Real-Time Dynamic Mode Scheduling Using Single-Integration Hybrid Optimization

Anastasia Mavrommati, *Student Member, IEEE,* Jarvis Schultz, *Member, IEEE,* and Todd D. Murphey, *Member, IEEE*

***Abstract*—This paper introduces and implements a method for real-time mode scheduling in linear time-varying switched systems subject to a quadratic cost functional. The execution time of switched system algorithms is often prohibitive for real-time applications and typically may only be reduced at the expense of approximation accuracy. We address this trade-off by taking advantage of system linearity to formulate a projection-based approach such that no simulation is required during open-loop optimization. A numerical example shows how the proposed open-loop algorithm outperforms methods employing common numerical integration techniques. Additionally, we follow a receding-horizon scheme to schedule the modes of a customized experimental setup in real time, using the Robot Operating System (ROS). In particular, we demonstrate—both in Monte-Carlo simulation and in experiment—that optimal mode scheduling efficiently regulates a cart and suspended mass system and rejects disturbances online.**

***Note to Practitioners*—This paper is motivated by the problem of reliable and fast implementation of mode scheduling algorithms in real-time applications where control authority is discrete. Switched systems cover a range of real-world control platforms like antilock braking systems and other valve-operated settings. However, most current approaches to dynamic compensation rely on specialized ODE solvers to simulate the hybrid dynamics and thus exhibit high execution times and/or approximation errors. In our approach, we only require that a set of data is calculated offline and stored in memory so that online computational complexity is significantly reduced. Importantly, for memory efficiency, we show that approximation accuracy is independent of the number of stored samples i.e. the size of the stored dataset. Using ROS, we apply the proposed algorithm to regulate the swing angle of a mass suspended from a planar robotic system in real time with hybrid control signals. Our experimental results verify that our algorithm is fast and rejects disturbances online even using inexpensive hardware for sensing and actuation. This result presents an opportunity for real-time optimal control of automation platforms with finite set of control signals.**

***Index Terms*—switching controllers, real-time experimental validation, receding-horizon control, optimal control**

I. INTRODUCTION

It is common in the automation industry that control authority is not continuous, for example, due to the discontinuous characteristics of actuators operated by valves. Optimal mode scheduling—model-based control of both the sequence and timing of operating modes—is a natural and efficient way of approaching these discrete[1] problems, with an abundance of algorithms proposed by the research community (e.g. [1], [2] and more—see below). However, despite the common use of model predictive control (MPC, in the sense of receding horizon control as defined in Section III-B) for continuous control problems, mode scheduling algorithms are rarely used in MPC schemes for real-time control in discrete control setting—in fact, the discontinuous components of automation platforms are sometimes disregarded to allow for application of continuous dynamic solutions (e.g. [3]). The main reasons for this lack of applicability are prohibitive execution times and/or high approximation errors resulting from the use of specialized ODE solvers for numerical integration of hybrid differential equations. We aim to overcome this drawback, by considering the problem of real-time mode scheduling for an autonomous linear time-varying switched system to optimize a quadratic performance metric. In particular, the contributions of this paper are: 1) The formulation of an open-loop mode scheduling algorithm (referred to as Single Integration Optimal Mode Scheduling—SIOMS) so that no differential equation needs to be solved for during optimization; 2) The formulation and experimental validation of receding-horizon SIOMS for real-time closed-loop mode scheduling so that a differential equation only needs to be integrated over a limited time interval—typically the time step of the receding-horizon window—rather than the full time horizon.

Switching control problems arise in a number of application domains in automation industry, such as robot locomotion [4], manufacturing production [5], [6], power electronics [7], telecommunications [8] and air traffic management [9]. Several algorithmic methods have been proposed to deal with scheduling problems. Many approaches have relied on a bi-level hierarchical structure with only a subset of the design variables considered at each level [6], [10]. Other proposed methods include: embedding methods [2], which relax, or embed, the integer constraint and find the optimal of the relaxed cost; relaxed dynamic programming [11], [12] where complexity is reduced by relaxing optimality within pre-specified bounds; variants of gradient-descent methods [13], [14]; and application-specific solutions [15].

The iterative projection-based approach as introduced in [16]–[18] forms the basis for the work in this paper. The mode scheduling problem is formulated as an infinite-dimensional optimal control problem where the variables to be optimized

Anastasia Mavrommati, Jarvis Schultz and Todd D. Murphey are with the Department of Mechanical Engineering, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208, USA

Email: stacymav@u.northwestern.edu; jschultz@northwestern.edu; t-murphey@northwestern.edu

[1]Here and for the rest of the paper, "discrete" denotes a finite number of control settings and is not to be confused with discretization in time, unless otherwise noted.

are a set of functions of time constrained to the integers. For a projection-based method, the design variables are in an unconstrained space but the cost is computed on the projection of the design variables to the set of admissible switched system trajectories. In [16], an iterative optimization algorithm is synthesized that employs the Pontryagin Maximum Principle and a projection-based technique. We adapt this algorithm by taking advantage of the linearity of the dynamical system under concern. The specific case of linear switching control has been extensively investigated by others (see [19]–[21]). The approaches in [22], [23] solve for a differential equation at each step of the iterative algorithm. Previous attempts to avoid the online integration of differential equations are limited to switching time optimization problems where the mode sequence is fixed [22]–[24].

In this paper, we extend our work in [25] to present and experimentally evaluate a projection-based mode scheduling algorithm (SIOMS) where a single set of differential equations is solved offline, so that no additional simulation is required during the open-loop optimization routine. *These offline solutions to differential equations are independent of the mode sequence and switching times* in contrast to [22]–[24]. Moreover, no assumption about the time-variance of the modes is made. Therefore, SIOMS does not exclude many important linear systems, such as time-varying power systems [26], traffic models [27], and nonlinear systems linearized about a trajectory.

Here, our objective is to emphasize the importance of SIOMS as a tool for model-based mode scheduling in real-time applications (see [6], [28], for example). Real-time implementation of switched system algorithms is often impractical due their dependence on numerical solutions of differential equations. Three main points are listed to support this argument. First, the use of ODE solvers often renders algorithm execution times prohibitive for real-time implementation [29], [30]. Second, the solution approximation through discretization does not always guarantee consistency (see Definition 3.3.6 in [31]), and lastly, discontinuous differential equations require specialized event-based numerical techniques that are prone to approximation errors [32].

SIOMS overcomes the aforementioned issues by avoiding online simulations. As far as the first point is concerned, one of the strongest assets of the proposed algorithm is that its *timing behavior—i.e., the execution time of a single iteration of the optimization algorithm—is independent of the choice of ODE solver*; it only depends on the number of multiplications and inversions required for the calculation of the optimality condition. As a result, *SIOMS is fast* and intrinsically free of the common trade-off between execution time and approximation accuracy that normally dictates the selection of numerical integration technique. Furthermore, authors in [30] address the second issue by proposing a time discretization that guarantees consistency for nonlinear systems. In this paper, by restricting our focus to linear time-varying systems, we introduce a method where *approximation accuracy and consistency are independent of the number of samples* used for approximation of the state and co-state trajectories. Lastly, to address the third point, SIOMS only requires offline integration of differential equations that are continuous and as smooth as each of the linear modes. Thus, our algorithm exhibits *robustness to numerical errors due to discontinuous vector fields*. All the above SIOMS advantages are verified through a method-comparison numerical study in Section IV.

Exploiting the aforementioned computational and timing advantages, we can use SIOMS for real-time control applications by means of a receding-horizon synthesis [33]. Importantly, in receding-horizon optimization with SIOMS, a simple update step removes the need for numerical integration over the full time horizon in between consecutive algorithm runs—only integration over a few time steps is required. Although stability analysis of closed-loop SIOMS is not provided in this paper, we include a short discussion on stability requirements based on established results in Section III-B.

Finally, we choose to experimentally validate SIOMS real-time implementability by regulating the swing angle of a mobile robot and suspended mass system, online, using a finite number of control actions (i.e. switched system modes). For additional complexity, the string holding the mass exhibits a pre-defined time-varying length. Although not intrinsically a switched system, our example resembles many systems that admit hybrid input by construction and thus are difficult to control (e.g. antilock braking systems (ABS) [34], tanks [35] and other valve-operated systems). Moreover, despite the fact that conventional real-time control of variants of this system has been extensively studied [36]–[38]), we are interested in showing how a limited number of actions may suffice for control, even with time-varying parameters; a result that opens a discussion for alternative, inexpensive actuation and sensing solutions in seemingly complex control platforms. Our experimental work—based on the Robot Operating System (ROS)—demonstrates that closed-loop SIOMS regulates the example system reliably in real time, while rejecting disturbances.

This paper is structured as follows: Section II reviews switched systems and their representations while stating the optimization problem. The single-integration mode scheduling algorithm is proposed in Section III where a receding-horizon approach for closed-loop control is also introduced. Details about the numerical implementation of open-loop SIOMS are provided in Section IV, along with a comparison with former implementations [16], [17]. Finally, Section V verifies closed-loop SIOMS through a Monte-Carlo analysis and a real-time experiment.

II. Review

A. Switched Systems

Switched systems are a class of hybrid systems [39], [40] that evolve according to one of N vector fields (modes) $f_i : \mathbb{R}^n \to \mathbb{R}, i \in \{1, ..., N\}$ at any time over the finite time interval $[T_0, T_M]$, where T_0 is the initial time and $T_M > 0$ is the final time. We consider two representations of the switched system, namely mode schedule and switching control. As a unique mapping exists between each representation [16], the two will be used interchangeably throughout the paper.

Definition 1 The *mode schedule* is defined as the pair $\{\Sigma, \mathcal{T}\}$ where $\Sigma = \{\sigma_1, ..., \sigma_M\}$ is the sequence of active modes $\sigma_i \in$

$\{1, ..., N\}$ and $\mathcal{T} = \{T_1, ..., T_{M-1}\}$ is the set of the switching times $T_i \in [T_0, T_M]$. The total number of modes in the mode sequence—which may vary across optimization iterations—is $M \in \mathbb{Z}^+$.

Definition 2 A *switching control* corresponds to a list of curves $u = [u_1, ..., u_N]^T$ composed of N piecewise constant functions of time, one for each different mode f_i. For all $t \in [T_0, T_M]$, $\sum_{i=1}^{N} u_i(t) = 1$, and for all $i \in \{1, ..., N\}, u_i(t) \in \{0, 1\}$. This dictates that the state evolves according to only one mode for all time. We represent the set of all admissible switching controls as Ω.

We will refer to the mode schedule corresponding to the switching control u as $\{\Sigma(u), \mathcal{T}(u)\}$.

