Suboptimal Control of Switched Systems with an Application to the DISC Engine

Michael Rinehart, Munther Dahleh, *Fellow, IEEE*, Dennis Reed, and Ilya Kolmanovsky, *Senior Member, IEEE*

***Abstract*— The growing stringency of fuel economy, emissions, and drivability requirements has led to proliferation of powertrain systems which have multiple discrete operating modes. Systematic approaches to the development of optimal and robust control systems for such powertrains is needed to contain their increased development times and costs. In this paper, we propose a new approach for controlling switched systems that is applicable to powertrain systems with multiple operating modes. Our approach reduces the complexity of computing a control law to a linear programming problem defined over a finite number of states in each operating mode. The methodology is designed to be suitable for practical applications while, under appropriate conditions, providing near-optimal performance. An application to the direct injection stratified charge engine with two distinct operating modes is given to illustrate our approach.**

This unabridged version contains an additional proof for a claim contained in the proof of Theorem 2.

***Index Terms*— Powertrain Control, Switched Systems, Optimal Control, Direct Injection Engines**

I. Introduction

THE use of advanced powertrain systems with multiple operating modes is growing increasingly widespread as the demands on fuel economy, emissions, drivability, and safety of passenger vehicles become more stringent. Examples of advanced systems of this kind include Variable Displacement engines, Direct Injection Stratified Charge (DISC) engines, Variable Compression Ratio engines, Homogeneous Charge Compression Ignition engines, and Hybrid Electric Vehicles. In each of these systems, a finite set of operating modes are employed to provide flexibility in meeting these diverse requirements.

The control systems for such powertrains must determine an optimal operating mode and transition to that mode without disturbing the driver, all while accurately delivering requested torque and air-to-fuel ratio. Field experience suggests that fuel economy and drivability can fall short of expectations if the mode selections or transitions are not optimally performed, but the general lack of systematic design techniques for this task often leads to the use of ad-hoc approaches that can result in degraded performance. In this paper we develop a methodology that attempts to address the above difficulty.

Abstracting from specific considerations related to individual powertrain systems, the broader focus of this paper is on the control of switched systems possessing controllable subsystems. Existing approaches for the control of switched systems applicable to this class of systems include the use of Control Lyapunov Functions [1], control derived through an approximation of the optimal value function [2], the standard optimal control framework applied to a reparameterization of the system [3], and receding horizon control (RHC).

In [1], performance is not a factor in determining when to switch between modes, and, rather, it is determined only when switching will lead to stability, which is not suitable for our performance-based setting. Though [2] and [3] present performance-centric frameworks for switched-system control, the former does not guarantee the stability of the closed-loop system, and the latter is not readily applicable to low-resource embedded hardware.

An application of RHC to the DISC engine is presented in [4], [5]. Because the multiparametric optimization technique used in these works requires an affine system model, the engine model is linearized with the engine speed treated as a constant parameter. A detailed comparison of the approach presented in this paper with the one in [4], [5] is reserved for future publications.

Rather than strive for global optimal control of this class of systems, we follow the approach taken in [6] where it is shown that high-performance, practical controllers can be constructed for a specific class of hybrid systems by parameterizing the controller actions by a finite set of actions. In this paper, we extend this approach to the control of controllable switched systems by constraining the switching portion of the control input and fixing the feedback controllers for each mode. We show that, under reasonable assumptions, the resulting system is guaranteed to converge to a reference signal while providing meaningful performance. These theoretical developments are described next, while an application to the DISC engine in presented in Section VI.

II. Background and Assumptions

A. Definitions

Consider a switched system

$$\dot{x}(t) = f_{i(t)}\left(x(t), u(t)\right) \tag{1}$$

where $x \in \Re^n$, $u \in \Re^m$, and $i : \Re \to Q$ is a piecewise-constant function continuous from the right. The set of *modes* for the switched system is a finite set $Q = \{1, \ldots, N\}$. Our objective will be to control the state of (1) by adjusting $u(t)$ and $i(t)$ so that $x(t)$ converges to 0 asymptotically while $i(t)$ becomes equal to the desired mode.

Michael Rinehart and Munther Dahleh are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA.

Dennis Reed and Ilya Kolmanovsky are with Ford Research and Advanced Engineering, Ford Motor Company, Dearborn, MI.

This research was sponsored in part by Ford Research and Advanced Engineering.

We assume the system state x is constrained to a fixed, closed set X_a when the system is in mode a; that is, if $i(t) = a$, then $x(t)$ must lie in X_a. X_a may be given as a physical constraint of the system, or it may be imposed artificially by the designer to reflect the desired state behavior. We similarly assume that the control u is constrained to a set U_a when $i(t) = a$, a constraint that also may be given or artificial.

We define a special subset of modes Q_0 as the set of modes a where the origin is an admissible state; that is, Q_0 is the set of modes a such that $0 \in X_a$. We assume that Q_0 is not empty.

Finally, we assume that (1) in a fixed operating mode $i(t) = a$ is a controllable system with respect to any pair of states in X_a.

We now define several important notations used throughout the paper:

- let $t_0 = 0$ and successively define the k^{th} *switching instance* t_k as the first time $i(t)$ changes value since time t_{k-1}, i.e. $t_k = \min_{t>t_{k-1}} \{t \mid i(t^-) \neq i(t)\}$
- define $x_k = x(t_k)$ as the k^{th} *switching state,*
- and define $a_k = i(t_k)$ as the k^{th} *operating mode* and denote the *mode sequence* as the list $(a_0, a_1, \ldots)$.

If the mode becomes a constant after some switching time t_k, i.e. $i(t) = a$ is constant for $t \geq t_k$, then as there are no more switches, we define $t_j = \infty$, $a_j = a$, $x_j = 0$ for all integers $j \geq k$.

We treat both u and i as design parameters in our system, and we view (1) as a collection of N subsystems. The *subsystem control* u guides the system in a fixed operating mode, and the *mode control* i determines the operating mode to apply.

B. Assumptions

For certain classes of systems that follow the form of (1) in a fixed mode $i(t) = a$, there already exist tools for constructing stabilizing controllers. The controller architecture in this paper attempts to leverage these techniques in order to simplify the overall controller design. Let u_a be such a controller for (1) under a fixed $i(t) = a$. For u_a to be applicable to our framework, it must satisfy the following assumption.

Assumption 1: For any initial state x_0 and target state x_d in X_a, there exists some time $T_a = T_a(x_0, x_d)$ such that system (1) *with $i(t) = a$, $x(0) = x_0$, and $u(t) = u_a(x(t), x_d)$ satisfies*

1) *admissibility of the state trajectory: $x \subset X_a$,*
2) *asymptotic stability: the closed-loop system is locally asymptotically stable about x_d,*
3) *and controllability: $x(t) = x_d$ for all $t > T_a$ if $T_a < \infty$, or $x(\infty) \in \{x_d, 0\}$ if $T_a = \infty$.*

where we define $x(\infty) = \lim_{t\to\infty} x(t)$.

Allowing $x(\infty)$ to be 0 is a technical assumption that will be made clearer in the case of optimal subsystem controllers. We term u_a the *subsystem controller for mode a* and term T_a the *tracking time*.

Finally, let $J_a(x_0, x_d)$ be the cost of applying u_a according to

$$J_a(x_0, x_d) = \int_0^{T_a} L\left(x, u_a(x, x_d)\right) dt \qquad (2)$$

subject to $x(0) = x_0$ and $i(t) = a$. $L(x, u)$ is a positive-definite, monotonically-increasing (in the norm of each argument) function. We only integrate up to T_a because we allow the system to stop accumulating cost once x_d or 0 are reached (if either is reached).

C. "Optional" Assumptions

Though u_a need not be an optimal controller, assumptions on the optimality and continuity of the cost function will be useful in obtaining some theoretical results.

Assumption 2: J_a and L are continuous functions of their arguments and are finite.

Assumption 3: u_a minimizes (2) *over all inputs u satisfying Assumption 1.*

Assumptions 2 and 3 lead us to the following results concerning the tracking time T_a.

Proposition 1: Let Assumptions 1-3 hold. If $a \notin Q_0$, then $T(x_a, x_d)$ is finite.

Proof: If $X_a \notin Q_0$, then, by the positive-definiteness and monotonity of L and the fact that X_a is closed and does not include 0, $L(x, 0)$ takes a minimum value $M > 0$ in X_a so that $L(x, u) \geq M$ for all u and all $x \in X_a$. Therefore, if $T_a = \infty$, then $J_a = \infty$, which contradicts Assumption 2. ■

Corollary 1: Let Assumptions 1-3 hold. If $T_a(x_0, x_d) = \infty$ for some x_0 and x_d, then $a \in Q_0$ and $x(\infty) = 0$.

Proof: By Proposition 1, $T_a = \infty$ implies $0 \in X_a$. Assume $x_d \neq 0$. If $x(\infty) \neq 0$, then $x(\infty) = x_d$ and so there exists a τ such that $x(\{t > \tau\})$ does not intersect a neighborhood of the origin. It can be shown by a proof similar to that of Proposition 1 that this implies $J_a = \infty$. Therefore, $x(\infty) = 0$. ■

III. Suboptimal Control of Switched Systems with Bounded Switching Regions

In this section, we consider the problem of controlling (1) subject to the constraint that a switch between two modes must occur in a bounded region of the state space. Specifically, if the system is switching from mode a to mode b at time t, then $x(t)$ must lie in the set X_{ab} where $X_{ab} \subset X_a \cap X_b$, and X_{ab} is assumed to be bounded. Switched systems with constrained state spaces, which occur frequently in practice, are an example of this class of systems.

