A Numerical Method for the Optimal Control of Switched Systems

Humberto Gonzalez, Ram Vasudevan, Maryam Kamgarpour, S. Shankar Sastry, Ruzena Bajcsy, Claire Tomlin

***Abstract*— Switched dynamical systems have shown great utility in modeling a variety of systems. Unfortunately, the determination of a numerical solution for the optimal control of such systems has proven difficult, since it demands optimal mode scheduling. Recently, we constructed an optimization algorithm to calculate a numerical solution to the problem subject to a running and final cost. In this paper, we modify our original approach in three ways to make our algorithm's application more tenable. First, we transform our algorithm to allow it to begin at an infeasible point and still converge to a lower cost feasible point. Second, we incorporate multiple objectives into our cost function, which makes the development of an optimal control in the presence of multiple goals viable. Finally, we extend our approach to penalize the number of hybrid jumps. We also detail the utility of these extensions to our original approach by considering two examples.**

I. INTRODUCTION

A natural extension of classical dynamical systems are switched dynamical systems wherein the state of a system is governed by a finite number of differential equations. The control parameter for such systems has a discrete component, the sequence of modes, and two continuous components, the duration of each mode and the continuous input. Switched systems arise in numerous modeling applications [3], [8]. Stemming from Branicky et al.'s seminal work that established a necessary condition for the optimal trajectory of switched systems in terms of quasi-variational inequalities [2], there has been growing interest in the optimal control of such systems. However, Branicky provided only limited means for the computation of the required control.

Several address just the continuous component of the optimal control of an unconstrained nonlinear switched system while keeping the sequence of modes fixed. Given a fixed mode schedule, Xu et al. develop a bi-level hierarchical optimization algorithm: at the higher level, a conventional optimal control algorithm finds the optimal continuous input assuming fixed mode duration and at the lower level, a conventional optimal control algorithm finds the optimal mode duration while keeping the continuous input fixed [12]. Axelsson et al. consider the special case of unconstrained nonlinear autonomous switched systems (i.e. systems wherein the control input is absent) and develop a similar bi-level hierarchical algorithm: at the higher level, the algorithm updates the mode sequence by employing a single mode insertion technique, and at the lower level, the algorithm assumes a fixed mode sequence and minimizes the cost functional over the switching times [1], [5].

Recently, we generalized Axelsson's approach by constructing an optimal control algorithm for constrained nonlinear switched dynamical systems [6]. We developed a bi-level hierarchical algorithm that divided the problem into two nonlinear constrained optimization problems. At the lower level, our algorithm assumed a fixed modal sequence and determined the optimal mode duration and optimal continuous input. At the higher level, our algorithm employed a single mode insertion technique to construct a new lower cost sequence. The result of our approach was an algorithm that provided a sufficient condition to guarantee the local optimality of the mode duration and continuous input while decreasing the overall cost via mode insertion. Though this was a powerful outcome given the generality of the problem under consideration, it suffered from three shortcomings which made its immediate application difficult. First, if our algorithm was initialized at an infeasible point it was unable to find a feasible lower cost trajectory. Unfortunately, initializing an optimization algorithm with a feasible point is nontrivial. Second, our algorithm did not incorporate multiple objectives into its cost function, which is useful for path planning type tasks. Finally, our algorithm did not penalize the number of hybrid jumps. In this paper, we design a new algorithm to address these three deficiencies and detail the utility of this modified approach on two examples.

This paper is organized as follows: Section II provides the mathematical formulation of the problem under consideration, Section III describes the optimal control algorithm which is the primary result of this paper, Section IV details the proof of convergence of our algorithm, Section V considers a numerical implementation of the optimal control scheme, Section VI presents numerical experiments, and Section VII concludes the paper.

