta_{mbt})^2\right) W_f \tag{A.6}$$

θ_a, θ_b, and δ_{mbt} are affine functions of λ that depend on the combustion regime. The quadratic impact of the spark timing's deviation from the MBT spark timing on indicated torque (and, therefore, on brake torque) cannot be approximated well enough by any single linearization of (A.6). However, we can generate and partition multiple

Table A.1: Example partitioning for linearizations

Model #	1	2	3	4
Minimum deviation from MBT	0	5	10	15
Maximum deviation from MBT	5	10	15	∞

linearizations of (A.6) according the range of this deviation. For example, consider the partitioning in Table A.1 that can be used to generate four such linearizations.

Given a nominal input W_f^d and $W_t h^d$ (which gives a nominal MBT spark timing δ_{mbt}^d), different nominal values δ^d of the spark timing are chosen, one within each range of δ_{mbt}^d according to the partitions. Each selection of δ^d gives a different linearization of the DISC engine model.

Let $L \subset Z^+$ be the set of model references and assume there are N linear approximations ($|L| = N$). Denote the maximum deviation for model $l \in L$ as $\Delta_{max}(l)$ and the minimum deviation as $\Delta_{min}(l)$. When applying model generated by linearization l to determine a suitable control input, the spark timing is restricted to the range

$$\delta_{mbt} - \Delta_{max}(l) \leq \delta \leq \delta_{mbt} - \Delta_{min}(l) \tag{A.7}$$

We extend (A.1) to include multiple linearizations and denote the system obtained using the l^{th} linearization as

$$y(n) = C(l)\hat{p}_m(n) + D(l)u(n) \tag{A.8}$$

Determining the optimal control input requires a new MP-QP (which we will denote $\tilde{Q}_p$) that depends on the linearization being used to approximate the mode. $\tilde{Q}_p$ is exactly identical to Q_p with the exception that it takes as an additional argument the linearization to apply. In addition to using this model approximation, (A.7) is appended as an additional constraint in the program.

Extending the controller to using multiple linearizations requires the controller to both determine the linearization to use and the optimal input to apply for that

linearization. Of course, each of these decisions depends upon the other since the "best" linearization depends on the MBT spark timing which, in turn, depends on the linearization used to compute the optimal input u to the plant.

We resolve this difficulty by solving the problem in two steps. First, using just the reference and not the integrated error, a nominal optimal control input is computing from each linearization. Then, using the linearization that gives the best predicted performance, the control signal u that is applied to the plant is computed with the integrated error included. The new controller algorithm is presented below.

$$\begin{aligned}
&1.\ \text{For each } l \in L, \text{let } y_l = \tilde{Q}_p(y_{ref}, l) \\
&2.\ \text{Let } l^* = \arg\min_l \|y_{ref} - y_l\| \\
&3.\ x(n+1) = (1-\alpha)(y_{ref} - y(n)) + e_I(n) \\
&4.\ y'(n+1) = \tilde{Q}_p(y_{ref} + x(n+1), l^*) \\
&5.\ e_I(n+1) = y'(n+1) - y_{ref} \\
&6.\ u(n+1) = D^{-1}(l^*)\left(y'(n+1) - C(l^*)\,\hat{p}_m\right)
\end{aligned} \tag{A.9}$$

It should be noted that if the feedback information from x is used for determining the linearization in step 2, a "chattering" between multiple linearizations may result. Therefore, the nominal linearization to apply is chosen independently of the feedback error.

A.5 Simulations

For these simulations, we used four linearizations of the DISC engine model, and each linearization required approximately 20 regions in the 3D parameter space parameterized by p_m and y_{ref} for the MP-QP. We compare these results against the speed-gradient controller presented in [16], which is designed for the same application.

Speed-gradient (SG) control is a model-based Lyapunov design technique that relies on the minimization of a cost function J (a function of the plant's states) to drive

a system to a reference eqequilibriumoint. In an application of SG the DISC engine, J is constructed as the sum of a performance function J_p and a barrier function J_b. J_p penalizes excursions of certain engine parameters away from their nominal set-points so as to force the system to drive these parameters to eqequilibriumJ_b is designed to prevent constraint violations of the acactuatorsnd outputs by heavily penalizing the potential for any parameter to exceed its adadmissibleange. A novel derivation of the SG method and its application to the DISC engine as well as a numerical algorithm for checking the stability of the closed-loop system are presented in [16].