Using preconstructed subsystem controllers $\{u_a\}$ that each satisfy Assumption 1, we seek to provide a suboptimal solution for the following problem: given an initial state $x(0) = x_0$ and initial operating mode $i(0) = a_0$, determine a control function pair (u, i) that minimizes

$$J\left(a_0, x_0\right) = J\left((a_0, x_0), (u, i)\right) = \int_0^{\infty} L(x, u) dt \qquad (3)$$

A. Switching States and the SRHSG

The basic idea behind the controller architecture we develop in this section is to limit the system to switching at a finite set of states, using the subsystem controllers $\{u_a\}$ to optimally travel among such states before tracking the origin.

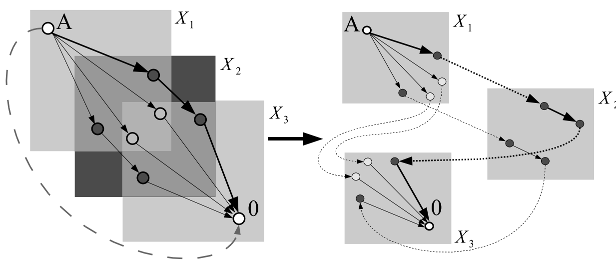

Fig. 1. Illustration of various trajectories originating at state $A \in X_1$ in mode 1 and terminating $0 \in X_3$ in mode 3. Solid lines represent the physical transition of the state, and dotted lines represent a switch to the next mode, with the state fixed at the time of the switch. The trajectory must pass through one of the target states (in S_{13} or $S_{12} \to S_{23}$) in order to switch modes. The thick trajectory depicts a sample trajectory in each representation.

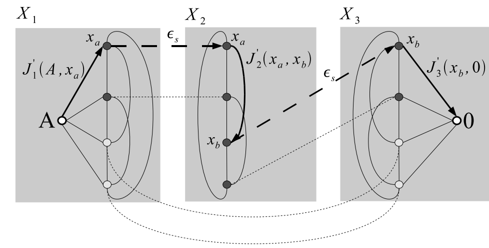

Fig. 2. Illustration of the SRHSG corresponding to the target states and modes of Figure 1. All target states of the same mode have edges among them, and an edge exists between the same state in two different modes. The trajectory of Figure 1 is highlighted in this figure, and the weight associated with each edge in this trajectory is labeled. Note that the edges are only drawn bi-directionally for clarity, and in fact two edges, one in each direction, should be substituted for each bi-directional edge since each of these edges will have a different weight.

First, choose a positive constant $r_s > 0$, termed the *switching radius*. Let $S_{ab} \subset X_{ab}$ be a chosen set of *target states* between modes a and b with the property that every two unique target states $x_1, x_2 \in S_{ab}$ are separated by a distance $2r_s$, i.e. $\|x_1 - x_2\| > 2r_s$. The target states in S_{ab} are the only states in X_{ab} at which we allow the system to switch modes; that is, if $i(t^-) = a$ and $i(t) = b$, then we must have $x(t) \in S_{ab}$[1]. By the assumption that X_{ab} is bounded, S_{ab} is finite.

We are also interested in knowing all of the target states in mode a. To this end, we define $S_a = \bigcup_b S_{ab}$, which is also finite.

We limit how the system travels among the target states by defining a directed graph that defines all of the possible trajectories the system may take.

Definition 1: For a given initial mode a_0 and state x_0, the *static robust hybrid switching graph* (SRHSG) is defined as the graph $G = (V, E)$ where the vertices V and edges E satisfy

1) every vertex $v \in V$ is a pair (a, x) where either 1) a is any mode and $x \in S_a$ in a target state of mode a, 2) $a \in Q_0$ and $x = 0$ is the origin, or 3) it is the initial pair $(a, x) = (a_0, x_0)$,
2) and every edge $e \in E$ is a pair $(v_1, v_2) = ((a, x), (b, y))$ where either 1) the mode is fixed and we travel between different target states of that mode: $a = b$, $x \in S_a$, and $x \neq y \in S_a \cup \{0\}$; or 2) the state is the same and the mode changes: $x = y \in S_{ab}$ and $a \neq b$.

An ordered pair $(v_1, v_2) \in E$ indicates a directed edge from v_1 to v_2.

The SRHSG serves to limit the possibilities of control to a finite set of actions. Nodes represent states at which the system may decide to switch modes or track another state while in the same mode. Edges represent 1) tracking if the state changes and the mode remains the same, or 2) mode transitioning in the form of switching if the state remains the same and the mode changes.

We note that we added the initial state (a_0, x_0) and the destination state $(b, 0)$ (over all modes $b \in Q_0$) as nodes in the SRHSG, but not as target states that the controller can use to change modes. For example, for a two mode system, there is no edge from the origin in mode 1 (vertex $(1, 0)$) to the origin in mode 2 (vertex $(2, 0)$).

[1]Actually, we will only require that $x(t)$ is "close" to a target state.

Of course, from a given node, the choice of the edge to traverse should not be arbitrary, and so we now impart costs on the edges to resolve this difficulty.

We define a revised cost $J'_a(x_m, x_n)$ as the cost of applying u_a to track x_n from x_m according to

$$J'_a(x_m, x_n) = \begin{cases} \infty, & x(\infty) \neq x_n \\ \int_0^{T_a} L\left(x, u_a(x, x_n)\right)\left(1 - 1_B(x)\right) dt, & x(\infty) = x_n \end{cases} \tag{4}$$

subject to (1), $x(0) = x_m$, and $i(t) = a$. Here, B is the ball of radius r_s about x_n, and 1_B is the indicator function of the set B ($1_B(x) = 1 \Leftrightarrow x \in B$). Basically, the cost ceases to accumulate once $x(t)$ is "close" to x_n in the sense that it is within a distance r_s of x_n.

We now define a weighting function $w : E \to \Re$ for the edges on G by

$$w\left((a_m, x_m), (a_n, x_n)\right) = \begin{cases} \epsilon_s, & a_m \neq a_n \\ J'_{a_m}(x_m, x_n), & a_m = a_n \end{cases} \tag{5}$$

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

Essentially, an edge for which x_n cannot be tracked from x_m (because the origin is tracked instead) is given weight ∞, which is necessary to prevent the system from attempting to transition to vertex (a, x_n) if it actually tracks vertex $(a, 0)$ instead.

B. Computing the Switching Path

For a given initial state x_0, initial mode a_0, and SRHSG G, consider the class of control pairs (u, i) that yield trajectories x and switching sequences $(a_0, a_1, \ldots)$ having the following properties:

1) each switching state is "close" to a target state: if $x_k \neq 0$ is a switching state, there is a vertex (a_k, x'_k) in G such that $\|x_k - x'_k\| \leq r_s$,
2) two consecutive non-zero target states are always unique: $x'_{k-1} \neq x'_k$ for $t_k < \infty$, and
3) u_a always tracks target states of the current mode: for $t_k < t < t_{k+1}$, we have $u(t) = u_{a_k}(x(t), x'_{k+1})$.

Because infinite-time tracking is allowed, the first condition yields switching in finite time by allowing the system to switch when the state is within the switching radius of a target state in G [2]. The second condition prevents the system from switching at the same state consecutively. Finally, the last requirement states that the controller may only track target states in G, even though the resulting switching state can vary slightly from this state.

Before detailing the computation of an optimal (u, i), we ascertain some useful properties of the established framework.

Proposition 2: Fix the set of target states S_{ab} between all modes a, b. For all initial pairs (a_0, x_0) and corresponding SRHSGs G, there is a positive time ΔT between switches after the first switch; that is, $t_{k+1} - t_k > \Delta T$ for all $k > 1$.

Proof: By the finiteness of the number of modes and target states, there exists a minimum positive tracking time

$$\Delta T = \min_{a \in Q, x_m, x_n \in S_a, x_m \neq x_n} T_a(x_m, x_n)$$

Clearly, $0 < \Delta T \leq (t_{k+1} - t_k)$ for $k > 1$, and it is independent of the initial pair. ∎

The existence of a minimum dwell time (after the first switch) is interesting because it precludes the possibility of generating an infinite number of switches in finite time, even if the switching penalty is set to zero.

Proposition 3: Any control pair (u, i) resulting in a finite cost J must result in a finite number of switches.

Proof: If there are an infinite number of switches, then $J > \sum_{k=1}^{\infty} \epsilon_s = \infty$, which is a contradiction. ∎

Lemma 3 is important because it asserts the existence of a final operating mode. If we constrain the system to terminate in a final operating mode b, then computing an optimal sequence of vertices to track from (a_0, x_0) to $(b, 0)$ is simply a matter of applying a shortest-path algorithm, which conventionally reduces to a linear programming problem [7]. This optimal list of vertices, termed the *switching path*, is the acyclic, finite path of least cost between these vertices [7]. We write the switching path as a tuple of pairs $((a_0, x_0), (a_1, x_1), \ldots, (a_N, x_N), (b, 0))$ for some N.

We note that by appending the SRHSG with a terminating node that connects without cost to all nodes of the form $(b, 0)$, the system will always terminate in an optimal final mode. In this paper, we do not append the SRHSG with a terminating node because, in some applications, it may be desirable to terminate in a pre-specified operating mode. To this end, we assume that the final operating mode b is specified. If the final mode is not a constraint, then we assume that b is optimal.