II. PROBLEM FORMULATION

In this section, we present the mathematical formalism and define the problem considered in the remainder of the paper. We are interested in the control of systems whose continuous trajectory is governed by a set of vector fields $f_q : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, where q belongs to $\mathcal{Q} = \{1, 2, \dots, Q\}$. Any trajectory of such a system is encoded by a sequence of discrete modes, a corresponding sequence of times spent in each mode, and the continuous input over time. We also require a mapping between each of our objectives and an element of our sequence of discrete modes. To formalize the

H. Gonzalez, R. Vasudevan, M. Kamgarpour, S. S. Sastry, R. Bajcsy, and C. Tomlin are with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA, 94720, {hgonzale,ramv,maryamka,sastry,bajcsy,tomlin}@eecs.berkeley.edu

This work was supported by the Air Force Office of Scientific Research (AFOSR) under Agreement Number FA9550-06-1-0312 and PRET Grant 18796-S2 and the National Science Foundation (NSF) under grants 0703787 and 0724681.

optimal control problem, we define four spaces: Σ is the **discrete mode sequence space**, $\mathcal{S}$ is the **transition time sequence space**, $\mathcal{U}$ is the **continuous input space**, and $\mathbb{N}^W$ is the **objective mapping space**, where $W \in \mathbb{N}$. One notational remark: given an element r of a sequence space, we write $r(i)$ to refer to the ith element of r where $i \in \mathbb{N}$.

First, we describe the discrete mode sequence space. For notational convenience we define an additional vector field, $f_0(\cdot,\cdot) = 0$, in which the trajectories stop evolving. The discrete mode sequence space is most readily thought of as an infinite dimensional space with elements that contain only a finite number of non-zero vector fields:

$$\Sigma = \bigcup_{N=1}^{\infty} \Sigma_N,$$
$$\Sigma_N = \left\{ \sigma \in \mathcal{Q}_0^{\mathbb{N}} \,\middle|\, \sigma(j) \in \mathcal{Q} \ j \leq N,\ \sigma(j) = 0 \ j > N \right\},$$

where $\mathcal{Q}_0 = \mathcal{Q} \cup \{0\}$. We define $\#\sigma = \max\{j \in \mathbb{N} \mid \sigma(j) \neq 0\}$, i.e. $\#\sigma$ is the number of modes in the sequence.

Second, let an element of the transition time sequence space be a sequence whose elements correspond to the amount of time spent in each discrete mode:

$$\mathcal{S} = \bigcup_{N=1}^{\infty} \mathcal{S}_N,$$
$$\mathcal{S}_N = \left\{ s \in l^1 \,\middle|\, s(j) \geq 0 \ \ \forall j \leq N,\ s(j) = 0 \ \forall j > N \right\},$$

where l^1 denotes the space of absolutely summable sequences. Third, we define the continuous input space, $\mathcal{U}$:

$$\mathcal{U} = \left\{ u \in L^2([0,\infty), \mathbb{R}^m) \mid u(t) \in U,\ \forall t \in [0,\infty) \right\}, \quad (1)$$

where $U \subset \mathbb{R}^m$ is a compact, connected set containing the origin.

Finally, let the objective mapping space, $\mathbb{N}^W$, be the set of W–tuples with elements in the natural numbers, where W is equal to the number of objectives in our problem, which we soon define. Elements in this space define a mapping between each of our objectives and an element of our discrete mode sequence space. The importance of defining the objectives in this fashion becomes clear only after we consider the implementation of our algorithm in Section V. We combine these four spaces together to define our optimization space, $\mathcal{X}$, as follows:

$$\mathcal{X} = \big\{ (\sigma, s, u, w) \in \Sigma \times \mathcal{S} \times \mathcal{U} \times \mathbb{N}^W \mid \\ s(k) = 0 \ \forall k > \#\sigma, \text{and } w(i) \leq \#\sigma \ \forall i \big\}, \quad (2)$$

and we denote $\xi \in \mathcal{X}$ by a 4–tuple $\xi = (\sigma, s, u, w)$. Maintaining the notion of absolute times, which we call the **jump time sequence** $\mu : \mathbb{N} \times \mathcal{X} \to [0,\infty)$ is also useful:

$$\mu(i;\xi) = \begin{cases} 0 & \text{if } i = 0 \\ \sum_{k=1}^{i} s(k) & \text{if } i \neq 0. \end{cases} \quad (3)$$

Let $\mu_f(\xi) = \|s\|_{l^1} = \sum_{k=1}^{\infty} s(k)$. We also define, $\pi : [0,\infty) \times \mathcal{X} \to \mathcal{Q}_0$ to return the mode corresponding to an absolute time t:

$$\pi(t;\xi) = \begin{cases} \sigma(\min\{i \mid \mu(i;\xi) > t\}) & \text{if } t < \mu_f(\xi) \\ \sigma(\#\sigma) & \text{if } t = \mu_f(\xi) \\ 0 & \text{if } t > \mu_f(\xi) \end{cases} \quad (4)$$