For a system with n states $x = [x_1, ..., x_n]^T$ and N different modes, the state equations are given by

$$\dot{x}(t) = F(t, x(t), u(t)) := \sum_{i=1}^{N} u_i(t) f_i(x(t), t) \tag{1}$$

subject to the initial condition $x(T_0) = x_0$. For this paper, we restrict our focus to linear time-varying systems so that

$$F(t, x(t), u(t)) := \sum_{i=1}^{N} u_i(t) A_i(t) x(t). \tag{2}$$

Alternatively, we may express the system dynamics with respect to the current mode schedule as follows:

$$F(t, x(t), \Sigma, \mathcal{T}) := \overline{A}(t, \Sigma, \mathcal{T}) x(t) \tag{3}$$

where $\overline{A}(t, \Sigma, \mathcal{T}) = A_{\sigma_i}(t)$ for $T_{i-1} \leq t < T_i$.

B. Problem Statement

Our objective is the minimization of a quadratic cost function

$$J(x, u) = \int_{T_0}^{T_M} \frac{1}{2} x(\tau)^T Q(\tau) x(\tau) d\tau + \frac{1}{2} x(T_M)^T P_1 x(T_M) \tag{4}$$

where x is the state, u the switching control and the pair (x, u) satisfy (1). Here, Q and P_1 are the running and terminal cost respectively, and are both symmetric positive semi-definite. Note that this cost functional can also be adapted to include reference trajectory, in which case the objective would be to minimize the error between the state and the reference ([24]).

C. Projection-based Optimization

From Definition 2 of an admissible switching control u, it follows that our optimization problem is subject to an integer constraint [16]. Let $\mathcal{S}$ represent the set of all pairs of admissible state and switching control trajectories (x, u), i.e. all pairs that satisfy the constraint (1) and are consistent with Definition 2 so that $u \in \Omega$. In [17], the authors propose a projection-based technique for handling these constraints set by $\mathcal{S}$. In particular, an equivalent problem is considered where the design variables (α, μ) belong to an unconstrained set $(\mathcal{X}, \mathcal{U})$ and the cost J is evaluated on the projection of these variables to the set $\mathcal{S}$. Now, the problem is reformulated as

$$\arg\min_{(\alpha, \mu)} J(\mathcal{P}(\alpha, \mu)) \tag{5}$$

where $\mathcal{P}$ is a projection—with $\mathcal{P}(\mathcal{P}(\alpha, \mu)) = (\mathcal{P}(\alpha, \mu))$—that maps curves from the unconstrained set $(\mathcal{X}, \mathcal{U})$ to the set of admissible switched systems $\mathcal{S}$. As the cost is calculated on the admissible projected trajectories, this problem is equivalent to the original problem described in II-B and (4).

The optimal mode scheduling algorithm developed in [16] utilizes the max-projection operator. The max-projection operator $\mathcal{P} : \mathcal{X} \times \mathcal{U} \to \mathcal{S}$ at time $t \in [T_0, T_M]$ is defined as

$$\mathcal{P}(\alpha(t), \mu(t)) := \begin{cases} \dot{x}(t) = F(t, x(t), u(t)), & x(T_0) = x_0 \\ u(t) = Q(\mu(t)) \end{cases} \tag{6}$$

where Q is a mapping from a list of N real-valued control trajectories, $\mu(\cdot) = [\mu_1(\cdot), ..., \mu_N(\cdot)]^T \in \mathbb{R}^N$ to a list of N feasible switching controls, $u \in \Omega$. We define Q as

$$Q(\mu(t)) = \begin{bmatrix} Q_1(\mu(t)) \\ \vdots \\ Q_N(\mu(t)) \end{bmatrix} \quad \text{with} \quad Q_i(\mu(t)) := \prod_{j \neq i}^{N} \mathbf{1}(\mu_i(t) - \mu_j(t)) \tag{7}$$

where $\mathbf{1} : \mathbb{R} \to \{0, 1\}$ is the step function given by

$$\mathbf{1}(t) = \begin{cases} 1, & t \geq 0 \\ 0, & else. \end{cases} \tag{8}$$

Notice that the max-projection operator does not depend on the unconstrained state trajectories $\alpha(\cdot)$. The unconstrained state α is included in the left hand side of the definition in order for $\mathcal{P}$ to be a projection.

D. Mode Insertion Gradient

The *mode insertion gradient* appears in previous studies [1], [41], [42]. Here, it is defined as the list of functions $d = [d_1(t), ..., d_N(t)] \in \mathbb{R}^N$ that calculate the sensitivity of cost J to inserting one of the N modes at some time t for an infinitesimal interval (i.e. $\frac{dJ}{d\lambda^+}$ as $\lambda^+ \to 0$). Each element of d is given by:

$$d_i(t) := \rho(t)^T (f_i(x(t), t) - f_{\sigma(t)}(x(t), t)), \quad i = 1, ..., N \tag{9}$$

where $x \in \mathbb{R}^n$ is the solution to the state equations (1) for all $t \in [T_0, T_M]$ and $\rho \in \mathbb{R}^n$, the co-state, is the solution to the adjoint equation[2]

$$\dot{\rho}(t) = -D_x F(t, x(t), u(t))^T \rho(t) - Q(t) x(t), \tag{10}$$

for all $t \in [T_0, T_M]$ subject to $\rho(T_M) = P_1 x(T_M)$. (In (9), $\sigma(t) : [T_0, T_M] \to \{1, ..., N\}$ is the function that returns the active mode at any time t.)

It has been shown in [43], that when a quadratic cost is optimized subject to a linear time-varying switched system, a linear mapping between state x and co-state ρ exists. Thus, we may express the co-state as

$$\rho(t) = P(t) x(t) \tag{11}$$

where $P(t) \in \mathbb{R}^{n \times n}$ is calculated by the following differential equation:

$$\dot{P}(t) = -\overline{A}(t, \Sigma, \mathcal{T})^T P(t) - P(t) \overline{A}(t, \Sigma, \mathcal{T}) - Q(t) \tag{12}$$

[2] $D_x f(\cdot)$ denotes the partial derivative $\frac{\partial f(\cdot)}{\partial x}$.

subject to $P(T_M) = P_1$. Note that this is the linear switched system analog to the Riccati equation from the LQR problem in classical control theory [44]. Using (2) and (11), the mode insertion gradient element can be written as

$$d_i(t) := x(t)^T P(t)^T [A_i(t) - A_{\sigma(t)}(t)]x(t). \tag{13}$$

E. Iterative Optimization

To calculate the switching control $u(t)$ that optimizes the quadratic performance metric (4), we follow an iterative approach. Iterative optimization computes a new estimate of the optimum by taking a step from the current estimate in a search direction so that a sufficient decrease in cost is achieved [13], [16], [42], [45]. A single iteration is commonly structured in the following scheme: Given a current iterate, i) Calculate a descent direction; ii) Calculate a step size; iii) Update the current iterate by taking a step in the descent direction. The procedure is repeated until a terminating condition is satisfied.

In the following section, we formulate an iterative projection-based algorithm for quadratic optimization of linear time-varying switched systems that requires no online simulations.

Algorithm 1 SIOMS

Offline:

- Solve for the STM $\Phi^j(t, T_0)$ and ATM $\Psi^j(t, T_M)$ $\forall j \in \{1, ..., N\}$ and $t \in [T_0, T_M]$.
- Choose initial $u^0 \to \{\Sigma(u^0), \mathcal{T}(u^0)\}$.
- Set $x(T_0) = x_0$ and $P(T_M) = P_1$.

Online iterative process:
Set $k = 0$, $u^k = u^0$.

1) Evaluate $x^k(t) := \chi(t, \Sigma(u^k), \mathcal{T}(u^k))$ as in Eq. (15).
2) Evaluate $P^k(t) := \varrho(t, \Sigma(u^k), \mathcal{T}(u^k))$ as in Eq. (21).
3) Evaluate the descent direction $-d^k(t)$ as in Eq. (28).
4) Calculate step size γ^k by backtracking.
5) Update: $u^{k+1}(t) = Q(u^k(t) - \gamma^k d^k(t))$.
6) If u^{k+1} satisfies a terminating condition, then exit, else, increment k and repeat from step 1.

III. Single Integration Optimal Mode Scheduling

A. Open Loop Control over Finite Time Horizon

The problem of optimizing an arbitrary cost functional $J(x, u)$ subject to the switching control $u(t)$ and switching system state $x(t)$ is considered in [16]. Here, we increase the computational performance of [16] in the special case of linear time-varying systems with quadratic performance metric. In particular, we reformulate this problem so that no differential equations are solved during the iterative optimization routine. Algorithm 1 provides a summary of SIOMS.

Consider the optimization problem constrained by the system dynamics (3), as described in Section II. The dynamic constraint dictates that a system simulation should be performed at each iteration in Algorithm 1 as soon as the next switching control has been calculated. In particular, the calculation of the mode insertion gradient (9) involves the solution of the state and adjoint equations, (3) and (10), while the max-projection operator also includes the state equation (3).

We follow a similar approach to the switching time optimization approach in [24], extending it to the situation where the mode sequence Σ is unknown. Building on the existence of a linear relationship between the state and co-state as described in Section II-D, we utilize operators to formulate algebraic expressions for the calculation of the state $x(t)$ and the relation $P(t)$ at any time $t \in [T_0, T_M]$. The operators are available prior to optimization through offline solutions to differential equations. Moreover, they are independent of the mode sequence and switching times.

In the switching time optimization case [24]—where the mode sequence is constant and the problem is finite-dimensional—a single optimization iteration involves only a finite number of state and co-state evaluations; these occur at the (finite) switching times for that particular iteration. However, mode scheduling is an infinite-dimensional optimal control problem and requires the time evolution of the state and co-state trajectories at each iteration.

Therefore, in order for the proposed algorithm to be feasible, an explicit mapping from time t to x and P is needed at each iteration, depending on the current mode schedule $\{\Sigma, \mathcal{T}\}$. The mapping, below in (15) and (21), only includes algebraic expressions dependent on solutions to pre-computed differential equations. The exact number of multiplications executed in each iteration depends on how many time instances the state and co-state must be evaluated.

For the rest of the paper, a variable with the superscript k implies that the variable depends directly on u^k, the switching control at the k^{th} algorithm iteration.

a) Evaluating $x(t)$: The operators for evaluating $x(t)$ are the state-transition matrices (STM) of the N modes. Let $\Phi^j(\cdot, T_0) : \mathbb{R} \to \mathbb{R}^{n \times n}$ denote the STM for the linear mode $j \in \{1, ..., N\}$ with $A_j(t)$. The STM are the solutions to the N matrix differential equations

$$\frac{d}{dt}\Phi^j(t, T_0) = A_j(t) \cdot \Phi^j(t, T_0), \quad j = 1, ..., N \tag{14}$$

subject to the initial condition $\Phi^j(T_0, T_0) = I_n$. The following two STM properties are useful for computing the state $x(t)$ given a mode schedule $\{\Sigma, \mathcal{T}\}$. For an arbitrary STM, Φ, characterized by $A(t)$, we have [46] :

1) $x(t) = \Phi(t, \tau)x(\tau)$
2) $\Phi(t_1, t_3) = \Phi(t_1, t_2)\Phi(t_2, t_3) = \Phi(t_1, t_2)\Phi(t_3, t_2)^{-1}$.