C. Computing and Applying the Control Law

Application of the SRHSG methodology in practice is fairly straight-forward, and we give a brief review of the algorithm here. First, construct a partial SRHSG G that contains only target states S_{ab} and edges E that appropriately connect them. Then, given an initial mode and state $v_0 = (a_0, x_0)$ and a final mode and state $v_f = (b, 0)$, append the G with these vertices, compute the edge weights using w, and find the optimal switching path $((a_0, x_0), (a_1, x_1), \ldots, (a_N, x_N), (b, 0))$ using a shortest-path algorithm.

After the switching path is known, the control (u, i) is computed as follows. Apply the subsystem controller u_{a_0} to track the first target state x_1 in the switching path. Once x_1 is tracked, switch to the next specified mode a_1 and track the subsequent target state x_2. Repeat this process until the origin is asymptotically tracked in the final operating mode b. An example for computing the SRHSG offline and efficiently storing the switching path in memory for online reference will be presented later in the DISC engine application.

D. Stability and Robustness

Though a closed-loop system controlled using the SRHSG scheme results in the state converging to the origin, it may not be Lyapunov stable. For example, consider a switched system with two modes controlled using a single target state. For a mode switch to occur, the system must track that target state, regardless of how close the initial state is to the origin.

An applicable notion of stability in the context of SRHSG can be given as follows.

Definition 2: System (1) with state x and pair (u, i) is *stable along a switching path* $(x_0, x_1, x_2, \ldots)$ if $x(t) \to 0$, and, for each k, the system given by (1) with state $\hat{x}$, fixed mode $\hat{i}(t) = a_k$, and feedback control $\hat{u}(t) = u_a(\hat{x}(t), x_{k+1})$ is locally asymptotically stable with respect to x_{k+1}.

In general, one can prune edges from the SRHSG so as to only allow transitions that satisfy Definition 2.

Because SRHSG effectively fixes the switching path, the above definition essentially asserts that the robustness of the SRHSG methodology is tantamount to the robustness of the subsystem controllers and the size of the switching regions. Smaller switching regions allow for more accurate tracking, which yields higher performance when model uncertainty and disturbances are negligible. As long as the switching regions about the target states are tracked, convergence is guaranteed.

E. Convergence with Optimal Subsystem Controllers

In this section, we show how uniformly increasing the density of the target states impacts the overall control law when the subsystem controllers are optimal. Though the results of this section do not impact a practical application of SRHSG, they do justify its use in an optimal control setting.

Let Assumptions 1-3 hold and set the switching radius $r_s = 0$ [3]. For an initial pair (a_0, x_0), let Ω^* be the set of all control pairs (u^*, i^*) satisfying

1) switching states are admissible: $x_k^* \in X_{a_{k-1}^* a_k^*}$,
2) no Zeno effects: $t_k^* \to \infty$,
3) optimal tracking of switching states: for $t_k^* < t < t_{k+1}^*$, $u^*(t) = u_{a_k^*}(x^*(t), x_{k+1}^*)$,
4) finite cost: $J^* < \infty$,
5) and the system tracks the origin if that is optimal: if $J_{a_k^*}(x_k^*, x_{k+1}^*) \geq J_{a_k^*}(x_k^*, 0)$, then $x_{k+1}^* = 0$

[2]This motivates a choice for r_s that is small enough to allow for accurate tracking but not so small that tracking is made unnecessarily difficult by the effects of noise.

[3]We can set $r_s = 0$ because optimality guarantees finite-time tracking between non-zero target states connected by a finite-weight edge.

where x^* is the trajectory resulting from applying the control pair, (t^*_k) are the switching instances, (a^*_k) are the modes, and J^* is the cost. The last condition prevents the system from tracking a non-zero state if tracking the origin results in the same or a better cost. The condition is hardly restrictive as it merely removes some non-optimal controls. Consequently, if $x^*_N = 0$, then $x^*_k = 0$ for all $k > N$. Also, it is important to note that no predefined target states are involved in these control laws.

Although we do not make any statements about the existence of an optimal control law in Ω^*, we do prove that there exists a SRHSG that yields a performance that can arbitrarily approximate or even surpass the quality of any control pair in Ω^*.

First, we construct a sequence target state sets that become increasingly dense. Let $(S^j_{ab})_j$ denote a sequence of target states between modes a and b with the following properties:

1) increasing density: for each $x \in X_{ab}$, there exists a $x_s \in S^j_{ab}$ such that $\|x - x_s\| < 1/j$.
2) target states are not removed upon refinement: $S^{j+1}_{ab} \supset S^j_{ab}$

Denote the corresponding SRHSG sequence as $(G^j)_j$.

Theorem 1: Given a control pair $(u^*, i^*) \in \Omega^*$ that results in a cost J^*, there exists a sequence of control pairs $\big((u^j, i^j)\big)$, corresponding to the sequence $(G^j)_j$, yielding costs (J^j) such that $J^j \to J^*$.

The proof of Theorem 1 as well as the proof of corollary below are provided in Appendix.

Corollary 2: Given a control pair $(u^*, i^*) \in \Omega^*$ that results in a cost J^*, there exists a sequence of control pairs $\big((u^j, i^j)\big)$, corresponding to $(G^j)_j$, yielding costs (J^j) such that $J^j \to J \le J^*$ with strict inequality if (u^*, i^*) is non-optimal.

Noteworthy in the proof of Corollary 2 is that by simply increasing the density of the target states, the resulting control law will eventually satisfy these conditions.

IV. Suboptimal Control of Homogeneous Switched Systems with Unbounded Switching Regions

In this section, we relax the constraint that a switch must occur in a bounded region of the state space while maintaining the objective to control mode switching for optimal performance. Such a scenario is typical if the individual modes correspond to multiple fixed structure controllers.

Of course, the application of a finite number of target states in an unbounded set is not sufficient to cover the set. We mitigate this issue by imposing the following properties on f_a and L.

1) $f_a(\alpha x, \alpha u) = \alpha f_a(x, u)$
2) $L(\alpha x, \alpha u) = \|\alpha\|^z L(x, u)$ for some $z > 1$.

Switched linear systems with quadratic performance costs fall into this class of systems with $z = 2$. We must also apply the following assumption about the subsystem controllers.

Assumption 4: For all a, $u_a(\alpha x_0, \alpha x_d) = \alpha u_a(x_0, x_d)$ and $T_a(\alpha x_0, \alpha x_d) = T_a(x_0, x_d)$.

If the subsystems are homogeneous then it is fairly natural to expect the subsystem controllers to be homogeneous as well. Therefore, we let this assumption hold for the remainder of the section.

Using such preconstructed controllers, we seek to provide a suboptimal solution for the following problem: given an initial state x_0 and initial operating mode a_0, find a control pair (u, i) that minimizes (3).

A. Switching States and the DRHSG

Though the controller architecture in this section applies the same idea of allowing switching at only a finite set of states, we limit our choice of target states to within a bounded set of the origin. We then leverage homogeneity to scale the states to provide an infinite set of target states while using a finite amount of memory.

Let C be the unit sphere in $\Re^n$ and choose a set of distinct states $S_C = \{s_0, s_1, s_2, \ldots, s_L\}$ that satisfy

1) $s_0 = 0$ and $s_k \in C$ for $k > 0$,
2) and states are separated by a distance $\|s_j - s_k\| > 2r_s$ for all $j \neq k$

The s_j's are called the *base states* for the system, and they will be used to generate an infinite set of target states.

Also, choose a set of positive scalars[4] $\Gamma = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ bounded by a constant M, and define the set of *dynamic target states* (DTSs) for the switched system as

$$S = \{\alpha_k s_j | \forall j, k\} \bigcup \{-\alpha_k s_j | \forall j, k\}$$

Essentially, the DTSs are positively and negatively scaled copies of the base states that are contained in the open ball $\{x \mid \|x\| < M\}$. The DTSs are the states that will scale to act as target states.

Definition 3: For a given initial mode a_0 and state x_0, the *Dynamic Robust Hybrid Switching Graph* (DRHSG) is the graph $G = (V, E)$ where the vertices V and edges E satisfy

1) every vertex $v \in V$ is a pair (a, x) where either 1) a is any mode and x is a DTS, 2) a is any mode and $x = 0$, or 3) it is the initial pair $(a, x) = (a_0, x_0)$,
2) and every edge $e \in E$ connects the initial state to a DTS: $e = (v_1, v_2) = ((a_0, x_0), (b, y))$ for each mode b and DTS y.

The DRHSG is far simpler than the SRHSG because, as it will be shown, homogeneity reduces the problem of computing a switching path to that of simply mapping a point to the unit sphere to a DTS in S. The DRHSG will scale with each switch.

Finally, define the weighting function w on G according to (5) with the following modification: to enforce homogeneity, the switching penalty is $\hat{\epsilon}_s = \|x_m\|\epsilon_s$, where $\epsilon_s > 0$. Because L and $\hat{\epsilon}_s$ are homogeneous, w is homogeneous.

B. Computing the Switching Path

As with SRHSG, we seek to use a fixed, finite set of target states to reduce the difficulty of computing a suboptimal control pair (u, i) to a linear program. To this end, for a given initial pair (a_0, x_0), we consider the class of control pairs (u, i)

[4] In general, we could associate a different set of scalars Γ_j with each base state s_j, but for ease, we use the single set of scalars Γ.