As noted previously, the MP-QP controller explicitly guarantees constraint satisfaction at every time step of the system without the use of a barrier function or a penalty on an actuator's deviation from its nominal, equilibrium value. Allowing the controller free range over all admissible control signals allows the closed-loop system to response more quickly to changes in reference without the risk of violating such constraints.

Figures A-2 and A-1 compare the two controllers during a purge operation. Both systems start in the stratified regime under torque-tracking mode and both switch combustion regimes when the intake-manifold pressure reaches approximately 50 kPa. When the purge signal is sent, the reference AFR decreases to its stoichiometric value 14.62:1, and the SG controller switches its mode to AFR-tracking mode (where greater emphasis is placed upon tracking the AFR). As can be seen in Figure A-2, the speed of the torque response significantly decreases as AFR tracking takes precedence.

In this simulation, the MP-QP controller is constantly kept in torque-tracking mode. Despite the emphasis on torque-tracking, the closed-loop system (which is guaranteed to asymptotically converge to the reference) achieves the desired stoichiometric AFR more quickly than its SG-controlled counterpart while allowing for the fast-tracking of torque references. Although there is a small deviation in the AFR from stoichiometry at time t = 1.7s when the reference torque changes, the deviation is minor and quickly resolved.

Notable in the MP-QP-controlled response is the overshoot in the torque response at time t = 1.7s. The overshoot is the result of a model-mismatch between the

nonlinear model and the linearized model used by the controller. Hence, the control signal that is optimally computed for the approximate system does not produce the expected output at the plant. The integrator, however, gradually accounts for the mismatch and, as the intake-manifold pressure is driven to its reference value, the error diminishes.

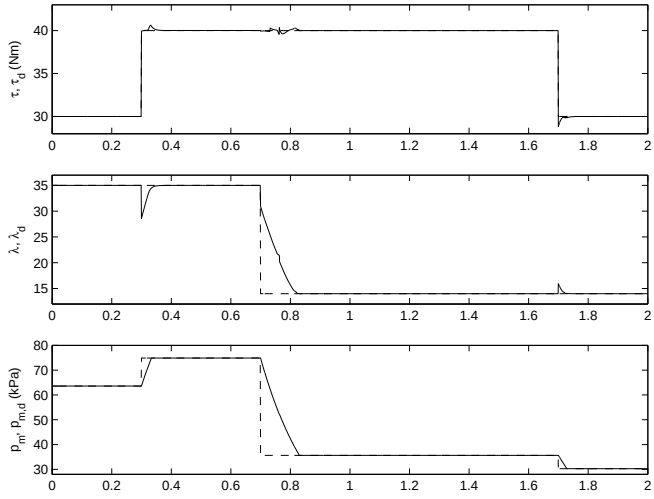


Figure A-1: MP-QP Controlled System

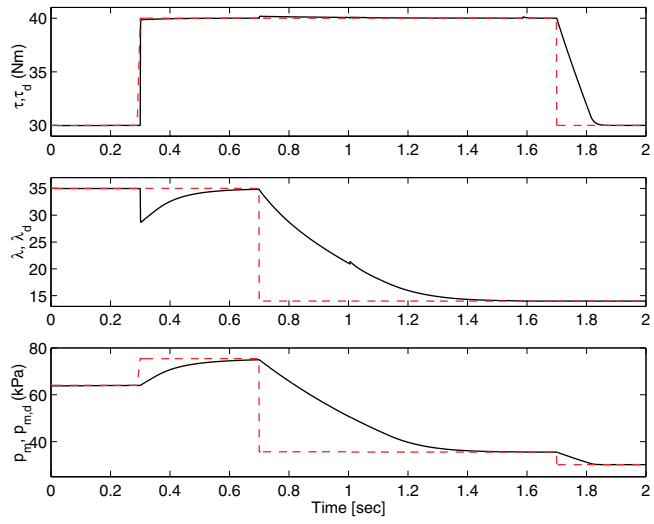


Figure A-2: SG Controlled System

Appendix B

Results Related to the Subsystem Controller

B.1 invertibility of $\Phi(t)$

The ODE relating Φ to W and Θ is

$$\dot{\Phi} + \Theta B_x W = 0 \tag{B.1}$$

where

- $\Theta(t) = W'(t)$
- $B_x = -B\Gamma^{-1}B'$, where $\Gamma = R + D'QD$, is negative-definite
- $A_\lambda = -A' - C'QD\Gamma^{-1}B'$

Therefore, $\Phi(t)$ is given by

$$\Phi(t) = \int_T^t W'(t)B\Gamma^{-1}B'W(t)dt \geq 0 \tag{B.2}$$

We directly apply the argument from [17] to our formulation and prove that $\Phi(t)$ is invertible by contradiction. If for any time $\tau < T$, $\Phi(\tau)$ is singular, then, by the continuity of the integrand, the integrand must be zero over the interval $[\tau, T]$.