We emphasize the importance of Property 2 in that it allows us to use a single operator for the evaluation of the state as explained in the following.

Proposition 3.1: The state $x(t)$ at all $t \in [T_0, T_M]$ depends on the mode schedule $\{\Sigma, \mathcal{T}\}$ and the STM $\Phi^j(\cdot, T_0)$ and is given by $x(t) := \chi(t, \Sigma, \mathcal{T})$ where

$$\begin{aligned} \chi(t, \Sigma, \mathcal{T}) = \sum_{i=1}^{M} \Big\{ &\Big[\mathbf{1}(t - T_{i-1}) - \mathbf{1}(t - T_i)\Big] \\ &\Phi^{\sigma_i}(t, T_0)\Phi^{\sigma_i}(T_{i-1}, T_0)^{-1} x(T_{i-1})\Big\} \\ &\text{subject to } x(T_0) = x_0, \end{aligned} \tag{15}$$

$\mathbf{1}(\cdot)$ is the step function defined in (8) and T_i , σ_i are the i^{th} switching time and corresponding active mode as defined in Section II-A.

Proof Using the STM properties 1 and 2, the state x at the i^{th} switching time is

$$x(T_i) = \overline{\Phi}(T_i, T_0)x_0 = \left[\prod_{j=i}^{1} \Phi^{\sigma_j}(T_j, T_{j-1})\right] x_0 \tag{16}$$

where $\overline{\Phi}(T_i, T_0)$ is the state-transition matrix corresponding to $\overline{A}(t, \Sigma, \mathcal{T})$ as defined in (3). Hence, the state evolution is defined as a piecewise function of time, each piece corresponding to a time interval between consecutive switching times $\{T_i, T_{i+1}\}$:

$$x(t) = \begin{cases} \Phi^{\sigma_1}(t, T_0)x(T_0), & T_0 \le t < T_1 \\ \Phi^{\sigma_2}(t, T_1)\Phi^{\sigma_1}(T_1, T_0)x(T_0), & T_1 \le t < T_2 \\ \quad\vdots & \quad\vdots \\ \Phi^{\sigma_M}(t, T_{M-1})[\prod_{j=M-1}^{1} \Phi^{\sigma_j}(T_j, T_{j-1})]x(T_0) & T_{M-1} \le t \le T_M \end{cases} \tag{17}$$

For a more compact representation of the state, we employ unit step functions and (16) to get

$$x(t) = \sum_{i=1}^{M} \left\{[\mathbf{1}(t - T_{i-1}) - \mathbf{1}(t - T_i)]\Phi^{\sigma_i}(t, T_{i-1})x(T_{i-1})\right\} \tag{18}$$

where, from STM property 2,

$$\Phi^{\sigma_i}(t, T_{i-1}) = \Phi^{\sigma_i}(t, T_0)\Phi^{\sigma_i}(T_{i-1}, T_0)^{-1}. \tag{19}$$

This concludes the proof.

Prior to the iterative optimization, the STM operators $\Phi^j(t, T_0)$ are solved offline for $t \in [T_0, T_M]$ and for all different modes $j = 1, ..., N$. Thus, given a mode schedule, the calculation of state $x(t)$ via (15) requires no additional integrations beyond the offline calculations used for $\Phi^j(t, T_0)$.

b) Evaluating $P(t)$: As proven in [24], an analogous operator to the STM exists for the evaluation of the relation $P(t)$ appearing in (11). As in [24], we will refer to the operator as the adjoint-transition matrix (ATM) and use $\Psi^j(\cdot, T_M) : \mathbb{R} \to \mathbb{R}^{n\times n}$ to denote the ATM corresponding to each mode $j \in \{1, ..., N\}$. The ATM are defined to be the solutions to the following N matrix differential equations:

$$\frac{d}{dt}\Psi^j(t, T_M) = -A_j(t)^T\Psi^j(t, T_M) - \Psi^j(t, T_M)A_j(t) - Q(t) \tag{20}$$

subject to the initial condition $\Psi^j(T_M, T_M) = 0_{n\times n}$.

The following two ATM properties will be useful for evaluating $P(t)$ given a mode schedule $\{\Sigma, \mathcal{T}\}$. For an arbitrary ATM, Ψ, characterized by $A(t)$ and associated STM Φ, and cost function defined by $Q(t)$, we have [24]:

1) $P(t) = \Psi(t, \tau) \circ P(\tau) := \Psi(t, \tau) + \Phi(\tau, t)^T P(\tau)\Phi(\tau, t)$
2) $\Psi(t_1, t_3) = \Psi(t_1, t_2) \circ \Psi(t_2, t_3) := \Psi(t_1, t_2) + \Phi(t_2, t_1)^T\Psi(t_2, t_3)\Phi(t_2, t_1)$.

Notice that Property 2 of ATM is equivalent to Property 2 of STM and similarly allows us to evaluate the co-state.

Proposition 3.2: The relation $P(t)$ at all $t \in [T_0, T_M]$ depends on the current mode schedule $\{\Sigma, \mathcal{T}\}$, the STM $\Phi^j(\cdot, T_0)$ and the ATM $\Psi^j(\cdot, T_M)$ and is given by $P(t) := \varrho(t, \Sigma, \mathcal{T})$ where

$$\begin{aligned} \varrho(t, \Sigma, \mathcal{T}) = \sum_{i=1}^{M} \Big\{ &\Big[\mathbf{1}(t - T_{i-1}) - \mathbf{1}(t - T_i)\Big]\cdot \\ &\Big[\Psi^{\sigma_i}(t, T_M) + \Phi^{\sigma_i}(t, T_0)^{-T}\Phi^{\sigma_i}(T_i, T_0)^T[P(T_i) \\ &-\Psi^{\sigma_i}(T_i, T_M)]\Phi^{\sigma_i}(T_i, T_0)\Phi^{\sigma_i}(t, T_0)^{-1}\Big]\Big\} \\ &\text{subject to } P(T_M) = P_1, \end{aligned} \tag{21}$$

$\mathbf{1}(\cdot)$ is the step function defined in (8) and T_i , σ_i are the i^{th} switching time and corresponding active mode as defined in Section II-A.

Proof From the ATM properties 1 and 2, $P(t)$ at the i^{th} switching time is

$$\begin{aligned} P(T_i) &= \overline{\Psi}(T_i, T_M) \circ P(T_M) \\ &= \overline{\Psi}(T_i, T_M) + \overline{\Phi}(T_M, T_i)^T P(T_M)\overline{\Phi}(T_M, T_i) \end{aligned} \tag{22}$$

where $\overline{\Psi}(T_i, T_M)$ is the adjoint-transition matrix corresponding to $\overline{A}(t, \Sigma, \mathcal{T})$ as defined above. From ATM property 2, this is equal to

$$\begin{aligned} \overline{\Psi}(T_i, T_M) =& \Psi^{\sigma_{i+1}}(T_i, T_{i+1}) \circ \cdots \circ \Psi^{\sigma_M}(T_{M-1}, T_M) \\ =& \sum_{m=i+1}^{M} \overline{\Phi}(T_{m-1}, T_i)^T\Psi^{\sigma_m}(T_{m-1}, T_m)\overline{\Phi}(T_{m-1}, T_i). \end{aligned} \tag{23}$$

As in the previous case, we aim to derive an expression for the evaluation of $P(t)$ at arbitrary time instances. Again, we will represent $P(t)$ as a piecewise function of time:

$$P(t) = \begin{cases} \Psi^{\sigma_M}(t, T_M) \circ P(T_M), & T_{M-1} \le t < T_M \\ \Psi^{\sigma_{M-1}}(t, T_{M-1}) \circ P(T_{M-1}), & T_{M-2} \le t < T_{M-1} \\ \quad\vdots & \quad\vdots \\ \Psi^{\sigma_1}(t, T_1) \circ P(T_1), & T_0 \le t < T_1 \end{cases} \tag{24}$$

For a more compact representation of $P(t)$, we employ unit step functions to get

$$P(t) = \sum_{i=1}^{M} \left\{[\mathbf{1}(t - T_{i-1}) - \mathbf{1}(t - T_i)][\Psi^{\sigma_i}(t, T_i) \circ P(T_i)]\right\} \tag{25}$$

where, from ATM property 2,

$$\Psi^{\sigma_i}(t, T_i) = \Psi^{\sigma_i}(t, T_M) + \Phi^{\sigma_i}(T_i, t)^T\Psi^{\sigma_i}(T_i, T_M)\Phi^{\sigma_i}(T_i, t). \tag{26}$$

Combining ATM property 1 with (21) and (26), we end up with the expression

$$\begin{aligned} P(t) = \sum_{i=1}^{M} \Big\{ &[\mathbf{1}(t - T_{i-1}) - \mathbf{1}(t - T_i)]\cdot \\ &[\Psi^{\sigma_i}(t, T_M) + \Phi^{\sigma_i}(T_i, t)^T[P(T_i) - \Psi^{\sigma_i}(T_i, T_M)]\Phi^{\sigma_i}(T_i, t)]\Big\} \end{aligned} \tag{27}$$

with $\Phi^{\sigma_i}(T_i, t) = \Phi^{\sigma_i}(T_i, T_0)\Phi^{\sigma_i}(t, T_0)^{-1}$. This completes the proof.

Prior to the iterative optimization, the ATM operators $\Psi^j(t, T_M)$ are solved offline for all $t \in [T_0, T_M]$ and for all different modes $j = 1, ..., N$. Thus, given a mode schedule, the calculation of $P(t)$ via (21) requires no additional integrations.

c) Calculating the descent direction using the mode insertion gradient: An iterative optimization method computes a new estimate of the optimum by taking a step in a search direction from the current estimate of the optimum so that a sufficient decrease in cost is achieved. The mode insertion gradient $d(t)$ defined above, has a similar role in the mode scheduling optimization as the gradient does for finite-dimensional optimization. It has been shown in [16], [41], [42] that $-d^k(t)$ is a descent direction.

Proposition 3.3: An element of $d^k(t)$ is given by

$$d_i^k(t) := \chi(t, \Sigma(u^k), \mathcal{T}(u^k))^T \varrho(t, \Sigma(u^k), \mathcal{T}(u^k)) \\ [A_i(t) - A_{\sigma^k(t)}(t)] \chi(t, \Sigma(u^k), \mathcal{T}(u^k)) \tag{28}$$

where $i = \{1, ..., N\}$.

Proof After the definition for the state and co-state, an equivalent expression for the mode insertion gradient may be obtained from (15),(21) and (9).

d) Update rule: A new estimate of the optimal switching control u^{k+1} is obtained by varying from the current iterate u^k in the descent direction and projecting the result to the set of admissible switching control trajectories. For this purpose, we employ the max-projection operator (6) and get a new estimate of the optimum,

$$\begin{aligned} u^{k+1}(t) &= Q(u^k(t) - \gamma^k d^k(t)) \\ x^{k+1}(t) &:= \chi(t, \Sigma(u^{k+1}), \mathcal{T}(u^{k+1})) \end{aligned} \tag{29}$$

where Q is given by (7). For choosing a sufficient step size γ^k, we may utilize a projection-based backtracking process as described in [18].