We suppress the dependence on ξ in μ, π, and μ_f whenever the choice of ξ is clear in context. For notational convenience, we write $\mu(i)$ for μ_i. We also define a metric on our space $d : \mathcal{X} \times \mathcal{X} \to [0,\infty)$:

$$d(\xi_x, \xi_y) = \mathbb{1}\{\sigma_x \neq \sigma_y\} + \|s_x - s_y\|_{l^1} + \|u_x - u_y\|_2 + \\ + \mathbb{1}\{w_x \neq w_y\}, \quad (5)$$

where $\|\cdot\|_{l^1}$ is the standard l^1–norm and $\|\cdot\|_2$ is the standard L^2–norm.

Given $\xi \in \mathcal{X}$ and $x_0 \in \mathbb{R}^n$, the corresponding trajectory, $x^{(\xi)}$, is the solution to:

$$\dot{x}(t) = f_{\pi(t)}\big(x(t), u(t)\big), \quad \forall t \geq 0, \quad x(0) = x_0, \quad (6)$$

where we have suppressed the dependence on x_0 in $x^{(\xi)}$ for convenience. To define the cost function for the optimization problem, first consider the function $J : \mathcal{X} \to \mathbb{R}$ defined as:

$$J(\xi) = \int_0^{\mu_f} L\big(x^{(\xi)}(t), u(t)\big)dt + \sum_{i=1}^{W} \phi_i\big(x^{(\xi)}(\mu_{w(i)})\big) \quad (7)$$

which we refer to as the **standard cost**. Note that each of the ϕ_i's is associated with an index $w(i)$, hence in practice each element $w(i)$ maps a discrete mode to a function ϕ_i, which we refer to as an objective. Then, we define $\widetilde{J} : \mathcal{X} \to \mathbb{R}$, the **cost function**, as $\widetilde{J}(\xi) = J(\xi) + C \cdot \#\sigma$. The reason to differentiate between the standard cost and the terms penalizing the number of jumps in the cost function becomes clear in Section III.

Given $\xi \in \mathcal{X}$ and a family of N_c functions, $h_j : \mathbb{R}^n \to \mathbb{R}$, $j \in \mathcal{J} = \{1, \ldots, N_c\}$, we also constrain the state by demanding the state to satisfy $h_j\big(x^{(\xi)}(t)\big) \leq 0$ for each $t \in [0, \mu_f]$ and each $j \in \mathcal{J}$. We compactly describe all the constraints by defining a new function ψ:

$$\psi(\xi) = \max_{j \in \mathcal{J}} \max_{t \in [0,\mu_f]} h_j\big(x^{(\xi)}(t)\big). \quad (8)$$

since $h_j\big(x^{(\xi)}(t)\big) \leq 0$ for each t and j if and only if $\psi(\xi) \leq 0$. With these definitions, we can state the multiple objective switched hybrid optimal control problem.

Multiple Objective Hybrid Optimal Control Problem:

$$\begin{aligned} &\min_{\xi \in \mathcal{X}} \ \widetilde{J}(\xi) \\ &\text{s.t.} \quad \psi(\xi) \leq 0 \end{aligned} \quad (9)$$

We make the following assumptions on the dynamics, cost, and constraints:

Assumption 1: *The functions L and f_q are Lipschitz and differentiable in x and u for all $q \in \mathcal{Q}$. In addition, the derivatives of these functions with respect to x and u are also Lipschitz.*

Assumption 2: *The functions ϕ_i and h_j are Lipschitz and differentiable in x for all $i \in \{1, \ldots, W\}$ and $j \in \mathcal{J}$. In* *addition, the derivatives of these functions with respect to x are also Lipschitz.*

Assumption 1, together with the controls being measurable and uniformly bounded functions, is sufficient to ensure the existence, uniqueness, and boundedness of the solution to our differential equation (6). Assumption 2 is a standard assumption on the objectives and constraints and is used to prove the convergence properties of the algorithm defined in the next section. Next, we develop an algorithm to solve our problem.