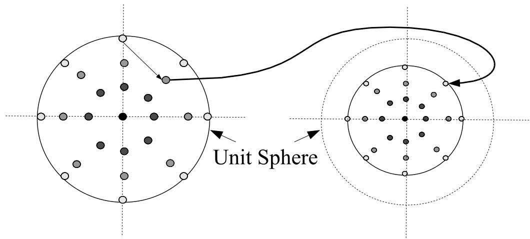

Fig. 3. Illustration of how the dynamic target states scale after a switch.

that yields trajectories x and mode sequences $(a_0, a_1, \ldots)$ having the following properties:

1) each switching state is "close" to a DTS scaled by the previous switching state: x_k ($k > 1$) is such that there exists a DTS $x'_k \in \|x_{k-1}\|S$ with $\|x_k - x'_k\| \leq \|x_{k-1}\|r_s$,
2) two consecutive non-zero switching states are always unique: $x_{k-1} \neq x_k$ for $t_k < \infty$,
3) and u_a always tracks DTSs of the current mode: for $t_k < t < t_{k+1}$, $u(t) = u_{a_k}(x(t), x'_{k+1})$.

Assuming $r_s = 0$ for a moment, the first requirement means that, from a switching state $x(t_k) = x_k$, the system may either track the origin or switch at one of the DTSs scaled by the previous DTS $\|x'_k\|$ (since $x'_k = x_k$ when $r_s = 0$). Figure 3 illustrates the requirement, which preserves homogeneity at each stage of the system's evolution. Because infinite-time tracking is allowed, switching in finite time is achieved by allowing the system to switch when the state is within a scaled factor of the switching radius from a DTS (again, to preserve homogeneity). The final requirement states that the controller tracks only the DTSs of G, even though the actual switching state may vary slightly from this state.

To compute the optimal switching path, we notice that if we start at a base state s_i, then once we switch modes at a DTS $\alpha_k s_j$ (or $-\alpha_k s_j$) and rescale the DTSs by α_k (same as scaling by $-\alpha_k$), we are simply at a scaled base state $\alpha_k s_j$ (or $-\alpha_k s_j$) with the new set of DTSs given by $\alpha_k S$. By homogeneity, in order to compute an optimal switching path, we need to only know the optimal DTS to track from each base state.

Let x_1 be a base state. From x_1, we can track any DTS in any mode. For now, assume that we always apply an optimal mode a^* for tracking; that is, a^* satisfies $J'_{a^*}(x_m, x_n) \leq J'_b(x_m, x_n)$ for all b, x_m, x_n, and we use the controller u_{a^*} for tracking. We can express optimal cost $J(x_1)$ from x_1 as

$$J(x_1) \leq \{J'_{a^*}(x_1, x_s) + J(x_s) \mid \forall x_s \in S\}$$

where $J(x_s)$ denotes the cost-to-go from the DTS x_s.

If the optimal cost of each DTS were known, then the optimal cost from each base state would be known as well by homogeneity: $J(\alpha_k s_j) = |\alpha_k|^z J(s_j)$. Writing this relationship for all DTSs, we arrive at the following linear program that yields the optimal cost from each base state s_j subject to a free initial mode

$$\max \sum_{j=0}^{n} J(s_j) \text{ subject to}$$

$$J(s_j) \leq \{J'_{a^*}(s_j, \alpha s_k) + |\alpha|^z J(s_k), \forall k, \alpha \in \{\Gamma, -\Gamma\}\}$$

From any base state, the optimal DTS to track is given by

$$\chi(s_j) = \arg\min_{x_s \in S} \{J'_{a^*}(s_j, x_s) + J(x_s)\}$$

where $J(x_s)$, the cost-to-go from the DTS x_s, is known since it is simply a scaled cost of a base state. We note that it is implicit in the above formulation that we know the optimal mode to apply.

Now, given any initial state x_0 with $\|x_0\| = 1$ and an initial mode a_0, the optimal DTS to track is simply given as

$$\chi_{a_0}(x_0) \in \arg\min_{x_s \in S} \{J'_{a_0}(x_0, x_s) + J(x_s)\}$$

By homogeneity, we can extend both functions to R^n by a simple scaling:

$$\hat{\chi}(x) = \|x\|\chi(\frac{x}{\|x\|}), \quad \hat{\chi}_a(x) = \|x\|\chi_a(\frac{x}{\|x\|})$$

C. Computing and Applying the Control Law

Application of DRHSG in practice is similar to the application of SRHSG. From an initial state x_0 in mode a_0, track the DTS $x_1 = \hat{\chi}_{a_0}(x_0)$. Once x_1 is tracked, track the scaled DTS $x_2 = \hat{\chi}(x_1)$ in the optimal operating mode. Repeat this process.

We apply the notion of stability given in Definition 2 to DRHSG.

D. Convergence with Optimal Subsystem Controllers

In this section, we show how uniformly increasing the density of the DTSs impacts the overall control law when the subsystem controllers are optimal. Though the results of this section do not impact a practical application of DRHSG, they do justify its use in an optimal setting.

Let Assumptions 1-3 hold, set $r_s = 0$, and choose some $M > 0$ (corresponding to the bound on the α_j's). For a given pair (a_0, x_0), let Ψ^* be the set of control pairs (u^*, i^*) that satisfy

1) switching states are bounded in scale from one another: $\|x_{k+1}\| = R_k\|x_0\|$ for some (R_k) such that, $\sum_{k=0}^{\infty}(R_k)^z < \infty$ and $0 \leq R_{k+1} \leq (M - \delta)R_k$ for some $0 < \delta < M$,
2) no Zeno effects: $t^*_k \to \infty$,
3) optimal tracking of switching states: for $t_k < t < t_{k+1}$, $u^*(t) = u_{a^*_k}(x^*(t), x^*_{k+1})$,
4) finite cost: $J^* < \infty$,
5) and the system tracks the origin if that is optimal: if $J_{a^*_k}(x^*_k, x^*_{k+1}) \geq J_{a^*_k}(x^*_k, 0)$, then $x^*_{k+1} = 0$

where x^* is the trajectory resulting from applying the control pair, (t^*_k) are the switching instances, (a^*_k) are the modes, and J^* is the cost. The first condition bounds the scale between switching states, which is necessary for approximating the trajectory. The purpose of the final condition is the same as that given in the SRHSG case. Also, it is important to note that a control pair in Ψ^* is not limited to tracking predefined DTSs.

First, we construct a sequence of DRHSGs corresponding to DTS sets that grow increasingly dense. Construct the sequence $(S^j)_{j \in Z^+}$ so that it has the following properties

1) increasing density: for each $x \in \overline{B(0,M)}$, there exists an $x_s \in S$ so that $\|x - x_s\| < 1/j$
2) target states are not removed upon refinement: $S^{j+1} \supset S^j$

Denote the corresponding DRHSG sequence as (G^j).

Theorem 2: Given a control pair $(u^*, i^*) \in \Psi^*$ that results in a cost J^*, there exists a sequence of control pairs $\big((u^j, i^j)\big)$, corresponding to the sequence $(G^j)_j$, yielding costs (J^j) such that $J^j \to J^*$.

The proof of Theorem 2 as well as the corollary below are provided in Appendix.

Corollary 3: Given a control pair $(u^*, i^*) \in \Psi^*$ that results in a cost J^*, there exists a sequence of control pairs $\big((u^j, i^j)\big)$, corresponding to the sequence $(G^j)_j$, yielding costs (J^j) such that $J^j \to J \leq J^*$ with strict inequality if (u^*, i^*) is non-optimal.

Once again, by simply increasing the density of the target states in a uniform manner, the performance of the control law converges to the optimal law in Ψ^* (if one exists).

V. Industrial Application of S/DRHSG

Computing the optimal control pair for a S/DRHSG requires the computation of the edge weights of G as well as the switching path, which requires a large amount of computing effort for most systems. To overcome this hurdle, we leverage the granularity of the S/DRHSG construction by using the fact that small shifts in the initial state should not impact the switching path[5]. For SRHSG, we quantize the state space and store in each quantization region the target state to track. For DRHSG, only a quantization of the unit sphere is necessary. In either case, once a switch occurs, the controller references the memory to determine the next target state and mode. A concrete example of storing the switching path in memory is given in the application of SRHSG to the DISC engine.

An advantage of the S/DRHSG methodology is the separation of design between the subsystem controllers and the switching law. Although the optimal convergence properties discussed in the previous sections require certain optimality and continuity conditions, a practical application requires only that the subsystem controllers satisfy Assumption 1. This allows for the use of a number of design techniques in the construction of the subsystem controller. This is in contrast to the approaches described in [8], [2], and [3] that compute the subsystem controllers as part of the design process.

Another interesting feature of S/DRHSG design is that the system's model is not required by the design process so long as the states can be measured because the switching path computation only requires the tracking costs between system states. These costs may be found analytically, by simulation, or even by direct experimentation on the system. This property is especially useful for applications where the subsystem controllers are highly complex or, perhaps, given by a third party and not fully modeled.

Finally, the performance of S/DRHSG may be scaled according to the resource constraints on the controller hardware. By simply increasing the number of switching and quantization regions, higher performance is achieved at the expense of memory consumption. Clearly, the reverse holds as well. In fact, the only design parameters for SRHSG and DRHSG are the locations of the target states and quantization regions.

[5]This can be formally proven if Assumption 2 holds

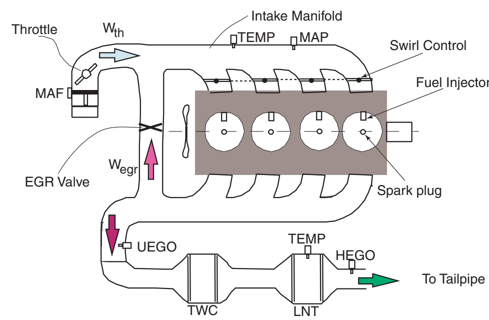

Fig. 4. The DISC Engine.