The reader is referred to [16], [17] for a more detailed description of these algorithm steps, along with the associated proofs for convergence.

e) Calculating the optimality condition: The optimality function $\theta^k \in \mathbb{R}$ is [16]

$$\theta^k := d_{i_0}^k(t_0) \tag{30}$$

where

$$(i_0, t_0) = \underset{i \in \{1,...,N\}, t \in [T_0, T_M]}{\arg\min} d_i(t). \tag{31}$$

The limit of the sequence of optimality functions is proven to go to zero as a function of iteration k in [16]. This allows us to utilize θ^k also as a terminating condition for the iterative algorithm.

B. A Receding-Horizon Approach

Section III-A provides an offline approach for computing an open-loop optimizer for the problem in Section II-B. Here, we follow a receding-horizon approach in order to achieve closed-loop optimization over an infinite time horizon.

Receding-horizon control strategies (often referred to as MPC strategies [11], [33], [47], [48]) have become quite popular recently, partly due to their robustness to model uncertainties or to sensor measurement noise. This paper's approach enables real-time closed-loop execution of finite-horizon optimal control algorithms. Based on our performance evaluation in the next section, the finite-horizon SIOMS is well-suited for receding-horizon linear switched-system control because it is fast and accurate.

A receding-horizon scheme for optimal mode scheduling can be implemented as follows. From the current time t and measured state $x(t)$ as the initial condition in (1), use SIOMS to obtain an optimal switching control $u_t(\tau)$ for $\tau \in [t, t + T]$ where $T := (T_M - T_0)$ in Algorithm 1. Apply the calculated control for time duration δ with $0 < \delta \leq T$ to drive the system from $x(t)$ at time t to $x(t + \delta)$. Set $t \leftarrow t + \delta$ and repeat. This scheme requires execution of the optimal mode scheduling algorithm every δ seconds.

Following Algorithm 1, SIOMS requires an offline calculation of operators before the online iterative process is executed. However, in order for SIOMS to be efficient in a receding-horizon approach, it is undesirable to recalculate each STM and ATM every δ seconds for the next T seconds. Instead, each STM and ATM of the previous time interval $[T_0, T_M]$ are updated for the new information on $[T_M, T_M + \delta]$ only (Fig. 1). Such an approach is feasible because of the following lemma.

Lemma 3.4: Suppose $\Phi(t, T_0)$ and $\Psi(t, T_M)$ are known for all $t \in [T_0, T_M]$. Assuming also that $\Phi(t, T_M)$ and $\Psi(t, T'_M)$ are known for all $t \in [T_M, T'_M]$, the STM and ATM for the time interval $t \in [T'_0, T'_M]$ with $T_0 < T'_0$ and $T_M < T'_M$ are given by

$$\Phi(t, T'_0) = \begin{cases} \Phi(t, T_0)\Phi(T'_0, T_0)^{-1}, & T'_0 \leq t < T_M \\ \Phi(t, T_M)\Phi(T_M, T_0)\Phi(T'_0, T_0)^{-1}, & T_M \leq t \leq T'_M \end{cases} \tag{32}$$

and

$$\Psi(t, T'_M) = \begin{cases} \Psi(t, T_M) \circ \Psi(T_M, T'_M), & T'_0 \leq t < T_M \\ \Psi(t, T'_M), & T_M \leq t \leq T'_M. \end{cases} \tag{33}$$

Proof The proof of Lemma 3.4 is a straightforward consequence of STM property 2 and ATM properties 1 and 2 in Section III-A.

Despite its simplicity, Lemma 3.4 is the key to efficient real-time execution of a receding-horizon hybrid control scheme. Using Lemma 3.4 with $T'_0 = T_0 + \delta$ and $T'_M = T_M + \delta$,

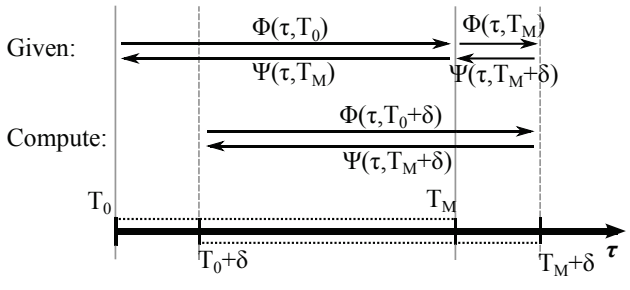

Fig. 1. An illustration of the operators update step in a receding-horizon scheme. A differential equation needs to be integrated only over a limited time interval δ rather than the time horizon $(T_M - T_0) := T$.

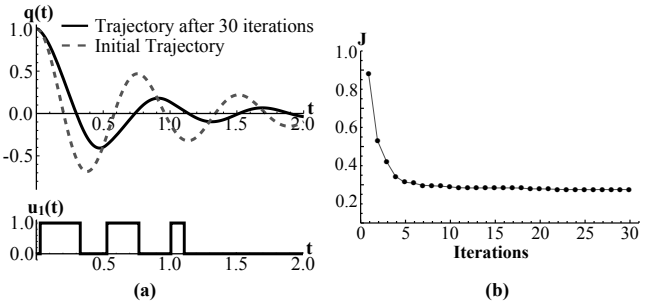

Fig. 2. Spring-Mass-Damper vibration control: (a) Optimal trajectory and switching control and (b) the cost versus iteration count.

we formulate Algorithm 2 that allows for real-time closed-loop SIOMS execution. The proposed formulation requires a numerical integration over the limited time interval δ rather than the full time horizon T (step 3.1 in Algorithm 2). A graphical representation of the operators update every δ seconds (step 3 in Algorithm 2) is given in Fig. 1.

As mentioned in the introduction, this paper does not address stability of the receding-horizon SIOMS algorithm. Many papers offer a detailed stability analysis for receding-horizon algorithms, e.g. [33], [49], [50], with only a few focusing on switched systems in particular [51], [52]. The latter establish stability criteria that rely on hysteresis and dwell-time conditions. However, as is obvious from our main example in Section V, we wish to give special consideration to applications where SIOMS is used for system control with hybrid inputs by expressing the problem in a switched system framework. The foundation of a stability proof for this SIOMS implementation is given in [53] where authors provide stability conditions on the design parameters of model predictive control algorithms with input discontinuities. In short, conditions are imposed on the time horizon T, the terminal and running cost in (4) and the terminal constraint set.

Algorithm 2 Receding-Horizon SIOMS

- Initialize current time t, finite horizon T and control duration δ.
- Solve for $\Phi^j(\tau, t)$ and $\Psi^j(\tau, t+T)$ $\forall j \in \{1, ..., N\}$ and $\tau \in [t, t+T]$.

Do every δ seconds while control $u_t(\tau)$ is applied:

1. Update $T_0 \leftarrow t$, $T_M \leftarrow t+T$ and set $x(T_0) = x(t)$.
2. Run online part of Algorithm 1 to get $u_t(\tau)$ for $\tau \in [t, t+\delta]$.
   - **3**.1 Solve for $\Phi^j(\tau, T_M)$ and $\Psi^j(\tau, T_M+\delta)$ $\forall \tau \in [T_M, T_M+\delta]$. *
   - **3**.2 Get $\Phi^j(\tau, T_0+\delta)$ and $\Psi^j(\tau, T_M+\delta)$ $\forall j \in \{1, ..., N\}$ and $\tau \in [T_0+\delta, T_M+\delta]$ from known $\Phi^j(\tau, T_0)$ and $\Psi^j(\tau, T_M)$ using Lemma 3.4. *
   - **3**.3 Update $\Phi(\tau, T_0) \leftarrow \Phi(\tau, T_0+\delta)$ and $\Psi(\tau, T_M) \leftarrow \Psi(\tau, T_M+\delta)$. *

* In a real-time application, step 3 can be executed at any time when processing requirements are low, without increasing the amount of time needed for calculation of control (i.e. steps 1-2).

IV. Open-Loop Implementation and Evaluation

In this section, SIOMS is implemented in a standard open-loop manner (see Algorithm 1) and its performance is evaluated in terms of i) execution time, ii) error of approximation and iii) computational complexity.

As a baseline example, we use SIOMS to apply switched stiffness vibration control on an unforced spring-mass-damper system. A linear time-invariant system is particularly suited for evaluation purposes as an analytical solution exists and can be compared with the computed numerical solution. Variants of this example system have been used extensively in literature for the evaluation of hybrid controllers [54], [55]. Denoting by k_i the variable spring stiffness and by m and d the mass and damping coefficient, the system equations take the form in (2) with

$$A_i(t) = \begin{pmatrix} 0 & 1 \\ -\frac{k_i}{m} & -\frac{d}{m} \end{pmatrix} \tag{34}$$

and $N = 2$ i.e. two possible modes. The state vector is $x = [q(t), \dot{q}(t)]^T$, where $q(t)$ is the mass position. System parameters are defined as $m = 1$, $d = 2$, $k_1 = 30$, and $k_2 = 70$. Our objective is to find the mode schedule that minimizes the system vibration and is accordingly characterized by the quadratic cost functional (4) with $Q = diag[1, 0.1]$, $P_1 = 0_{2\times 2}$ and $[T_0, T_M] = [0, 2]$. As an initial estimate $u^0(t)$, the system is in mode 2 with an initial condition $x_0 = [1, 0]^T$ and cost $J_0 \approx 0.98$.

Fig. 2a shows the optimal switching control and corresponding optimal $q(t)$ trajectory after 30 SIOMS iterations. The cost is reduced to $J \approx 0.38$ (Fig. 2b).

A. Execution Time and Approximation Error

The execution time of iterative optimal control algorithms might be prohibitive for real-time applications [30]. It is often the case that appropriate numerical techniques for integrating the state and adjoint equations, (1) and (10), improve execution times. However, there is a trade-off to consider—a fast numerical ODE solver might be prone to approximation errors. In open-loop SIOMS (Algorithm 1), no differential equations need to be numerically solved as part of the online iterative process. Hence, we will show that both the online execution time and approximation error can be kept low at the same time.

Referring to Algorithm 1, a set of operator trajectories is pre-calculated and stored offline, covering the full time horizon $[T_0, T_M]$. In practice, the exact number of stored samples $\mathcal{N}$ needs to be determined to reflect the processor's computational capacity and memory availability.[3] We use the mass-spring-damper to illustrate how the SIOMS execution time and error of approximation vary across different choices of sample sizes (Fig. 3).

For comparison purposes, we additionally evaluate the performance of the projection-based mode scheduling algorithm in [16] using the same example employing two different numerical techniques, namely the Forward and Improved Euler methods, for the integration of the state and adjoint equations. In contrast to SIOMS where expressions exist for the state and co-state evaluation ((15) and (21)), here, the solution to the state and co-state equations, (2) and (10), is approximated in every algorithm iteration. The Forward Euler method pro-

[3]Interpolating methods may be used for intermediate time instances.