III. Optimization Algorithm

In this section, we present an optimization algorithm to determine a numerical solution to the Multiple Objective Hybrid Optimal Control Problem. Given $\xi \in \mathcal{X}$, the algorithm works by employing a *variation* that inserts a mode $\hat{\alpha}$ and control $\hat{u}$ at time $\hat{t}$ into ξ for a length of time determined by the argument to the variation. The algorithm stops when this variation does not produce either a reduction in the cost or infeasibility. Our goal is the construction of an optimization algorithm, called a *Phase I/Phase II* algorithm, which can find an optimal point from any initial condition.

We begin by providing a high level description of a bi-level hierarchical algorithm that divides the problem into two nonlinear constrained optimization problems:

Bi-Level Optimization Scheme

Stage 1: Given a fixed discrete mode sequence, employ a *Phase I/Phase II* algorithm to find either a locally optimal transition time sequence and continuous control or a locally optimal infeasible transition time sequence and continuous control.

Stage 2: Given a transition time sequence and continuous control, employ the variation, ρ, to modify the discrete mode sequence to find either a lower cost discrete mode sequence if the initialization point is feasible, or find a less infeasible discrete mode sequence if the initialization point is infeasible. Repeat Stage 1 using the modified discrete mode sequence.

In order to formalize this high level description, we first observe that Stage 1 can be transformed into a classical optimal control problem over the switching instances and continuous control (Section V describes this transformation). Let $\hat{a} : \mathcal{S} \times \mathcal{U} \to \mathcal{S} \times \mathcal{U}$ be a function that solves Stage 1, while satisfying the following assumption:

Assumption 3: *Given $\xi = (\sigma, s, u, w)$ and letting $J_{\sigma,w}(s,u) = J(\sigma,s,u,w)$ and $\psi_{\sigma,w}(s,u) = \psi(\sigma,s,u,w)$, $\hat{a}$ is a descent Phase I/Phase II algorithm, i.e. $J_{\sigma,w}\big(\hat{a}(s,u)\big) \leq J_{\sigma,w}(s,u)$ whenever $\psi(\xi) \leq 0$, and $\psi_{\sigma,w}\big(\hat{a}(s,u)\big) \leq \psi_{\sigma,w}(s,u)$ whenever $\psi(\xi) > 0$.*

Section V describes algorithms that satisfy this assumption.

To formalize Stage 2, we begin by defining the variation, ρ. Given $\xi \in \mathcal{X}$, consider an insertion of a mode, $\hat{\alpha}$ and control, $\hat{u}$, at time $\hat{t}$. This insertion is characterized by

$$\eta = (\hat{\alpha}, \hat{t}, \hat{u}) \in \mathcal{H}_\xi \equiv \mathcal{Q} \times \mathcal{T}_\xi \times U \tag{10}$$

where $\mathcal{T}_\xi = [0, \mu_f]$. Given $\xi \in \mathcal{X}$ and $\eta \in \mathcal{H}_\xi$, we let $\rho : [0,\infty) \to \mathcal{X}$ denoted $\rho(\lambda;\xi,\eta)$ describe this type of insertion (ρ is defined explicitly in Definition 2 and its argument denotes the duration of the insertion). We employ the variation, ρ, to characterize when an optimal point has been reached. Notice that after Stage 1 the only way to locally reduce the cost or infeasibility is by modifying the discrete mode sequence. Given this procedure, a point $\xi = (\sigma, s, u, w) \in \mathcal{X}$ satisfies our **optimality condition** if (s,u) is a locally optimal solution to Stage 1 and if the best modification of the discrete mode sequence via the variation, ρ, does not produce a decrease in the cost, $\widetilde{J}$, whenever the point ξ is feasible, or does not produce a decrease in the constraint, ψ, whenever the point ξ is infeasible.