VI. Application of SRHSG to the DISC Engine

The direct injection stratified charge (DISC) engine is an example of a modern, complex engine where the complexity in control lies in the inclusion of two operating modes (homogeneous and stratified) that accomodate trade-offs in fuel economy, power output, and emissions.

In homogeneous operation, fuel is injected during the intake stroke, providing an approximately uniform air-fuel mixture distribution 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) is normally maintained around the stoichiometric value of 14.6:1.

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 at the time instant coincident with the spark. The typical AFR for this mode of operation is about 35:1, significantly higher than that of the PFI engine. However, in this operating regime, oxides of nitrogen, NO_x, are not efficiently converted by conventional Three Way Catalyst (TNC) and must be stored by an additional specialized catalyst, called the Lean NO_x Trap (LNT), which, over time, becomes saturated and must be regenerated by temporarily switching the engine into the homogeneous regime and then operating at a rich AFR.

The combustion mode to use depends on the amount of torque demanded by the driver, the engine's speed, and the catalyst's state. When torque demands or engine speeds are high, the engine should be operated in homogeneous mode. When the demanded torque or engine speed is low to moderate, and the LNT is operating efficiently, stratified operation should be used to improve efficiency. Ultimately, a high-level controller uses the torque demanded by the driver and the catalyst's state to determine the appropriate AFR to apply. The purpose of the DISC engine controller is to track the torque and mode reference so as to guarantee convergence to these set-points while assuring high transient performance.

A. Nonlinear, Speed-Dependant DISC Engine Model

In this paper, we treat a slightly simplified version of the nonlinear DISC engine model presented in [9] that ignores external exhaust-gas recirculation ($W_{egr} = 0$). Since closing the EGR valve is the most practical and typically-used measure to deal with combustion stability limits during mode transitions, this assumption is reasonable. The form of the nonlinear, speed-dependent DISC engine model is

$$\begin{aligned} \dot{p} &= K\left(w_{cyl} - w_{th}\right) \\ \lambda &= \frac{w_{cyl}}{w_f} \\ \tau &= \left(\theta_1 + \theta_2(\delta - \delta_{mbt})^2\right) w_f + \tau_p + \tau_f \end{aligned} \tag{6}$$

where the control inputs to the system are: the mass flow rate of air through the throttle w_{th}, the fueling rate w_f, and the spark timing δ.

The system outputs are: the intake manifold pressure (IMP) p, the AFR λ, and the brake torque τ.

And the following are internal parameters for the model:

1) engine speed N that is treated as a known (potentially varying) parameter
2) mass flow rate of air into the cylinder w_{cyl} that depends on p and N
3) coefficients θ_1 and θ_2 that depend on λ and N
4) pumping and frictional torque losses τ_p and τ_f that depend on p, λ, and N
5) maximum-brake-torque (MBT) spark timing δ_{mbt} that depends on p, λ, and N; maximum torque (and, hence, maximum fuel efficiency) is achieved when δ is equal to δ_{mbt}

Some of the above parameters also depend on the combustion mode $i(t) \in \{1, 2\}$, which is also a control input. In this paper, $a = 1$ indicates stratified mode, and $a = 2$ indicates homogeneous mode. The details of this model may be found in [9].

B. Parameter Constraints and Control Objectives

In both operating modes, there exist actuator saturations and other practical limits that constrain the ranges of w_{th} to $[w_{th,min}, w_{th,max}]$, w_f to $[w_{f,min}, w_{f,max}]$, and δ to $[0, \delta_{mbt}]$, which is conventional for δ. To avoid misfires and excessive emissions caused by either too rich or too lean an air-fuel mixture, λ is specially bounded to a range that depends on the combustion mode a: $\lambda_{min}(a) \le \lambda \le \lambda_{max}(a)$.

The goal of DISC engine control is to optimally track a given output reference $(\lambda_{ref}, \tau_{ref}, \Delta_{ref})$ according to

$$\int_0^\infty \left|\left| \begin{bmatrix} \lambda(t) - \lambda_{ref}(t) \\ \tau(t) - \tau_{ref}(t) \\ (\delta_{mbt}(t) - \delta(t)) - \Delta_{ref}(t) \end{bmatrix} \right|\right|^2 dt \tag{7}$$

The last reference parameter Δ_{ref} gives the desired difference between δ and δ_{mbt} to balance fuel-efficiency and emissions. In reality, δ_{mbt} is not a fed-back quantity, and, rather, a model of it as a function of p, λ, τ, and a [9] is used. Consequently, we regulate $(\delta_{mbt} - \delta)$ for a given set of these parameters by setting δ to the correct timing. In this paper, Δ_{ref} is fixed to a small, positive constant.

For a fixed engine speed, these three references in combination yield the unique operating mode reference a_{ref} and IMP reference p_{ref} required for tracking the triplet. As N varies, p_{ref} varies accordingly. Since N is relatively constant over a small time interval and since p enters affinely in (6) for a fixed N, solving for p_{ref} is straight-forward.

C. DISC Engine Subsystem Controllers

As p is the key system state in the model, the target states for the SRHSG are placed at various IMP pressures. In order to change modes, the subsystem controllers must track one of these target states p_s. Consequently, while in a mode of operation $a \neq a_{ref}$, the reference to the subsystem controllers will take the form $(p_s, \lambda_{ref}, \tau_{ref})$ where the target state p_s may not be equal to p_{ref}. For simplicity, we use p_d to denote the pressure to track, which may be a target state p_s or the reference p_{ref}.

We now consider the construction of the subsystem controllers for each operating mode of the DISC engine. First, we observe that there is a nice separation in (6) between the control inputs that impact p and those that impact λ and τ. Therefore, we consider the design of controllers for tracking p_d and $(\lambda_{ref}, \tau_{ref})$ separately.

1) IMP Controller: Discretizing and linearizing (6) about a fixed engine speed, we arrive at the following state equation

$$p(t+1) = Ap(t) + Bw_{th}$$

where the dynamics of the throttle do not depend on the combustion mode. Because the range of w_{th} is bounded[6], it is natural to consider the application of a discrete-time bang-bang control. Let $\hat{W}_{th}(p_d, p)$ be the bang-bang control law used to track a given IMP reference p_d.

Of course, as w_{th} represents the desired air-flow through the throttle, w_{th} is actually a reference for another controller. To provide a reference trajectory that is practical for tracking, a low-pass filter is applied to $\hat{W}_{th}$:

$$\dot{w}_{th} = \alpha(\hat{W}_{th}(p_d, p) - w_{th})$$

Let $W_{th}(p_d, p)$ be the filtered version of $\hat{W}_{th}$.

In simulation, we assume that the throttle is able to accurately track the slowly-varying w_{th}, and to account for modeling errors, we apply a simple anti-windup integrator scheme. Because the remaining portions of the DISC engine controller only require the IMP is tracked reasonably well, any appropriate stock controller for this portion of the control system may be equally substituted.

2) Torque and AFR Controller: Over a short time period, p and N may be considered constants. Now, due to constraints on w_f and δ, the reference pair $(\lambda_{ref}, \tau_{ref})$ may not be achievable, forcing us to solve the following nonlinear, constrained optimization problem

$$\min_{w_f, \delta} \left|\left| \begin{bmatrix} \lambda_{ref} - \lambda \\ \tau_f - \tau \end{bmatrix} \right|\right|^2 \tag{8}$$

subject to the constraints on the input and output parameters. In this section, we present a simple approach to solving this

[6] $w_{th,min}$ is always fixed at 0, and $w_{th,max}$ depends on the IMP.

particular problem by means of a fast quantization search that is intended to be performed on-line.

We observe that by fixing p and N, w_{cyl} is known, and so the range $\underline{w}_f \le w_f \le \overline{w}_f$ for the allowable fueling rate can be computed: $\underline{w}_f = \max\{w_{f,min}, w_{cyl}/\lambda_{max}\}$ and $\overline{w}_f = \min\{w_{f,max}, w_{cyl}/\lambda_{min}\}$.

For w_f in this range, δ_{mbt} is uniquely defined, and so we can compute the value of δ that minimizes the error in τ using the following algorithm, which we denote by $\Delta_a(p, N, w_f)$

$\delta_{diff} \leftarrow ((\tau_{ref} - \tau_p - \tau_f)w_f - \theta_1)/\theta_2$
if $\delta_{diff} > 0$ **then**
  $\delta \leftarrow \max\{\underline{\delta}, \min\{\overline{\delta}, \delta_{mbt} - \sqrt{\delta_{diff}}\}\}$
else if $\theta_2 < 0$ **then**
  $\delta \leftarrow \max\{\underline{\delta}, \min\{\overline{\delta}, \delta_{mbt}\}\}$
else {set to the lowest extreme}
  $\delta \leftarrow \underline{\delta}$
end if

Here, $\underline{\delta}$ and $\overline{\delta}$ are the minimum and maximum desired spark timings, respectively. Essentially, Δ_a tries to reverse solve for the difference $\delta_{diff} = (\delta_{mbt} - \delta)$ required to optimally match the reference torque while meeting the constraints on δ. If the optimum cannot be obtained due to the condition $\delta_{diff} < 0$, then, depending on the sign of θ_2, δ should be set to one of its extremes.