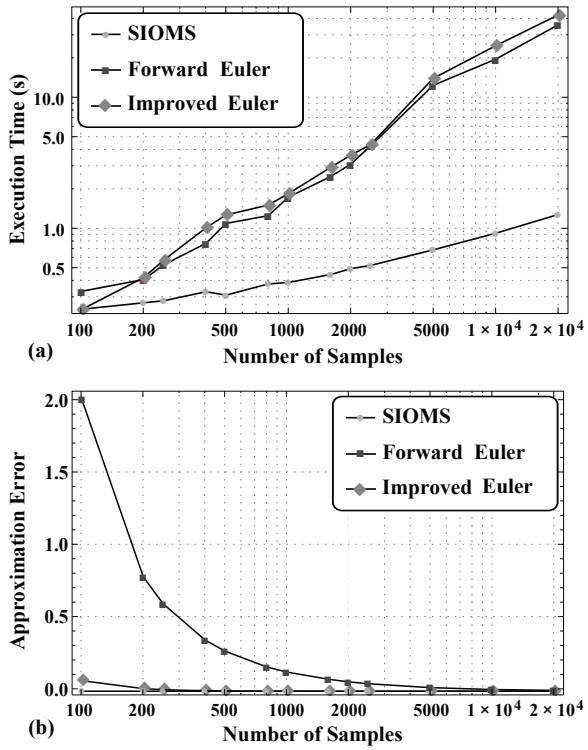

Fig. 3. Variation of (a) online execution times and (b) approximation errors (2-norm of the root-mean-squared differences between the analytic and computed state values) with respect to the selected number of samples evaluated across 3 different optimization methods. SIOMS can achieve both objectives (i.e. fast execution and high approximation accuracy) for a wide range of sample sizes.

vides the following approximation to the state and co-state trajectories of a general linear time-varying switched system:

$$\begin{aligned} x(t_{h+1}) &= (\mathbb{I} + \Delta t \cdot \overline{A}(t_h, \Sigma, \mathcal{T}))x(t_h), \qquad x(t_0) = x_0 \\ \rho(t_h) &= (\mathbb{I} + \Delta t \cdot \overline{A}(t_{h+1}, \Sigma, \mathcal{T}))^T \rho(t_{h+1}) + \Delta t \cdot Q x(t_{h+1}), \qquad \rho(t_N) = P_1 x(t_N) \end{aligned} \tag{35}$$

where $\mathbb{I}$ is the $n \times n$ identity matrix, $t_{h+1} = t_h + \Delta t$ with Δt the step size and $\overline{A}(t, \Sigma, \mathcal{T})$ is defined in (3). The Euler method is simple but can be unstable and inaccurate. On the other hand, Improved Euler (i.e. two-stage Runge Kutta) maintains simplicity but with reduced approximation errors. It applies the following approximation:

$$\begin{aligned} x(t_{h+1}) &= \Big[\mathbb{I} + \frac{\Delta t}{2} A(t_h) + \frac{\Delta t}{2} A(t_{h+1})(\mathbb{I} + \Delta t \cdot A(t_h))\Big] x(t_h), \qquad x(t_0) = x_0 \\ \rho(t_h) &= \Big[\mathbb{I} + \frac{\Delta t}{2} A(t_{h+1})^T + \frac{\Delta t}{2} A(t_h)^T (\mathbb{I} + \Delta t \cdot A(t_{h+1})^T)\Big] \rho(t_{h+1}) \\ &\quad + \Delta t (\mathbb{I} + \frac{\Delta t}{2} A(t_h)^T) Q x(t_{h+1}), \qquad \rho(t_N) = P_1 x(t_N) \end{aligned} \tag{36}$$

where $\mathbb{I}$ is the $n \times n$ identity matrix and $t_{h+1} = t_h + \Delta t$ with Δt the step size. It is assumed for notational simplicity that $A(\cdot) := \overline{A}(\cdot, \Sigma, \mathcal{T})$ defined in (3). Notice that both approximation methods for the state and co-state, (35) and (36), depend on the step size Δt as opposed to SIOMS state and co-state expressions (15) and (21) that are independent of a step size.

In the following example, the execution time and error of approximation are measured against the selected number of samples $\mathcal{N}$. The step size Δt is constant so that the samples are evenly-spaced. For the Forward and Improved Euler methods, the number of samples $\mathcal{N}$ determines the fixed step size Δt used for online integration of (1) and (10) resulting in the approximations (35) and (36). However, in SIOMS the number of samples $\mathcal{N}$ does not determine the step size used in the offline numerical integration—instead, the STM and ATM equations, (14) and (20), are numerically solved[4] and the resulting trajectories are sub-sampled with the desired sampling frequency $1/\Delta t$ to create the final stored data points. Note that we are only able to perform this additional sub-sampling interpolation because it does not affect the total execution time of the online algorithm portion. The fact that the sub-sampling process is applied on smooth trajectories produced by the continuous vector fields in (14) and (20)—along with the fact that the expressions for evaluating the state and co-state, (15) and (21), do not depend on any discretization step size—guarantees that approximation accuracy of each sample does not drop as the number of samples decreases.[5] Regardless of the particular choice of Δt, the role of Δt has the same impact on all three representations of state and co-state evolution (SIOMS, Forward and Improved Euler)—in each case, Δt determines the number of samples $\mathcal{N}$ (that can be) available (without interpolation) during each iteration. All methods were implemented in MATLAB, on a laptop with an Intel Core i7 chipset.

The results are summarized in Fig. 3. Figure 3a illustrates the variation of online execution time with respect to the selected number of samples. Execution time refers to the number of seconds required for 10 algorithm iterations—no significant change in cost is observed in subsequent iterations as seen in Fig. 2b. In all cases, the final optimal cost was found to be in the range 0.45-0.5. Both Euler methods exhibit a similar rising trend with the execution time reaching a maximum of 13 seconds when 20,001 samples are used (i.e. step size of 0.0001 secs). With SIOMS, however, a significantly lower increase rate is observed with a maximum online execution time at only approximately 1.3 seconds. The reasoning for this observed difference is that with Euler methods, all samples of the state and co-state trajectories must be calculated in every iteration whereas in SIOMS one only needs to calculate the state and co-state values necessary for the procedures of the algorithm (e.g. computation of new switching times) using the expressions (15) and (26) respectively.

The variation of approximation error with respect to the number of samples is depicted in Fig. 3b. Here, by approximation error we refer to the 2-norm of the root-mean-squared (RMS) differences between the analytic and computed state

[4]For this example, equations (14) and (20) are solved by a fixed-step Improved Euler's method (i.e. two-stage Runge Kutta) with step size equal to 10^{-4}.

[5]Choosing to use interpolating methods might be concerning with regard to approximation accuracy of the full state and co-state trajectories. Regardless, there are two ways to keep approximation errors low: i) by using higher-order interpolating methods and ii) by using a larger number of samples. With SIOMS, we can select a large number of samples without dramatically increasing the execution time (Fig. 3).

values for all states at sample points. As explained earlier, the error with SIOMS remains approximately zero (≈ 0.0002) regardless of the sample size. The trade-off between computation time and approximation error is particularly obvious with the Forward Euler's method, where the error only approaches zero when a maximum number of samples is employed by which time the corresponding execution time is prohibitive. Interestingly, Improved Euler's method starts with a lower error (≈ 0.07) and drops to its minimum value of ≈ 0.0002 when 1600 samples and above are used. With the lowest approximation error (≈ 0.0002), Improved Euler can achieve a minimum execution time of approximately 3 seconds compared to 0.2 seconds achieved by SIOMS. With low execution time ($\approx$ 0.2 with 100 samples used), Improved Euler can achieve a minimum error close to 0.1 compared to 0.0002 achieved by SIOMS.

B. Computational Complexity

In Section III, we showed that all the state and co-state information needed in Algorithms 1 and 2, is encoded in the STM, $\Phi^j(t, T_0)$, and ATM, $\Psi^j(t, T_M)$, $\forall j \in \{1, ..., N\}$ which are solved for all $t \in [T_0, T_M]$ prior to the optimization routine. Therefore, the calculation of $x^k(t)$ and $P^k(t)$ and consequently the optimality condition θ^k relies simply on memory calls and matrix algebra. No additional differential equations need to be solved for during optimization.

The algorithm complexity can be discussed in terms of the number of matrix multiplications involved in each iteration. Recall that at each iteration, $x(t)$ is given by (15) and $P(t)$ by (21), but the total number of state and co-state evaluations depends on the number of time instances the descent direction (28) must be evaluated (e.g. for the calculation of θ^k in (31)). Taking this into consideration, we will look at the algebraic calculations required for the evaluation of the state, co-state and descent direction at a single time instance t.

First, for executional efficiency, one may calculate all the state and co-state values at the switching times, $x(T_i)$ and $P(T_i)$, given the current mode schedule $(\Sigma(u^k), \mathcal{T}(u^k))$ at the beginning of each iteration. To compute the state, begin with $x(T_0) = x_0$ and then recursively calculate

$$x(T_i) = \Phi^{\sigma_i}(T_i, T_{i-1})x(T_{i-1}) \forall i \in \{1, ..., M-1\}. \tag{37}$$

Using STM property 2 and following a similar approach as in the derivation of (15), this computation comes down to $2(M-1)$ matrix multiplications, assuming that all $\Phi^j(t, T_0)^{-1}$ for all $j \in \{1, ..., N\}$ have also been stored in memory. Similarly, begin with $P(T_M) = P_1$ and then recursively calculate

$$\begin{aligned} P(T_i) =& \Psi^{\sigma_{i+1}}(T_i, T_M) + \Phi^{\sigma_{i+1}}(T_{i+1}, T_i)^T \\ & [P(T_{i+1}) - \Psi^{\sigma_{i+1}}(T_{i+1}, T_M)]\Phi^{\sigma_{i+1}}(T_{i+1}, T_i) \end{aligned} \tag{38}$$

for all $i \in \{1, ..., M-1\}$. Note that the derivation of the above expression is identical to the derivation of (21). Knowing that all $\Phi^{\sigma_{i+1}}(T_{i+1}, T_i)$ have already been calculated in (37), another $2(M-1)$ multiplications are required for the calculation of $P(t)$. To summarize, the standard computational cost of the algorithm comes down to a total of $4(M-1)$ multiplications per iteration.