Given any function $F : \mathcal{X} \to \mathbb{R}^k$ for some $k \in \mathbb{N}$, let us define the directional derivative of F composed with ρ for $\lambda > 0$ by:

$$D^{(\xi,\eta)}F = \lim_{\lambda \downarrow 0} \frac{1}{\lambda}\left[F\big(\rho(\lambda;\xi,\eta)\big) - F(\xi)\right] \tag{11}$$

In order to check if we have arrived at a point that satisfies the optimality condition, we first study the effect of the variation, ρ, on the standard cost, J, using the first order approximation of its change due to the insertion, $D^{(\xi,\eta)}J$. If this derivative is negative, then intuitively one can argue that it is possible to decrease the standard cost via the variation. Note that $D^{(\xi,\eta)}\widetilde{J}$ is not well defined because the limit does not exist, hence we only consider the variation with respect to the standard cost, and we account for the variation of the part of the cost that penalizes the number of discrete modes as an additional condition in our algorithm. Second, consider the first order approximation of our constraint, ψ, with respect to ρ, denoted by $D^{(\xi,\eta)}\psi$, to determine if the infeasibility decreases due to the insertion. Again intuitively, if this derivative is negative then it is possible to decrease the infeasibility via the variation.

Using these results, we define an **optimality function**, $\theta : \mathcal{X} \to (-\infty, 0]$, as follows:

$$\theta(\xi) = \min_{\eta \in \mathcal{H}_\xi} \zeta(\xi,\eta) \tag{12}$$

where

$$\zeta(\xi,\eta) = \begin{cases} \max\left\{D^{(\xi,\eta)}J,\, D^{(\xi,\eta)}\psi + \gamma_1\psi(\xi)\right\} & \text{if } \psi(\xi) \leq 0, \\ \max\left\{D^{(\xi,\eta)}J - \gamma_2\psi(\xi),\, D^{(\xi,\eta)}\psi\right\} & \text{o.w.,} \end{cases} \tag{13}$$

and $\gamma_1, \gamma_2 > 0$ are design parameters. Note that $\theta(\xi) \leq 0$ for each $\xi \in \mathcal{X}$, since given a value ξ we can always perform an insertion that leaves the trajectory unmodified, e.g. given $\xi \in \mathcal{X}$ and $\hat{t} \in \mathcal{T}_\xi$ choose $\eta = (\pi(\hat{t}), \hat{t}, u(\hat{t}))$, hence in that case $D^{(\xi,\eta)}J = D^{(\xi,\eta)}\psi = 0$.

To appreciate the utility of our optimality function consider three cases. First, if at a feasible point, $\psi(\xi) \leq 0$, and if $\theta(\xi) < 0$, then a mode insertion which reduces the standard cost while remaining feasible is possible. Second, if at an infeasible point, $\psi(\xi) > 0$, and if $\theta(\xi) < 0$, then a mode insertion which reduces the infeasibility without resulting in too large an increase in the standard cost is possible. Third, if we are at a feasible point and the standard cost cannot be decreased using the variation ρ, or if we are at an infeasible point and the infeasibility cannot be decreased using the variation ρ, then $\theta(\xi) = 0$. Therefore, the optimality function can serve as a stopping criterion for the Bi-Level Optimization Scheme since its zeros encode points that satisfy our optimality condition. Note that the γ_1, γ_2 terms in the optimality function capture the possibility that the reduction in cost or constraint may result in too large an increase in the infeasibility or cost, and therefore maybe undesirable. These additional terms have been shown in practice to result in better quality optima (Section 2.5 of [11] describes these observations).

Given $\alpha, \beta \in (0, 1)$, we also define the maximum insertion length $\lambda^{(\xi,\eta)}$ as follows:

$$\lambda^{(\xi,\eta)} = \max_{k \in \mathbb{N}} \big\{ \beta^k \mid \psi\big(\rho(\beta^k; \xi, \eta)\big) \leq 0, \\ J\big(\rho(\beta^k; \xi, \eta)\big) - J(\xi) \leq \alpha \beta^k \zeta(\xi, \eta) \big\} \tag{14}$$

whenever $\psi(\xi) \leq 0$, and

$$\lambda^{(\xi,\eta)} = \max_{k \in \mathbb{N}} \big\{ \beta^k \mid \psi\big(\rho(\beta^k; \xi, \eta)\big) - \psi(\xi) \leq \alpha \beta^k \zeta(\xi, \eta) \big\} \tag{15}$$

otherwise.

Algorithm 1 describes our numerical method to solve the Multiple Objective Hybrid Optimal Control Problem.