To compute the optimal w_f, we apply an iterative search that essentially uses a quantization on the range of w_f to find the pair (w_f, δ) that best minimizes (8). Let $W\Delta_a(p, N, \lambda_{ref}, \tau_{ref})$ be given by the following algorithm:

Compute $\underline{w}_f$ and $\overline{w}_f$
for all $l \in \{1, \ldots, M\}$ **do**
  Choose M_l points $w^l_{f,j} \in [\underline{w}_f, \overline{w}_f]$, with $w^l_{f,1} = \underline{w}_f$, $w^l_{f,M_l} = \overline{w}_f$, and $w^l_{f,j} < w^l_{f,j+1}$.
  for all $j \in \{1, \ldots, M_l\}$ **do**
    $\delta^l_j = \Delta_a(p, N, w^l_{f,j}, \tau_{ref})$
  end for
  Let k correspond to the pair $(w^l_{f,k}, \delta^l_k)$ that best minimizes the output error (8)
  $\underline{w}_f \leftarrow \max\{\underline{w}_f, w^l_{f,k-1}\}$ and $\overline{w}_f \leftarrow \min\{\overline{w}_f, w^l_{f,k+1}\}$
end for
$(w_f, \delta) \leftarrow (w^M_{f,k}, \delta^M_k)$

M is the number of iterations and M_l ($1 \le l \le M$) is the number of grid points along the fueling rate to apply during iteration l. Essentially, after the first stage of the search completes, the algorithm refines the search, using the grid points along w_f adjacent to the optimal value as the new boundaries $\underline{w}_f$ and $\overline{w}_f$ for the next iteration. This smaller region is quantized as before, and the search proceeds. Figure 5 illustrates the algorithm.

The number of required searches in $W\Delta_a$ is simply $\sum_{i=0}^{M} M_l$. It is recommended that higher resolutions are applied first ($M_1 > M_2 > \cdots$) so that "good" center points are applied to the next stage. In practical tests, the accuracy achieved by this iterative approach outperformed that of a single-stage search using significantly more total grid points.

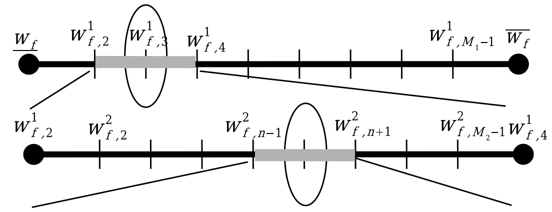

Fig. 5. Illustration of the iterative quantized search for the optimal pair (w_f, δ). The range of the fueling rate is quantized to M_l points at which the optimal δ is computed. The best point becomes the center-point of the next stage of the search.

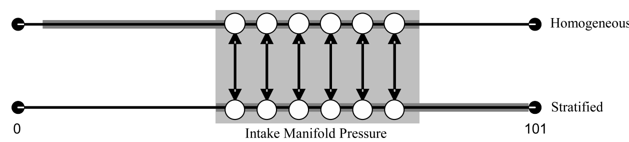

Fig. 6. Illustration of placing the switching states along the common regions of the IMP pressure of each mode.

3) Subsystem Control Law: Given a reference $(p_d, \lambda_{ref}, \tau_{ref})$, let

$$\begin{aligned}(w_{th}, w_f, \delta) &= U_a(p, p_d, \lambda_{ref}, \tau_{ref}, N) \\ &= (W_{th}(p_d, p), W\Delta_a(p, N, \lambda_{ref}, \tau_{ref}))\end{aligned}$$

be the subsystem control law that drives the IMP to p_d while minimizing the AFR and torque tracking errors.

Let $J_a(p_0, p_d, \lambda_{ref}, \tau_{ref})$ be the cost (4) of tracking the reference $(p_d, \lambda_{ref}, \tau_{ref})$ in a mode a starting with an IMP of p_0. If $p_d = p_{ref}$, the cost is clearly finite. Otherwise, if p_d is a target state, the cost is finite since we are able to (closely) track p_d in finite time before switching.

D. Applying SRHSG to the DISC Engine

Assume for now a fixed engine speed $N(t) = N_{op}$. In the interval $[p_{low}, p_{high}]$ of the IMP that is common to both operating modes, choose a finite set of pressures $\{p_s\}$ to act as target states for the controller, as illustrated in Figure 6.

Because the DISC engine is a constrained (as well as non-homogeneous) system, SRHSG is used. We apply it as follows: given an initial mode and state (a_0, p_0) as well as a reference $(\tau_{ref}, \lambda_{ref})$, compute the corresponding reference mode and reference IMP (a_{ref}, p_{ref}) and search for the target state p_s that provides the minimum cost $J_{a_0}(p_0, p_s, \lambda_{ref}, \tau_{ref}) + J_{a_{ref}}(p_s, p_{ref}, \lambda_{ref}, \tau_{ref})$ at which to switch.

To simplify the online use of this scheme, we first force λ_{ref} to be as high as possible in stratified mode and as close to stoichiometry as possible in homogeneous mode. This simplification allows us to further reduce the reference to the pair (τ_{ref}, a_{ref}).

Now, for each mode a, let $\{\tilde{p}^a_k\}$ and $\{\tilde{\tau}^a_k\}$ be finite sets of quantization points for the IMP and torque. For a given fixed engine speed N_{op} and for any two modes a and b, associate to each pair $(\tilde{p}^a_k, \tilde{\tau}^b_l)$ the optimal state p_d to track, which may be a target state p_s or the reference IMP p_{ref}. We denote this mapping as $T_{ab}(\tilde{p}^a_k, \tilde{\tau}^b_l)$ and call T the *switching table*. Its use will be explicitly detailed later.

From a practical standpoint, only one switch is necessary or desirable while tracking a reference in another mode, and so we know that p_{ref} is tracked if $a = b$. Consequently, T_{11} and T_{22} are empty, which lessens the memory requirements of T. A simple compression scheme for reducing the memory requirements of T_{12} and T_{21} is presented in [10]. The compression scheme relies on the fact that many adjacent quantization points along the IMP are mapped to the same target state.

Of course, if $N \neq N_{op}$, p_{ref} will change. However, we will show in simulations that if the target states stored in T_{12} and T_{21} (which are computed using N_{op}) are applied, the resulting system performance is still acceptable.

E. Putting it Together

Algorithm 1 is the full DISC engine controller.

Algorithm 1 Full DISC Engine Controller

Read the reference (a_{ref}, τ_{ref})
if the reference has changed since the last time step **then**
    $tracking \leftarrow false$
    Determine λ_{ref} from (a_{ref}, τ_{ref})
end if
Compute p_{ref} from $(\lambda_{ref}, \tau_{ref})$ and $N(t)$
if $i(t) = a_{ref}$ **then**
    $p_d \leftarrow p_{ref}$
else {a change of modes is required}
    if tracking AND ($p(t)$ has tracked or overshot p_d) **then**
        $i(t) \leftarrow a_{ref}$
        $tracking \leftarrow false$
        $p_d \leftarrow p_{ref}$
    else if NOT tracking **then**
        $p_d \leftarrow T_{i(t)a_{ref}}(p(t), \tau_{ref})$
        $tracking \leftarrow true.$
    end if
end if
$(w_{th}(t), w_f(t), \delta(t)) \leftarrow U_{i(t)}(p(t), p_d, \lambda_{ref}, \tau_{ref}, N(t))$

For a constant reference and an initial (a_0, p_0), the controller first determines if it can track the reference or if it must change modes. If a change of modes is necessary, the controller references the table to retrieve a target state. A change of modes results if the target state is tracked.

In the implementation of the controller, we use a $tracking$ flag and check for overshooting. These measures ensure that the switching table is only applied if the reference changes or if the mode changes, at which time the next target state in the switching path is tracked. Overshoot detection is necessary because the system may overshoot the target state within the sampling period of the discrete-time controller. Since an overshoot implies the state was tracked within the period, the system should switch. Waiting for p to settle will result in poorer performance.

F. Simulations

The parameters for the simulation and the controller were as follows:

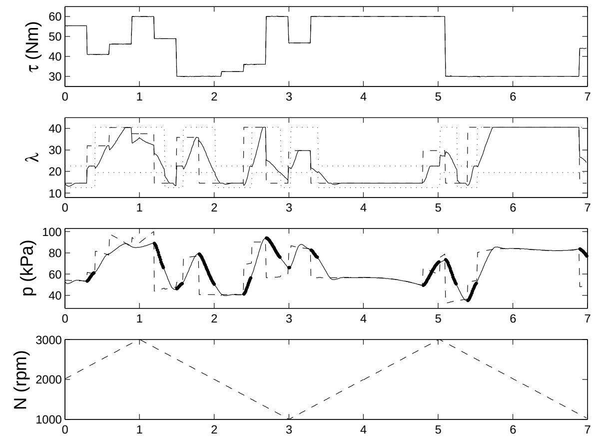

Fig. 7. Simulation of the engine output. Dashed lines are the references, solid lines are the responses, and dotted lines represent constraints. The thick portions of the IMP represent the period over which the IMP is tracking a target state.

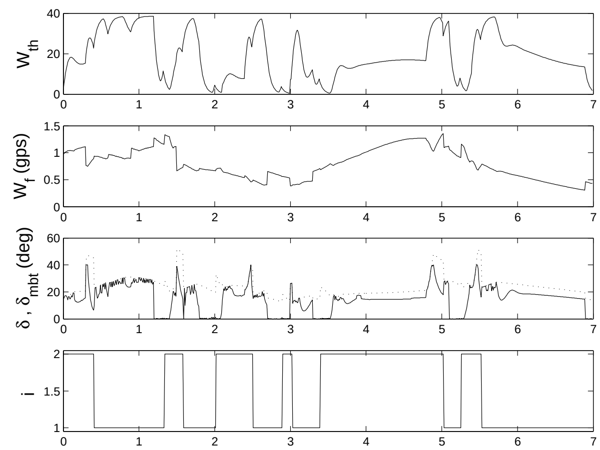

Fig. 8. Simulation of the control inputs. Solid lines are the responses, and the dotted line is δ_{mbt}.