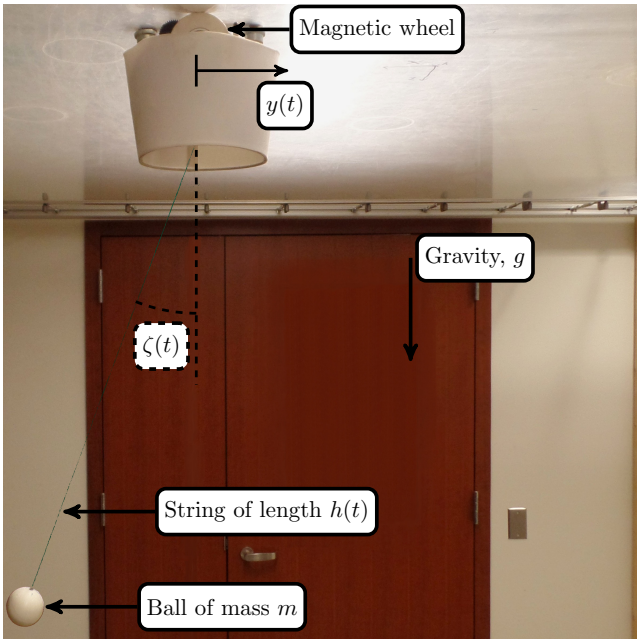

Fig. 4. The experimental setup consists of an one-dimensional differential drive mobile robot with magnetic wheels (i.e. cart) and a ball suspended by a string. The string changes length by means of an actuated reeling system attached on the robot. The system configuration is measured by a Microsoft Kinect at $\approx$ 30 Hz. The full state is estimated using an Extended Kalman Filter. The Robot Operating System (ROS) is used for collecting sensed data and transmitting control signals (i.e. robot acceleration values). See more in [56], [57].

Now, to evaluate equation (15) and (21) at any random time t during the optimization process, we only need 6 additional multiplications, 2 for the state $x(t)$ and 4 for the relation $P(t)$. Therefore, to evaluate the descent direction at any random time, 9 multiplications are required in total, including the algebra involved in (28).

Finally, each iteration of Algorithm 1 involves $4(M-1)$ multiplications for the calculation of $x(T_i)$ and $P(T_i)$, and 9λ additional multiplications where λ is the number of evaluations of the expression (28) for the descent direction.

V. Closed-Loop Simulation and Experimental Implementation

In this section, SIOMS is implemented in a closed-loop manner (see Algorithm 2) and is tested on a cart and suspended mass system in simulation and on an experimental setup.

The system model under concern is linear time-varying with two configuration variables, $q(t) = [y(t), \zeta(t)]^T$, where $y(t)$ is the horizontal displacement of the cart and $\zeta(t)$ is the rotational angle of the string as seen in Fig. 4. The length of the string varies with time. Denoting by $h(t)$ the time-varying string length, g the gravity acceleration and by m and c the mass and damping coefficient, the linearized system equations around the equilibrium $x = \mathbf{0}$, take the form in (2) with

$$A_i(t) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\alpha_i \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{g}{h(t)} & -\frac{c}{mh(t)^2} & -\frac{1}{h(t)}\alpha_i \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \tag{39}$$

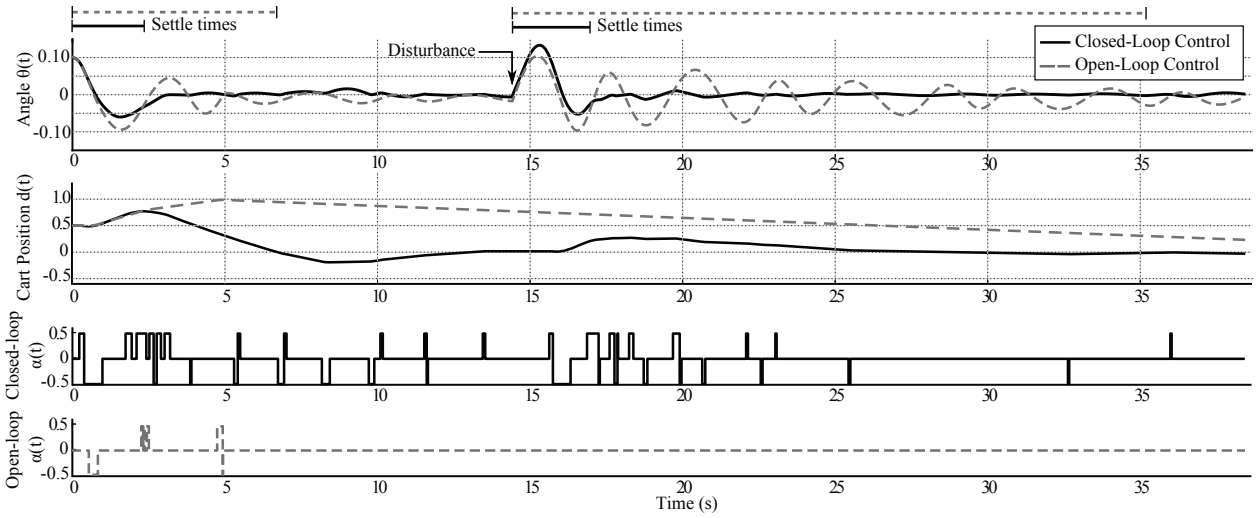

Fig. 5. Open-loop SIOMS (Algorithm 1 with $T_M = 40\ s$) vs Closed-loop SIOMS (Algorithm 2 with $\delta = 0.5$ and $T = 3\ s$) in simulation.

$$h(t) = sin(t) + 2 \tag{40}$$

and $N = 3$ i.e. three possible modes. The cart's horizontal acceleration α is directly controlled and can switch between the values $\alpha_1 = 0$, $\alpha_2 = -0.5$ and $\alpha_3 = 0.5$. Notice we have augmented the state-space from $\mathbb{R}^4$ to $\mathbb{R}^5$ in order to transform the originally affine model to the linear form in (2). The augmented state vector is $x = [y, \dot{y}, \zeta, \dot{\zeta}, \tilde{u}]$ where $\tilde{u}$ is the auxiliary state variable. System parameters are defined as $m = 0.124\ kg$, $c = 0.05$ and $g = 9.8\ m/s^2$.

Our objective is to find the mode schedule that minimizes the angle oscillation while the cart remains in a neighborhood near the origin and is accordingly characterized by the quadratic cost functional (4) with $Q = diag[0, 0, 10, 1, 0]$ and $P_1 = diag[0.1, 0.01, 10, 1, 0]$. The system starts at an initial condition $x_0 = [0.5, 0, 0.1, 0, 1]^T$.

A. Simulation Results

We apply Algorithm 2 to the optimal control problem stated previously and compare its performance with Algorithm 1 in terms of (1) disturbance rejection and (2) robustness to system parameter uncertainties. For real-time SIOMS execution, both algorithms were implemented in Python.

f) Disturbance rejection: We ran Algorithm 2 with parameters $\delta = 0.5$ and $T = 3\ s$ for a total of 40 seconds. A disturbance is applied at time $\approx 14\ s$. Each run of Algorithm 1 (i.e. 5 SIOMS iterations) lasted on average $0.04\ s$ of CPU time. Note that the algorithm was implemented in a real-time manner—starting the system simulation/integration from $t = 0\ s$, a new switching control is calculated and applied every $\delta = 0.5$ seconds using information about the current system state. For comparison, we additionally ran a one-time open-loop SIOMS (Algorithm 1) with $T_0 = 0\ s$ and $T_M = 40\ s$. The cost is reduced from $J_0 \approx 1.96$ to $J \approx 0.58$ after 15 iterations; the optimal switching control was pre-calculated and later applied to the system.

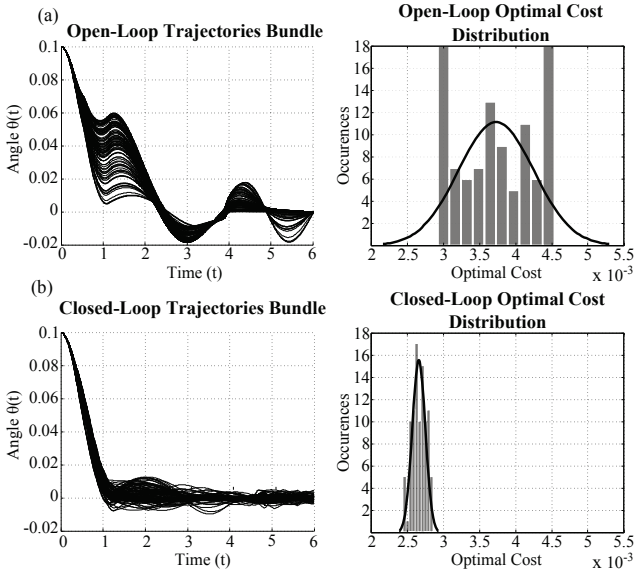

Fig. 6. Robustness to uncertainty in the damping coefficient through Monte-Carlo analysis. Angle trajectories bundle and optimal cost distribution for (a) open-loop SIOMS (Algorithm 1 with $T_0 = 0$ and $T_M = 6$) and (b) closed-loop SIOMS (Algorithm 2 with $\delta = 0.2$ and $T = 3$).

The results are illustrated in Fig. 5. Starting at an initial value of 0.1 *rad*, the angle has a settle time[6] of about 2.5 *s* with closed-loop control compared to 5 *s* when open-loop SIOMS is applied. As expected, the disturbance triggers a high angle oscillation with a settle time > 20 *s*, as the effect is not taken into account by the open-loop controller. The receding-horizon SIOMS, however, results in a much lower settle time of 2.5 *s*, providing an efficient real-time response to the random disturbance. The last 2 diagrams in Fig. 5 show the switched cart acceleration α with respect to time as

[6]Settle time is defined here as the time from the arrival of the disturbance until the angle reaches and stays within the settle boundary from -0.025 *rad* to 0.025 *rad* surrounding the origin.

calculated by each algorithm. In close-loop control where the most reliable performance is observed, a total of 65 switches occur with an average mode duration of $\approx 0.42\ s$ and a minimum mode duration (i.e. period during which the mode remain fixed) of $\approx 0.02\ s$.

g) Robustness to model uncertainties: In a subsequent comparison, we examine the robustness of Algorithm 2 to model uncertainties and compare its performance to Algorithm 1. In particular, we perform a Monte-Carlo analysis where both algorithms are run 100 times in the following scheme: the optimal switching control is calculated using the system model in (39) and is subsequently applied to an equivalent system with randomly added noise in the damping parameter, i.e. $c_{actual} = 0.05 + \omega$ where ω is a random real number in the range $[-0.05, 3.0]$ so that $c_{actual} \in [0, 3.05]$.

We demonstrate the results in Fig. 6. The diagrams on the left show the resulting angle trajectories for $t \in [0, 6]$ of all algorithm runs. It can be observed that open-loop SIOMS is more sensitive to changes in the damping coefficient compared to closed-loop SIOMS that exhibits a more robust performance. The distribution of optimal costs across all runs is given in the remaining diagrams of Fig. 6. For both open-loop and closed-loop SIOMS, the optimal cost is calculated as in (4) over the resulting trajectories $x(t)$ for all $t \in [0, 6]$. The mean optimal cost in open-loop SIOMS is ≈ 0.0037 compared to ≈ 0.0027 in the closed-loop implementation. In addition, with receding-horizon SIOMS (Algorithm 2) the standard deviation is $0.08 \cdot 10^{-3}$ which is significantly lower than the standard deviation $0.51 \cdot 10^{-3}$ observed in open-loop SIOMS (Algorithm 1).