Algorithm 1 Optimization Algorithm for the Multiple Objective Hybrid Optimal Control Problem

Data: $\xi_0 = (\sigma_0, s_0, u_0, w_0) \in \mathcal{X}$, $\alpha, \beta \in (0, 1)$, $\gamma_1, \gamma_2 > 0$.
Step 0. Let $(s_1, u_1) = \hat{a}(s_0, u_0)$ and $\xi_1 = (\sigma_0, s_1, u_1, w_0)$.
Step 1. Set $j = 1$.
Step 2. If $\theta(\xi_j) = 0$ then stop and return ξ_j.
Step 3. $\xi_{j+1} = a(\xi_j)$, where a is defined as follows:
 a. Let $\hat{\eta} = (\hat{\alpha}, \hat{t}, \hat{u}) \in \mathcal{H}_{\xi_j}$ be any point s.t. $\zeta(\xi_j, \hat{\eta}) < 0$, and let $(\tilde{\sigma}_j, \tilde{s}_j, \tilde{u}_j, \tilde{w}_j) = \rho(\lambda^{(\xi,\hat{\eta})}; \xi, \hat{\eta})$.
 b. Given $\tilde{\sigma}_j$, let $(s_{j+1}, u_{j+1}) = \hat{a}(\tilde{s}_j, \tilde{u}_j)$.
 c. Let $\xi_{j+1} = a(\xi_j) = (\tilde{\sigma}_j, s_{j+1}, u_{j+1}, \tilde{w}_j)$.
Step 4. If $\widetilde{J}(\xi_{j+1}) > \widetilde{J}(\xi_j)$ and $\psi(\xi_j) \leq 0$ then stop and return ξ_j.
Step 5. Replace j by $j + 1$ and go to Step 2.

Comparing steps of Algorithm 1 with our Bi-Level Optimization Scheme notice that Step (3b) encodes Stage 1 and Step (3a) encodes Stage 2. Therefore, the function, $a : \mathcal{X} \to \mathcal{X}$, encodes the Bi-Level Optimization Scheme. In order to understand the stopping rule in Step 4, note that $\#\sigma_{j+1}$ is always greater than $\#\sigma_j$, since the variation, ρ, always inserts a new mode into the sequence. Hence, assigning a cost per mode, $C > 0$, we only accept a new point, ξ_{j+1}, if the decrease in cost J with respect to ξ_j is larger than the increase in cost due to the addition of a new mode. The main result of this paper is that Algorithm 1 converges to a point that satisfies the optimality condition.

IV. Algorithm Analysis

In this section, we describe in detail the pieces that are required to prove that Algorithm 1 converges to a point that satisfies the optimality condition. Before we can analyze the convergence properties of the algorithm, we define the sufficient descent property.

Definition 1 (Sufficient Descent)**:** *A function $a : \mathcal{X} \to \mathcal{X}$ is said to have the sufficient descent property with respect to an optimality function, $\theta : \mathcal{X} \to (-\infty, 0]$ for a cost function J and feasible set $\mathcal{F}$, if for all $\xi \in \mathcal{X}$ with $\theta(\xi) < 0$, there exists a $\delta_\xi > 0$ and a neighborhood of ξ, $U_\xi \subset \mathcal{X}$, such that the following inequality is satisfied:*

$$J\big(a(\xi')\big) - J(\xi') \leq -\delta_\xi, \quad \forall \xi' \in U_\xi \cap \mathcal{F}. \tag{16}$$

Theorem 1 in [6] proves that if the standard cost and constraint functions are continuous and if a function, a, has the sufficient descent property with respect to an optimality function, then either the sequence constructed by the function, a, is finite and its last element belongs to the set of zeros of the optimality function or it is infinite and every accumulation point of the constructed sequence belongs to the set of zeros of the optimality function. Therefore, if we prove that the standard cost, J, and constraint, ψ, are continuous and that the function a, as defined in Step 3 of Algorithm 1, has the sufficient descent property with respect to the optimality function, θ, then Algorithm 1 converges to a point that satisfies the optimality condition.

This section is divided into a part where we prove the continuity of the standard cost and constraint and a part which proves the convergence of Algorithm 1. Several of the proofs are extensions of the results found in a technical report [7].

A. Continuity of the Cost and Constraints

First, we check that the standard cost, Equation (7), and the constraint, Equation (8), are continuous under Assumptions 1 and 2.