Assume there is some vector $b \neq 0$ such that $b'\Phi(\tau)b = 0$. Then $B'W(t)b = 0$ for all $t \in [\tau, T]$. Letting $z(t) = W(t)b$, we get from the differential equation governing $W(t)$ that

$$\dot{W}(t)b = -A'W(t)b \Rightarrow \dot{z}(t) = -A'z(t) \tag{B.3}$$

where

$$\begin{aligned} z(T) &= b \\ B'z(t)b &= 0, \text{ for all } t \in [\tau, T] \end{aligned} \tag{B.4}$$

Combining the above constraints with the solution $z(t) = -e^{A'(t-T)}b$, we obtain

$$b' \int_T^\tau e^{A(t-T)} BB' e^{A(t-T)} B' dt b = 0 \tag{B.5}$$

which is a contradiction since the system is controllable. Hence, $\Phi(t)$ is invertible.

B.2 Positive Semi-Definiteness of $-B_\lambda$

Proving $-B_\lambda$ is positive semi-definite is tantamount to showing that $(Q - QD(R + D'QD)^{-1}D'Q) \geq 0$. For simplicty, we let $P = \sqrt{Q}D$ and equivently prove that

$$P(R + P'P)^{-1}P' \leq I \tag{B.6}$$

The proof of (B.6) is spanned over the three subcases presented in the following sections. In each, we denote the dimensions of P as $p \times m$ and the dimsensions of R as $m \times m$.

B.2.1 P is Invertible

If P is invertible, we can easily simplify (B.6) to

$$\begin{aligned} &\left((P')^{-1}RP^{-1} + (P')^{-1}P'PP^{-1}\right)^{-1} \leq I \\ \Longleftrightarrow &\sigma_{min}\left((P')^{-1}RP^{-1} + I\right) \geq 1 \\ \Longleftrightarrow &\sigma_{min}\left(M + I\right) \geq 1 \end{aligned} \tag{B.7}$$

Let the vector v correspond to the minimum singular value of $(M + I)$, which is the same as the minimum eigenvalue since the expression is symmetric. Expanding (B.7) by multiplying by v, we get

$$\sigma_{min}\left(M + I\right)^2 = v'M'Mv + v'MIv + v'Iv \geq v'Iv \tag{B.8}$$

which is true since M is positive semi-definite.

B.2.2 P is Square and Non-invertible

For the case of P square and non-invertible, we let $\tilde{P} = (P + \epsilon UV^T)$ where U and V^T correspond to the singular value decomposition of P ($P = U\Lambda V^T$). Therefore, for all $\epsilon > 0$, $\tilde{P}$ is invertible and so the previous result gives that

$$\tilde{P}(R + \tilde{P}'\tilde{P})^{-1}\tilde{P}' = (\tilde{M} + I)^{-1} \leq I \tag{B.9}$$

Taking the limit as $\epsilon \to 0$ gives (B.6) since the singular values of the $(\tilde{M}+I)^{-1}$ (which are equal to the eigenvalues since the matrix is symmetric) change continuously as a function of $\tilde{M}$, which in turn changes continuously as a function of ϵ.

B.2.3 P is not Square

To resolve the case of P not square, we simply append P with an appropriately-sized 0-matrix to make it square. We then apply the previous results to conclude that (B.6) is true.