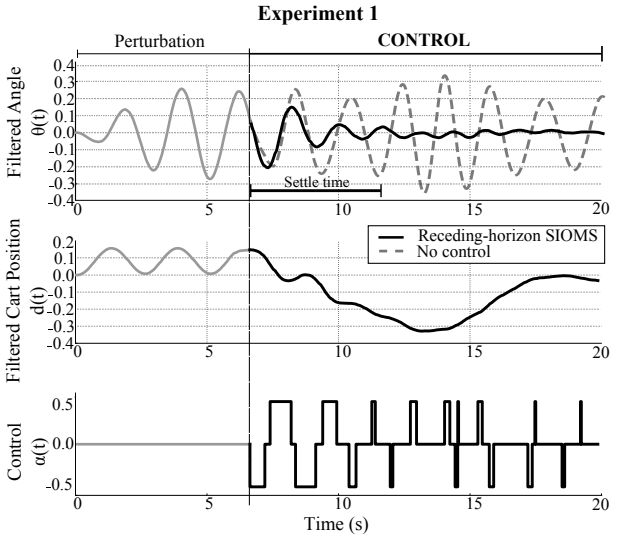

Fig. 7. An example trial of Experiment 1. First, the robot follows a sinusoidal trajectory perturbing the string angle. Approximately 6.6 seconds later, receding-horizon SIOMS is applied in real time and drives the angle back to the origin in approximately 4.8 seconds. Without control, the angle exhibits high oscillations with minimal decay.

B. Experimental Results

In this section, the performance of the closed-loop hybrid controller (Algorithm 2) is evaluated experimentally on a real cart and suspended mass system (Fig. 4). More information about this experimental platform can be found in [56], [57]. Due to geometric constraints and model discrepancies, a few changes in the parameters were made as follows: $h(t) = 0.4sin(t) + 1$ in (40), $c = 0.001$ in (39), $\delta = 0.4$ and $T = 5\ s$ in Algorithm 2. The same objective as in simulation was pursued i.e. real-time angle regulation with the robot position maintained close to the origin. The weight matrices in (4) were set as $Q = diag[0, 0, 1000, 0, 0]$ and $P_1 = diag[1, 0, 100, 0, 0]$. We ran 2 sets of experiments to illustrate the features of the hybrid controller based on Algorithm 2.

In Experiment 1, the SIOMS controller is initially inactive and we perturb the string angle by setting a predefined oscillatory trajectory to the cart/robot[7]. After approximately 6.6 seconds, the controller is activated to optimally drive the angle to zero using Algorithm 2. One example trial of Experiment 1 is illustrated in Fig. 7. During the perturbation, the angle exhibits an oscillatory response with peak amplitude at 0.25 *rad*. Once receding-horizon SIOMS is applied, the string angle starts approaching the origin with a settle time of 4.8 seconds and the robot moves slightly to the left before returning to the origin. For comparison purposes, Fig. 7 also shows the angle trajectory for the case when no control was applied following the perturbation (i.e. $\alpha(t) = 0$). One may observe that the uncontrolled system is highly underdamped with no settle time achieved in a time horizon of ≈ 14 seconds. Note that the sinusoidal change in peak amplitude and frequency is a result of the time-varying string length.

We repeated Experiment 1 for four different perturbation levels, each characterized by the peak angle amplitude achieved. Three trials per perturbation were run i.e. twelve trials in total. As performance metrics, we used a) the number of switches per second, b) the average mode duration and c) the settle time for the string angle. Our goal was to verify the reliability and efficacy of the controller in noisy conditions induced by sensor and model deficiencies. The results are given in Table I. Throughout the trials, the number of switches per second ranges from 2.3 to 3.8. The average mode duration also exhibits low variation among different trials with a range from 0.27 to 0.38 seconds. As expected, settle times increase with higher perturbation levels but remain fairly close among trials of the same perturbation.

In Experiment 2, we sought to evaluate the performance of the hybrid controller when random disturbances occur in real time. To achieve this, the experiment is initialized at $y = 0$, $\zeta = 0$, $\alpha = 0$ and zero velocities. With the receding-horizon SIOMS controller activated, a person pushes the suspended mass to create real-time disturbances. The controller responds to the disturbance to regulate the angle and drive it back to zero[8]. An example trial of Experiment 2 is presented in Fig. 8 where four consecutive disturbances of varied amplitudes are applied. One may observe that the controller regulates the

[7]A video of the experiment is available in *https://vimeo.com/nxrlab/sioms1*.
[8]A video of the experiment is available in *https://vimeo.com/nxrlab/sioms2*.

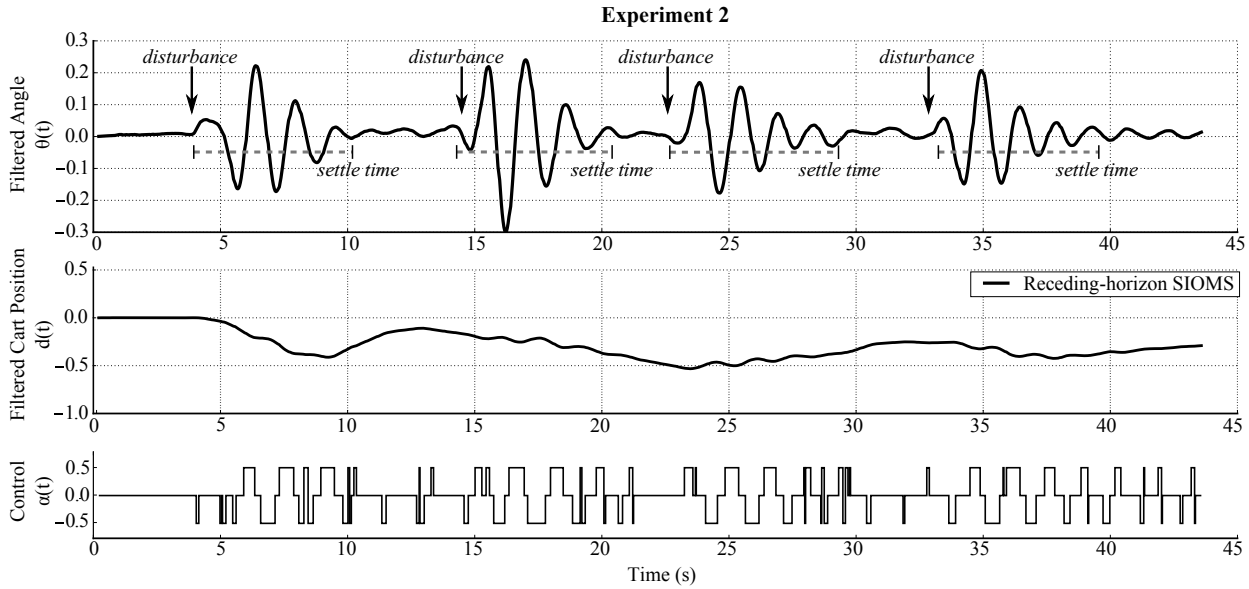

Fig. 8. An example trial of Experiment 2. SIOMS controller is always active while a person applies random disturbances by pushing the suspended ball four times sequentially. The controller reacts in real time to regulate the angle. Approximate settle time is at 6 seconds.

TABLE I
WE RAN 12 TRIALS OF EXPERIMENT 1 WITH 4 DIFFERENT PERTURBATION LEVELS.

	Perturbation 1 peak angle = 0.18*rad*			**Perturbation 2** peak angle = 0.25*rad*			**Perturbation 3** peak angle = 0.33*rad*			**Perturbation 4** peak angle = 0.4*rad*		
Trial	*1*	*2*	*3*	*1*	*2*	*3*	*1*	*2*	*3*	*1*	*2*	*3*
switches / second	2.80	3.20	2.60	2.36	2.61	2.80	3.88	2.77	2.46	2.89	2.59	2.66
average mode duration (s)	0.34	0.27	0.32	0.31	0.35	0.32	0.38	0.33	0.35	0.31	0.32	0.34
settle time (s)	2.9	3.6	4.2	3.4	4.2	4.8	7.2	7.5	6.9	9.4	8.6	9.1

angle with settle times of approximately 6 seconds in all four cases. Furthermore, as a result of the terminal cost applied at $y(t)$, the robot does not deviate significantly from the origin.

VI. CONCLUSIONS

Our objective in this paper is to achieve fast and consistent, real-time mode scheduling by taking advantage of linearity of a switched system. In general, mode scheduling is challenging due to the fact that both the mode sequence and the set of switching times must be optimized jointly. Thus, execution time of an optimization is often prohibitive for real-time applications and can only be reduced at the expense of approximation accuracy. In addition, the numerical implementation of optimal mode scheduling algorithms requires consistent solution approximations that are prone to numerical errors due to discontinuities of the switched system under concern.

We addressed these issues by introducing an algorithm (SIOMS) for scheduling the modes of linear time-varying switched systems subject to a quadratic cost functional. By solving a single set of differential equations offline, open-loop SIOMS requires no online simulations while closed-loop SIOMS only involves an integration over a limited time interval rather than the full time horizon. The proposed algorithm is fast and free of the trade-off between execution time and approximation errors. Furthermore, in practical implementation, the proposed solution of the state and adjoint equations is independent of the selected step size. For this reason, approximation accuracy and consistency in SIOMS does not depend on the number of samples used for approximation of the state and co-state trajectories. We verified the aforementioned advantages using a numerical example and comparing SIOMS to algorithms that include common integration schemes. Finally, to verify the efficacy of receding-horizon SIOMS in real-world applications, we performed a real-time experiment using ROS. Our experimental work demonstrated that a cart and suspended mass system can be regulated in real time using closed-loop hybrid control signals. Future work will focus on formally establishing stability of the receding-horizon SIOMS controller, based on results in [53].

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under awards IIS-1426961. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Y. Wardi and M. Egerstedt, "Algorithm for optimal mode scheduling in switched systems," in *American Control Conference*, 2012, pp. 4546–4551.

[2] S. Wei, K. Uthaichana, M. Žefran, R. A. DeCarlo, and S. Bengea, "Applications of numerical optimal control to nonlinear hybrid systems," *Nonlinear Analysis: Hybrid Systems*, vol. 1, no. 2, pp. 264–279, 2007.

[3] T. Johansen, I. Petersen, J. Kalkkuhl, and J. Lüdemann, "Gain-scheduled wheel slip control in automotive brake systems," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 6, pp. 799–811, 2003.

[4] R. O'Flaherty and M. Egerstedt, "Low-dimensional learning for complex robots," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 19–27, 2015.

[5] S. Lee and V. Prabhu, "A dynamic algorithm for distributed feedback control for manufacturing production, capacity, and maintenance," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 628–641, 2015.

[6] Y. Qiao, N. Wu, and M. Zhou, "Real-time scheduling of single-arm cluster tools subject to residency time constraints and bounded activity time variation," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 3, pp. 564–577, 2012.