1) target states for p are spaced 2 kPa apart
2) quantization regions along p and τ_{ref} are centered 3 kPa and 2 Nm apart, respectively
3) a three stage search is used in $W\Delta_a$ with 12, 8, and 5 search points applied at each stage
4) the switching table T was designed using $N_{op} = 2000$ rpm, and it uses 29.6 KB of memory
5) the edge weights of the SRHSG were obtained through simulation of the closed-loop system
6) $\Delta_{ref} = 5$ deg and $r_s = 0.1$ kPa
7) the controller sampling period is 10ms
8) torque error is penalized 100 times more heavily than AFR error by a weighting of the norms in (7) and (8)

The last condition indicates that the system always operates in *torque-tracking mode* [9]. Convergence to the air-to-fuel ratio reference is guaranteed in steady-state. It took less than one hour to generate the switching table using a 2.4 GHz PC.

Figure 7 shows the simulated response of the DISC engine to a series of reference torques. To illustrate the impact of engine speed, $N(t)$ is varied throughout the simulation.

Suppose a_{ref} and τ_{ref} change at some time t_0. The controller first determines λ_{ref}. $p_{ref}(t)$ is repeatedly computed as a function of the references and $N(t)$. If $i(t_0) = a_{ref}$, then there is no switching, and the controller simply tracks $p_{ref}(t)$. If $i(t_0) \neq a_{ref}$, then a switch must occur, and the switching state p_s is computed at time t_0; p_s does not change with $N(t)$. Once $p(t) = p_s$, the system switches modes ($i(t) = a_{ref}$), and $p_{ref}(t)$ is tracked. In the figure, the $p(t)$ is made bolder over time periods where it is tracking p_s.

In a fixed mode of operation, tracking is fairly accurate. As $N(t)$ varies, $p_{ref}(t)$ varies accordingly so that the triplet $(\lambda_{ref}, \tau_{ref}, \Delta_{ref})$ can be tracked. The subsystem controllers minimize the difference of $\lambda(t)$ and $\tau(t)$ from λ_{ref} and τ_{ref} respectively.

In Figure 8, it can be seen that w_{th} does not vary too quickly, a consequence of applying a low-pass filter on the bang-bang controller output as a way to account for the throttle actuation dynamics. Also, δ is always contained in its appropriate range, and, in steady-state, it is approximately 5 degrees below δ_{mbt}. There is some chattering in the δ signal, which is a consequence of the quantized search in $W\Delta_a$.

VII. Conclusion

In this paper, new methods for controlling classes of switched systems with controllable subsystems were presented. It was shown that the S/DRHSG switching schemes allow for the design of computationally-practical switched system controllers that are both stabilizing and approximately suboptimal. Several advantages to S/DRHSG for industrial applications include a modular design that separates the subsystem controller design from the switching logic, a robust framework that guarantees convergence and stability along the switching path, and the ability to scale performance with resource requirements.

A successful application of SRHSG to the DISC engine was presented. Although the memory requirements of the controller were not significant, the system is able to track references quickly and accurately. This is due in part to the ability to design complex subsystem controllers separately from the mode control logic.

Acknowledgment

The authors wish to thank Dr. Davor Hrovat of Ford Research and Advanced Engineering, Ford Motor Company for his collaboration on this work.

Appendix

Proof of Theorem 1 and Corollary 2

The proof of Theorem 1 is given below. In the proof, $B(x, r)$ denotes the ball of radius r centered at x.

Proof: We examine the case $x_0 \neq 0$ only. Choose $\epsilon > 0$. First, we find a neighborhood of the origin so that the cost of applying any single-mode infinite-horizon law is less than ϵ. For all $a \in Q_0$, the continuity of J_a implies $\sup_{x \in S} \{J_a(x, 0)\}$ exists and is finite over any bounded neighborhood $S \subset X_a$ of the origin. Let $r > 0$ be such that $B(0, r)$ satisfies $\sup_{x \in B(0,r) \cap X_a} \{J_a(x, 0)\} < \epsilon$ for all modes $a \in Q_0$.

Let τ_1 be a time instant such that that remaining given cost at this time $J^*(x^*(\tau_1)) < \epsilon$, let τ_2 be the time instant such that $\|x^*(\{t > \tau_2\})\| < r$, and let τ_3 be the time instant for which $i(\{t > \tau_3\}) \subset Q_0$. Define $\tau = \max\{\tau_1, \tau_2, \tau_3\}$, and let $K = (\arg\max_k \{t_k^* \leq \tau \mid x_k^* \neq 0\} + 1)$. Note that t_K^* satisfies the three conditions listed above. Also, $x_k^* \neq 0$ for $k < K$, but, by the construction of Ω^*, if $x_K^* = 0$ then $x_k^* = 0$ for all $k \geq K$.

Let

$$A(x, S_{ab}) = \arg \min_{x_s \in S_{ab} \bigcup \{0\}} \{\|x - x_s\| \mid x_s = 0 \text{ iff } x = 0\}$$

yield the target state in S_{ab} that best approximates x, using 0 if and only if $x = 0$.

By the uniform continuity of J_a over any bounded set, there exists a γ such that for all modes $a, b \in Q$ and states $y, z \in X_{ab}$, $|J_a(y, z) - J_a(\tilde{y}, \tilde{z})| < \epsilon/K^2$ for all $\|\tilde{y} - y\| < \gamma$ and $\|\tilde{z} - z\| < \gamma$. Choose an integer $N_1 > \frac{1}{\gamma}$.

By optimality, $J_{a_k^*}(x_k^*, x_{k+1}^*) < J_{a_k^*}(x_k^*, 0)$ for all k such that $x_{k+1}^* \neq 0$. By continuity, there are open neighborhoods U_k and V_k about x_k^* and x_{k+1}^* respectively such that for any state $x_a \in U_k$ and $x_b \in V_k$, $J_{a_k^*}(x_a, x_b) < J_{a_k^*}(x_a, 0)$ (so that x_b rather than 0 is tracked by the subsystem controllers). Let α be such that $B(x_k^*, \alpha) \subset U_k$ and $B(x_{k+1}^*, \alpha) \subset V_k$ for all $0 \leq k < K$ such that $x_{k+1}^* \neq 0$, and choose an integer $N_2 > \frac{1}{\alpha}$.

For some $j > \max\{N_1, N_2\}$, we construct a control pair $(u^j, i^j) \in \Omega_{G^j}$ that "follows" the given trajectory in the following sense

1) $x_k = A\left(x_k^*, S^j_{a_{k-1}^* a_k^*}\right)$ for $k \leq K$; $x_{K+1} = 0$
2) for $k < K$, $a_k = a_k^*$, and choose a mode $a_K \in Q_0$ that can be switched to from x_K

An upper bound for the difference between J^* and J^j is

$$\begin{aligned}
|J^* - J^j| &\leq \sum_{k=0}^{K-1} \left| J_{a_k^*}\left(x_k^*, x_{k+1}^*\right) - J_{a_k^*}\left(x_k, x_{k+1}\right) \right| \\
&\quad + \left| J^*(x_K^*) - J_{a_K^*}(x_K, 0) \right| \\
&\leq \sum_{k=0}^{K-1} \frac{\epsilon}{K^2} + 2\epsilon \leq \sum_{k=0}^{K-1} \frac{\epsilon}{k^2} + 2\epsilon \leq \sum_{k=0}^{\infty} \frac{\epsilon}{k^2} + 2\epsilon \\
&\leq \left(\frac{\pi^2}{6} + 2\right)\epsilon
\end{aligned}$$

■

The proof of Corollary 2 is below.

Proof: By Theorem 1, there exists a sequence $\left((\hat{u}^j, \hat{i}^j)\right)$ whose costs $(\hat{J}^j)$ converge to J^*. Let $(u^j, i^j) = \Theta\left(G^j\right)$, and let (J^j) be the resulting cost sequence. As J^j is a decreasing, positive sequence, it converges. Since $J^j \leq \hat{J}^j$, we have $J^j \to J \leq J^*$. It is straight forward to prove the remainder of the claim. ■

Proof of Theorem 2 and Corollary 3

The proof of Theorem 2 is given below.

Proof: We examine the case for $x_0 \neq 0$ only. Choose $\epsilon > 0$. Define r, τ, K, and $A(\cdot, \cdot)$ as they are defined in the proof of Theorem 1 but noting that $X_a = \Re^n$, $Q = Q_0$, and

x^* follows the restrictions defined by the class Ω^*. Let (R_k) be the sequence associated with x^*, and set $\delta > 0$ so that $R_{k+1} \leq (M-\delta)R_k$. Once again, our definition of K ensures $x_k^* \neq 0$ for $0 \leq k < K$, and if $x_K^* = 0$ then $x_k^* = 0$ for all $k \geq K$.

For a given j, we attempt to construct a control pair (u^j, i^j) satisfying

1) $a_k = a_k^*$ for $k \leq K$; a_{K+1} is arbitrary
2) $x_{k+1} = A(x_{k+1}^*, \|x_k^j\|S^j)$ for $k \leq K$; $x_{K+1} = 0$

Now we must determine a lower bound N such that for all $j > N$, such a pair (u^j, i^j) exists.