Consider the case of $p < m$ (P is "wide"). Let $\hat{P} = \begin{bmatrix} P & 0 \end{bmatrix}^T$ be a square matrix of dimension $m \times m$. Because $\hat{P}$ is square, we know that

$$\hat{P}(R + \hat{P}'\hat{P})^{-1}\hat{P}' \leq I \tag{B.10}$$

Therfore, we only need to show that this expression is equivelent to (B.6). Simple

substition and basic algebra reveal this to be true

$$\begin{aligned}
&\hat{P}(R+\hat{P}'\hat{P})^{-1}\hat{P}' \\
&\quad= \begin{bmatrix} P \\ 0 \end{bmatrix} \left(R + \begin{bmatrix} P' & 0 \end{bmatrix} \begin{bmatrix} P \\ 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} P' & 0 \end{bmatrix} \\
&\quad= \begin{bmatrix} P \\ 0 \end{bmatrix} (R+P'P)^{-1} \begin{bmatrix} P' & 0 \end{bmatrix} \\
&\quad= \begin{bmatrix} P(R+P'P)^{-1}P' & 0 \\ 0 & 0 \end{bmatrix} \leq I
\end{aligned} \tag{B.11}$$

Of course, the above inequality is true if and only if (B.6) is true.

For the case of $p > m$ (P is "tall"), we need to manipulate R as well. Let $\hat{P} = \begin{bmatrix} P & 0 \end{bmatrix}$ be a square matrix of dimension $p \times p$, and let $\hat{R} = \begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix}$ also be a matrix of dimension $p \times p$.

Because $\hat{P}$ and $\hat{R}$ are each square, we have

$$\hat{P}(\hat{R}+\hat{P}'\hat{P})^{-1}\hat{P}' \leq I \tag{B.12}$$

Once again, we need to show the above inequality is equivalent to (B.6). Simple substitution and simplification give that

$$\begin{aligned}
&\hat{P}(\hat{R}+\hat{P}'\hat{P})^{-1}\hat{P}' \\
&\quad= \begin{bmatrix} P & 0 \end{bmatrix} \left(\begin{bmatrix} R & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} P' \\ 0 \end{bmatrix} \begin{bmatrix} P & 0 \end{bmatrix} \right)^{-1} \begin{bmatrix} P' \\ 0 \end{bmatrix} \\
&\quad= \begin{bmatrix} P & 0 \end{bmatrix} \left(\begin{bmatrix} R+P'P & 0 \\ 0 & I \end{bmatrix} \right)^{-1} \begin{bmatrix} P' \\ 0 \end{bmatrix} \\
&\quad= P(R+P'P)^{-1}P' \leq I
\end{aligned} \tag{B.13}$$

The two expressions are equivalent and, therefore, for general P, (B.6) is true.

B.3 Convergence of Matrix Parameters

The solution S_∞ to the ARE (3.28) "stabilizes" the matrix A_x in that $(A_x + B_x S_\infty) < 0$. Furthermore, the approximation of $S(t)$ as S_∞ for small t allows us to approximate the solution for $\Theta(t)$ governed by (3.26). Specifically, for some large final time $(\tau + T)$ (where the signifigance of τ is given in 3.5.1), we approximate $\Theta(t)$ as follows

$$\begin{aligned}\Theta(t) &= \exp\left[\int_t^{\tau+T} A_x + B_x S dt\right] \\ &\approx \exp\left[\int_t^{T} A_x + B_x S_\infty dt + \int_T^{\tau+T} A_x + B_x S dt\right] \\ &= \exp\left[\int_t^{T} A_x + B_x S_\infty dt\right] \exp\left[\int_T^{\tau+T} A_x + B_x S dt\right]\end{aligned} \tag{B.14}$$

For any $T \geq 0$ (corresponding to the final time $(T + \tau)$ from which $S(t)$ is computed backwards), the exponential on the right is a constant whereas the exponential on the left is comparitively easy to compute. More importantly, by the negative definiteness of $(A_x + B_x S_\infty)$, small t and large T will yeild $\theta(t) \approx 0$. The same may be said of $W(t)$ since $W = \Theta'$.

It is further apparent by the ODE governing Φ that a large final time yields $\dot{\Phi}(t) \approx 0$ for small t, so that $\Phi(t)$ approaches a constant as $t \to -\infty$.

To prove $Z(t)$ converges as $t \to -\infty$, we examine the dynamics of the unit solutions $x_i(t) = Z(t)e_i$, where $[e_i]_j = 1$ if $i = j$ and 0 otherwise. We then write $Z(t) = \begin{bmatrix} x_1(t) & x_2(t) & \dots & x_n(t) \end{bmatrix}$. For each i, the dynamics of $x_i(t)$ for small t are given by

$$\begin{aligned}\dot{x}_i(t) &\approx -(A'_x + S_\infty B_x)x_i(t) + (-S_\infty E_x + E_\lambda)e_i \\ &= \tilde{A}x_i(t) + u_i\end{aligned} \tag{B.15}$$

where u_i is a constant. Because the dynamics of (B.15) are integrated backwards in time, the positive-definiteness of $\tilde{A}$ yields that the system is stable, implying that $x_i(t)$ will converge to some constant.