[7] S. Mariéthoz, S. Almér, M. Bâja, A. Beccuti, D. Patino, A. Wernrud, J. Buisson, H. Cormerais, T. Geyer, H. Fujioka, U. Jonsson, C.-Y. Kao, M. Morari, G. Papafotiou, A. Rantzer, and P. Riedinger, "Comparison of hybrid control techniques for buck and boost DC-DC converters," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 5, pp. 1126–1145, 2010.

[8] S. Bhattacharya, A. Khanafer, and T. Basar, "Switching behavior in optimal communication strategies for team jamming games under resource constraints," in *IEEE International Conference on Control Applications*, 2011, pp. 1232–1237.

[9] M. Kamgarpour, W. Zhang, and C. J. Tomlin, "Modeling and optimization of terminal airspace and aircraft arrival subject to weather uncertainties," in *AIAA Guidance, Navigation and Control Conference*, 2011, pp. 6516–6521.

[10] H. Gonzalez, R. Vasudevan, M. Kamgarpour, S. S. Sastry, R. Bajcsy, and C. Tomlin, "A numerical method for the optimal control of switched systems," in *IEEE Conference on Decision and Control*, 2010, pp. 7519–7526.

[11] K. Deng, Y. Sun, S. Li, Y. Lu, J. Brouwer, P. Mehta, M. Zhou, and A. Chakraborty, "Model predictive control of central chiller plant with thermal energy storage via dynamic programming and mixed-integer linear programming," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 565–579, 2015.

[12] D. Gorges, M. Izak, and S. Liu, "Optimal control and scheduling of switched systems," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 135–140, 2011.

[13] Y. Wardi, M. Egerstedt, and M. Hale, "Switched-mode systems: Gradient-descent algorithms with Armijo step sizes," *Discrete Event Dynamic Systems*, pp. 1–29, 2014.

[14] B. Passenberg, P. E. Caines, M. Sobotka, O. Stursberg, and M. Buss, "The minimum principle for hybrid systems with partitioned state space and unspecified discrete state sequence," in *IEEE Conference on Decision and Control*, 2010, pp. 6666–6673.

[15] M. Zhang and S. Bhattacharya, "Scheduling and motion planning for autonomous grain carts," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 3422–3427.

[16] T. M. Caldwell and T. D. Murphey, "Projection-based optimal mode scheduling," in *IEEE Conference on Decision and Control*, 2013, pp. 5307–5314.

[17] ——, "Projection-based switched system optimization," in *American Control Conference*, 2012, pp. 4552–4557.

[18] ——, "Projection-based switched system optimization: Absolute continuity of the line search," in *IEEE Conference on Decision and Control*, 2012, pp. 699–706.

[19] Z. Sun and S. S. Ge, "Analysis and synthesis of switched linear control systems," *Automatica*, vol. 41, no. 2, pp. 181–195, 2005.

[20] W. Zhang and J. Hu, "On optimal quadratic regulation for discrete-time switched linear systems," in *International Conference on Hybrid Systems: Computation and Control*. Springer, 2008, pp. 584–597.

[21] Y. Kouhi, N. Bajcinca, and R. G. Sanfelice, "Suboptimality bounds for linear quadratic problems in hybrid linear systems," in *IEEE European Control Conference*, 2013, pp. 2663–2668.

[22] A. Giua, C. Seatzu, and C. Van Der Mee, "Optimal control of switched autonomous linear systems," in *IEEE Conference on Decision and Control*, vol. 3, 2001, pp. 2472–2477.

[23] X. Xu and P. J. Antsaklis, "Optimal control of switched autonomous systems," in *IEEE Conference on Decision and Control*, vol. 4, 2002, pp. 4401–4406.

[24] T. M. Caldwell and T. D. Murphey, "Single integration optimization of linear time-varying switched systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1592–1597, 2012.

[25] A. Mavrommati and T. D. Murphey, "Single-integration mode scheduling for linear time-varying switched systems," in *American Control Conference*, 2014, pp. 3948–3953.

[26] M. J. Neely, "Energy optimal control for time-varying wireless networks," *IEEE Transactions on Information Theory*, vol. 52, no. 7, pp. 2915–2934, 2006.

[27] M. Fanti, G. Iacobellis, A. Mangini, and W. Ukovich, "Freeway traffic modeling and control in a first-order hybrid Petri net framework," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 1, pp. 90–102, 2014.

[28] Q. Duan, J. Zeng, K. Chakrabarty, and G. Dispoto, "Real-time production scheduler for digital-print-service providers based on a dynamic incremental evolutionary algorithm," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 701–715, 2015.

[29] R. Vasudevan, H. Gonzalez, R. Bajcsy, and S. S. Sastry, "Consistent approximations for the optimal control of constrained switched systems—Part 1: A conceptual algorithm," *SIAM Journal on Control and Optimization*, vol. 51, no. 6, pp. 4463–4483, 2013.

[30] ——, "Consistent approximations for the optimal control of constrained switched systems—Part 2: An implementable algorithm," *SIAM Journal on Control and Optimization*, vol. 51, no. 6, pp. 4484–4503, 2013.

[31] E. Polak, *Optimization: Algorithms and consistent approximation*. Princeton University Press, 1997.

[32] B. Brogliato and V. Acary, "Numerical methods for nonsmooth dynamical systems," *Lecture Notes in Applied and Computational Mechanics*, 2008.

[33] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[34] H. Jing, Z. Liu, and H. Chen, "A switched control strategy for antilock braking system with on/off valves," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 4, pp. 1470–1484, 2011.

[35] O. Stursberg and S. Engell, "Optimal control of switched continuous systems using mixed-integer programming," in *15th IFAC World Congress of Automatic Control*, vol. 15, no. 1, 2002, pp. 558–558.

[36] S. Garrido, M. Abderrahim, A. Gimenez, R. Diez, and C. Balaguer, "Anti-swinging input shaping control of an automatic construction crane," *IEEE Transactions on Automation Science and Engineering*, vol. 5, no. 3, pp. 549–557, 2008.

[37] N. Sun, Y. Fang, and H. Chen, "A new antiswing control method for underactuated cranes with unmodeled uncertainties: Theoretical design and hardware experiments," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 453–465, 2015.

[38] P. Cruz, M. Oishi, and R. Fierro, "Lift of a cable-suspended load by a quadrotor: A hybrid system approach," in *American Control Conference*, 2015, pp. 1887–1892.

[39] R. Goebel, R. G. Sanfelice, and A. Teel, "Hybrid dynamical systems," *IEEE Control Systems*, vol. 29, no. 2, pp. 28–93, 2009.

[40] J. Lygeros, "An overview of hybrid systems control," in *Handbook of Networked and Embedded Control Systems*. Springer, 2005, pp. 519–537.

[41] M. Egerstedt, Y. Wardi, and H. Axelsson, "Transition-time optimization for switched-mode dynamical systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 110–115, 2006.

[42] H. Gonzalez, R. Vasudevan, M. Kamgarpour, S. S. Sastry, R. Bajcsy, and C. J. Tomlin, "A descent algorithm for the optimal control of constrained nonlinear switched dynamical systems," in *International Conference on Hybrid Systems: Computation and Control*, 2010, pp. 51–60.

[43] T. M. Caldwell and T. D. Murphey, "Single integration optimization of linear time-varying switched systems," in *American Control Conference*, 2011, pp. 2024–2030.

[44] B. D. Anderson and J. B. Moore, *Optimal control: Linear quadratic methods*. Courier Corporation, 2007.

[45] H. Axelsson, Y. Wardi, M. Egerstedt, and E. Verriest, "Gradient descent approach to optimal mode scheduling in hybrid dynamical systems," *Journal of Optimization Theory and Applications*, vol. 136, no. 2, pp. 167–186, 2008.

[46] J. P. Hespanha, *Linear systems theory*. Princeton University Press, 2009.

[47] P. Bouffard, A. Aswani, and C. Tomlin, "Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 279–284.

[48] X. Liu and X. Kong, "Nonlinear model predictive control for dfig-based wind power generation," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 4, pp. 1046–1055, 2014.
[49] L. Magni, G. De Nicolao, and R. Scattolini, "Output feedback and tracking of nonlinear systems with model predictive control," *Automatica*, vol. 37, no. 10, pp. 1601–1607, 2001.
[50] R. G. Sanfelice, "Input-output-to-state stability tools for hybrid systems and their interconnections," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1360–1366, 2014.
[51] M. A. Müller and F. Allgöwer, "Improving performance in model predictive control: Switching cost functionals under average dwell-time," *Automatica*, vol. 48, no. 2, pp. 402–409, 2012.
[52] J. Hu, J. Shen, and W. Zhang, "A generating function approach to the stability of discrete-time switched linear systems," in *ACM International Conference on Hybrid Systems: Computation and Control*, 2010, pp. 273–282.
[53] F. A. Fontes, "A general framework to design stabilizing nonlinear model predictive controllers," *Systems & Control Letters*, vol. 42, no. 2, pp. 127–143, 2001.
[54] R. Meyer, M. Žefran, and R. A. DeCarlo, "A comparison of the embedding method to multi-parametric programming, mixed-integer programming, gradient-descent, and hybrid minimum principle based methods," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1784–1800, 2014.
[55] K. A. Cunefare, S. De Rosa, N. Sadegh, and G. Larson, "State-switched absorber for semi-active structural control," *Journal of Intelligent Material Systems and Structures*, vol. 11, no. 4, pp. 300–310, 2000.
[56] A. D. Wilson, J. Schultz, and T. D. Murphey, "Trajectory optimization for well-conditioned parameter estimation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, pp. 28–36, 2015.
[57] ——, "Trajectory synthesis for Fisher information maximization," *IEEE Transactions on Robotics*, vol. 30, pp. 1358–1370, 2014.

Anastasia Mavrommati received the joint B.S./M.S. degree in electrical and computer engineering from University of Patras, Greece in 2012, and the M.S. degree in mechanical engineering from Northwestern University, Evanston, IL, USA, in 2015.

She is currently a Ph.D. candidate in mechanical engineering with the Neuroscience and Robotics Laboratory at Northwestern University. Her research interests are focused on real-time algorithms for information acquisition and hybrid control.

Jarvis Schultz received both a B.S. and M.S. in mechanical engineering from the University of Wyoming, and he received his Ph.D. in mechanical engineering from Northwestern University.

He currently works as the Assistant Director of the MS in Robotics program at Northwestern. His research interests are focused on minimal infrastructure embedded control for complex nonlinear systems.

Todd D. Murphey received his B.S. degree in mathematics from the University of Arizona and the Ph.D. degree in Control and Dynamical Systems from the California Institute of Technology.

He is an Associate Professor of Mechanical Engineering at Northwestern University. His laboratory is part of the Neuroscience and Robotics Laboratory, and his research interests include robotics, control, computational methods for biomechanical systems, and computational neuroscience.

Honors include the National Science Foundation CAREER award in 2006, membership in the 2014-2015 DARPA/IDA Defense Science Study Group, and Northwestern's Professorship of Teaching Excellence. He is a Senior Editor of the IEEE Transactions on Robotics.