For $0 \leq k < K$, let $x_{k+1}^* = Lx_k^*$ for a matrix L satisfying $\|L\| < (M-\delta)$. If $\|x_k^* - x_k\| \leq \frac{\|x_k^*\|\delta}{M}$, then it can be shown that (see below for a proof of this result)

$$\|x_{k+1}^* - x_{k+1}\| = \|Lx_k^* - A(Lx_k^*, \|x_k\|S^j)\| < \frac{\|x_k\|}{j} \quad (9)$$

This inequality is important because it means that x_k is close enough to x_k^* so as to allow us to make the difference $\|x_{k+1}^* - x_{k+1}\|$ arbitrarily small by increasing j. Let $W = \max\{M, 1\}$. Since $\|x_k\| < M^k\|x_0\|$, for an integer N_1 satisfying

$$N_1 > \frac{W^K\|x_0\|}{\delta \min_{0\leq k<K}\|x_k^*\|}$$

$j > N_1$ gives for $0 \leq k < K$

1) $\|x_k^* - x_k\| \leq \frac{\|x_k^*\|\delta}{M}$
2) $\|x_{k+1}^* - x_{k+1}\| \leq \frac{\|x_k\|}{j} \leq \frac{W^K\|x_0\|}{j}$

By the uniform continuity of J_a over the compact set $\overline{B(0,W)}$, for all modes a there exists a γ such that for all $y \in C$ and $z \in B(0, M-\delta)$, $|J_a(y,z)) - J_a(\tilde{y}, \tilde{z})| < \epsilon$ for all $\|\tilde{y} - y\| < \gamma$ and $\|\tilde{z} - z\| < \gamma$. Let $N_2 > \frac{W^K\|x_0\|}{\gamma}$.

By homogeneity and optimality, $J_{a_k^*}(\frac{x_k^*}{\|x_k^*\|}, \frac{x_{k+1}^*}{\|x_k^*\|}) < J_{a_k^*}(\frac{x_k^*}{\|x_k^*\|}, 0)$ for all k such that $x_{k+1}^* \neq 0$. By continuity, there are open neighborhoods U_k and V_k about $\frac{x_k^*}{\|x_k^*\|}$ and $\frac{x_{k+1}^*}{\|x_k^*\|}$ respectively such that for any state $x_a \in U_k$ and $x_b \in V_k$, $J_{a_k^*}(x_a, x_b) < J_{a_k^*}(x_a, 0)$ (so that x_b rather than 0 is tracked by the subsystem controllers). Let α be such that $B(\frac{x_k^*}{\|x_k^*\|}, \alpha) \subset U_k$ and $B(\frac{x_{k+1}^*}{\|x_k^*\|}, \alpha) \subset V_k$ for all $0 \leq k < K$ such that $x_{k+1}^* \neq 0$, and choose an integer $N_3 > \frac{1}{\alpha}$.

For $j > \max\{N_1, N_2, N_3\}$, an upper bound for the difference between J^* and J^j is

$$\begin{aligned}|J^* - J^j| &\leq \sum_{k=0}^{K-1} |J_{a_k^*}(x_k^*, x_{k+1}^*) - J_{a_k^*}(x_k, x_{k+1})| \\ &\quad + |J^*(x_K^*) - J_{a_K^*}(x_K, 0)| \\ &\leq \sum_{k=0}^{K-1} \|x_k^*\|^z(|J_{a_k^*}(x_k^*/\|x_k^*\|, x_{k+1}^*/\|x_k^*\|) \\ &\quad - J_{a_k^*}(x_k/\|x_k^*\|, x_{k+1}/\|x_k^*\|)|) + 2\epsilon \\ &\leq \left(\|x_0\| \sum_{k=0}^{\infty} (R_k)^z + 2\right)\epsilon\end{aligned}$$

■

We now prove that equation (9) is true.

Proof: We use the notation from the proof of Theorem 2. We can express the difference between a given state x_{k+1}^* and the approximating switching state x_{k+1} as

$$\|x_{k+1}^* - x_{k+1}\| = \|Lx_k^* - A(Lx_k^*, \|x_k\|S^j)\| \quad (10)$$

where L is some matrix satisfying $\|L\| < (M-\delta)$.Of course, if Lx_k^* is contained in the region "covered" by $\|x_k\|S^j$ (namely $B(0, \|x_k\|M)$ as the switching states are all scaled by $\|x_k\|$) then we know we can bound the difference (10) by $\frac{\|x_k\|}{j}$ ($\frac{1}{j}$ scaled by $\|x_k\|$). Therefore, a sufficient condition to ensure

$$\|x_{k+1}^* - x_{k+1}\| < \frac{\|x_k\|}{j} \quad (11)$$

is

$$\begin{aligned}&Lx_k^* \in B(0, \|x_k\|M) \\ &\Leftrightarrow \|Lx_k^*\| < M\|x_k\|\end{aligned} \quad (12)$$

If $x_{k+1}^* = 0$, we can track it perfectly, so we are only interested in bounding the difference for non-zero states. For these states, since $0 < \|L\| < (M-\delta)$, a sufficient condition for (12) is

$$\begin{aligned}\|x_k\| &> \frac{(M-\delta)}{M}\|x_k^*\| \\ &= \|x_k^*\| - \frac{\delta}{M}\|x_k^*\|\end{aligned} \quad (13)$$

which itself is sufficiently satisfied by the following condition

$$\|x_k^* - x_k\| < \frac{\delta}{M}\|x_k^*\| \quad (14)$$

Therefore, (14) $\Rightarrow$ (11).

Now, we want the $\frac{1}{j}$ factor to appear for the first K switches. First, if $\|x_k^* - x_k\|$ satisfies (14), we want $\|x_{k+1}^* - x_{k+1}\|$ to satisfy (14) for $k \to (k+1)$ to ensure that $\|x_{k+2}^* - x_{k+2}\|$ satisfies (11).

Assume (11) is true. We want to pick j large enough to guarantee $\frac{\|x_k\|}{j} < \frac{\delta}{M}\|x_{k+1}^*\|$ to yield (14) for $k \to (k+1)$. As j is the only unknown, solving for j yields

$$j > \frac{M\|x_k\|}{\delta\|x_{k+1}^*\|} \quad (15)$$

Of course, x_k is a function of j, but observing that $\|x_k\| < M^k\|x_0\|$ for all k, we satisfy (11) for the first K switches by bounding j by the following

$$j > \frac{W^K\|x_0\|}{\delta \min_{0\leq k<K}\|x_k^*\|} \quad (16)$$

which is independent of j. This condition ensures the first $(K-1)$ switches satisfy (14). Since $x_k^* \neq 0$ for $k < K$, it is well defined. Therefore, the first K switches satisfy

$$\|x_k^* - x_k\| < \frac{\|x_{k-1}\|}{j} < \frac{W^K\|x_0\|}{j} \quad (17)$$

which is exactly what was desired. ■

The proof of the Corollary 3 is similar to that of Corollary 2.

References

[1] N. H. El-Farra and P. Christofides, "Switching and feedback laws for control of constrained switched nonlinear systems," in *Proceedings of the 5th International Workshop on Hybrid Systems: Computation and Control*, 2002, pp. 164–178.

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

[3] X. Xu and P. Antsaklis, "Optimal control of switched systems based on a parameterization of switching instants," *IEEE Trans. Autom. Control*, vol. 49, no. 1, pp. 2–16, Jan. 2004.

[4] A. Bemporad, N. Giorgetti, I. Kolmanovsky, and D. Hrovat, "A hybrid approach to modeling and optimal control of disc engines," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 2, Dec. 2002, pp. 1582–1587.

[5] N. Giorgetti, A. Bemporad, I. Kolmanovsky, and D. Hrovat, "Explicit hybrid optimal control of direct injection stratified charge engines," in *Proceedings of IEEE International Symposium on Industrial Electronics*, Jun. 2005, pp. 391–398.

[6] E. Frazzoli, M. Dahleh, and E. Feron, "Robust hybrid control for autonomous vehicle motion planning," in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 1, Dec. 2000, pp. 821–826.

[7] D. Bertsekas, *Dynamic Programming and Optimal Control (vol. I)*, 3rd ed. Nashua, NH: Athena Scientific, 2005.

[8] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, pp. 407–427, 1998.

[9] J. Sun, I. Kolmanovsky, D. Brehob, J. Cook, J. Buckland, and M. Haghgooie, "Modeling and control problems for gasoline direct injection stratified charge (disc) engines," in *Proceedings 1999 IEEE Conference Control Applications*, 1999, pp. 471–477.

[10] M. Rinehart, M. Dahleh, and I. Kolmanovsky, "Optimal control of the disc engine using hierarchal and quantized control," in *Proceedings 2005 American Control Conference*, vol. 2, Jun. 2005, pp. 997–1002.

[11] M. Branicky, V. Borkar, and S. Mitter, "A unified framework for hybrid control: Model and optimal control theory," *IEEE Trans. Autom. Control*, vol. 43, no. 1, pp. 31–45, Jan. 1998.

[12] M. Branicky, "Multiple lyapunov functions and other analysis tools for switched and hybrid systems," *IEEE Trans. Autom. Control*, vol. 43, no. 4, pp. 475–482, Apr. 1998.

[13] E. Frazzoli, M. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," in *Proceedings of the American Control Conference*, vol. 1, Jun. 2001, pp. 43–49.

[14] I. Kolmanovsky, M. Druzhinina, and J. Sun, "Speed-gradient approach to torque and air-to-fuel ratio control in disc engines," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 5, pp. 671–678, Sep. 2002.