Since, for small t, $W(t)$ is zero and $Z(t)$ is a constant, $\Sigma(t)$ converges to a constant as well.

Bibliography

[1] A Bemporad, D Mignone, M Morari. "An efficient branch and bound algorithm for state estimation and control of hybrid systems". *Proceedings of the European Control Conference*, 1999

[2] A. Bemporad, M. Morari. "Control of systems integrating logic, dynamics, and constraints". *Automatica*, Vol. 35: 407-427, 1998.

[3] A. Bemporad, M. Morari. "Optimal controllers for hybrid systems: stability and piecewise linear explicit form". Technical Report AUT99-16. Autmatic Control Laboratory, ETH Zurich, Switzerland, 1999.

[4] A. Bemporad, M. Morari, V. Dua, E. N. Pistikopoulos. "The explicit linear quadratic regulator for constrained systems". *Automatica*, Vol. 38(1): 3-20, 2002.

[5] A. Bemporad, N. Giorgetti, I.V. Kolmanovsky, D. Hrovat. "A hybrid approach to modelling and optimal control of DISC engines". *Proceedings of the 41st IEEE Conference on Decision and Control*, Vol. 2:1582-1587, December 2002.

[6] A. Bemporad, A. Guia, C. Seatzu. "Synthesis of state-feedback optimal controllers for continuous-time switched linear systems". *Proceedings of the 41st IEEE Conference on Decision and Control*, pp. 3182-3187, December 2002.

[7] M. Branicky, V. Borkar, S. Mitter. "A unified framework for hybrid control: model and optimal control theory". *IEEE Transactions on Automatic Control*, Vol. 43(1): 31-45, January 1998.

[8] M. Branicky. "Multiple Lyapunov functions and other analysis tools for switched and hybrid systems". *IEEE Transactions on Automatic Control*, Vol. 43(4): 475-482, April 1998.

[9] A.E. Bryson, Y.C. Ho. "Applied Optimal Control". New York: John Wiley and Sons, 1975.

[10] G. Davrazos, N. T. Koussoulas. "A review of stability results for switched and hybrid systems". *Mediterranean Conference on Control and Automation*, 2001.

[11] N. H. El-Farra, P. Christofides. "Switching and feedback laws for control of constrained switched nonlinear systems". *Proceedings of the 5th International Workshop on Hybrid Systems: Computation and Control*, Lecture Notes in Computer Science, 2002.

[12] E. Frazzoli, M. Dahleh, E. Feron. "Real-time motion planning for agile autonomous vehicles". *Proceedings of the American Control Conference*, Vol. 1:43-49, June 2001.

[13] E. Frazzoli, M. Dahleh, E. Feron. "Robust hybrid control for autonomous vehicle motion planning". *Proceedings of the 39th IEEE Conference on Decision and Control*, Vol. 1:821-826, December 2000.

[14] A. Giua, C. Seatzu, C. Van Der Mee. "Optimal control of switched autonomous linear systems". *Proceedings of the 40th IEEE Conference on Decision and Control*, pp. 2472-2477, December 2001.

[15] S. Hedlund, A. Rantzer. "Optimal control of hybrid systems". *Proceedings of the 38th IEEE Conference on Decision and Control*, pp. 3972-3976, December 1999.

[16] I.V. Kolmanovsky, M. Druzhinina, J. Sun. "Speed-gradient approach to torque and air-to-fuel ratio Control in DISC Engines". *IEEE Transactions Control Systems Technology*, Vol. 10(5): 671-678, September 2002.

[17] W.E. Schmitendorf, S.J. Citron. "On the applicability of the sweep method to optimal control problems". *IEEE Transactions on Automatic Control*, pp. 69-72, 1969.

[18] J. Sun, I.V. Kolmanovsky, D. Brehob, J. Cook, J. Bruckland, M. Haghgooie. "Modelling and control problems for gasoline direct injection stratified charge (DISC) engines". *Proceedings 1999 IEEE Conference Control Applications*, pp. 471-477, 1999.

[19] X. Xu, P. Antsaklis. "Optimal control of switched systems based on a parameterization of switching instants". *Proceedings of the IFAC World Congress*, 2002.