Proposition 1: *The standard cost, J, as defined in Equation* (7) *is continuous.*

Proof. The result follows immediately by extending Proposition 3 in [7] to the case with multiple objectives ϕ_i. □

Proposition 2: *The constraint, ψ, as defined in Equation* (8) *is continuous.*

Proof. Using Lemma 5.6.7 together with Theorem 4.1.5 from [11] the result follows immediately. □

B. Optimality Function

In this section, we prove the convergence of Algorithm 1. Our algorithm works by inserting a new mode, $\hat{\alpha}$, for a duration of time $\lambda \geq 0$ centered at a time, $\hat{t}$, with input $\hat{u}$. We begin by defining this type of insertion.

Definition 2: *Given $\xi = (\sigma, s, u, w) \in \mathcal{X}$ and $\eta = (\hat{\alpha}, \hat{t}, \hat{u}) \in \mathcal{H}_\xi$, we define the map $\lambda \in [0, \infty) \mapsto \rho(\lambda; \xi, \eta) \in \mathcal{X}$ as the perturbation of ξ after the insertion of mode $\hat{\alpha}$, at* *time $\hat{t}$ using $\hat{u}$ as the control, for a time interval of length λ. Let*

$$\bar{\lambda} = \min\left\{\frac{1}{2}\left|\mu(i) - \hat{t}\right| \mid i \in \mathbb{N},\ |\mu(i) - \hat{t}| > 0\right\}, \quad (17)$$

then, abusing notation, we write $\rho(\lambda;\xi,\eta) = \big(\rho_\sigma(\lambda), \rho_s(\lambda), \rho_u(\lambda), \rho_w(\lambda)\big)$, where $\rho_\sigma(\lambda)$, $\rho_s(\lambda)$, and $\rho_u(\lambda)$ are defined as in Equations (A.15)-(A.17)*, respectively, in [7] and*

$$\rho_w(\lambda) = \begin{cases} \big(w(1) + \mathbb{1}\{\hat{t} \le \mu_{w(1)}\}, \ldots, \\ \quad \ldots, w(W) + \mathbb{1}\{\hat{t} \le \mu_{w(W)}\}\big) & \text{if } \hat{t} = \mu(i) \\ \big(w(1) + 2\cdot\mathbb{1}\{\hat{t} \le \mu_{w(1)}\}, \ldots, \\ \quad \ldots, w(W) + 2\cdot\mathbb{1}\{\hat{t} \le \mu_{w(W)}\}\big) & \text{if } \hat{t} \ne \mu(i) \end{cases} \quad (18)$$

whenever $\lambda \in [0, \bar{\lambda}]$, and $\rho(\lambda;\xi,\eta) = \rho(\bar{\lambda};\xi,\eta)$ whenever $\lambda > \bar{\lambda}$.

Proposition 3: *The function ρ is continuous in all its arguments.*

Proof. The result follows easily from the definition of ρ and observing that ρ is sequentially continuous in all its arguments. □

We need this property in order to understand the variation of the standard cost with respect to this insertion. We begin by studying the variation of the trajectory, $x^{(\rho(\lambda;\xi,\eta))}$, as λ changes. Abusing notation, we let $x^{(\lambda)} \equiv x^{(\rho(\lambda;\xi,\eta))}$, and note that $x^{(\xi)} \equiv x^{(0)}$. Also note that given $t \ge 0$, the map $\xi \mapsto x^{(\xi)}(t)$ is well defined, hence we can compute $D^{(\xi,\eta)}x(t)$ using the definition in Equation (11), which is a first order approximation of the trajectory with respect to ρ at $\lambda = 0$.

To reduce the number of cases we need to consider in future propositions, given $\xi \in \mathcal{X}$ and $\eta \in \mathcal{H}_\xi$, we define:

$$\Delta f(\xi,\eta) = \begin{cases} f_{\hat{\alpha}}\big(x(\hat{t}), \hat{u}\big) - f_{\pi(\hat{t}+\bar{\lambda})}\big(x(\hat{t}), u(\hat{t})\big) & \hat{t} = 0 \\ f_{\hat{\alpha}}\big(x(\hat{t}), \hat{u}\big) - f_{\pi(\hat{t}-\bar{\lambda})}\big(x(\hat{t}), u(\hat{t})\big) & \hat{t} = \mu_f \\ f_{\hat{\alpha}}\big(x(\hat{t}), \hat{u}\big) + \\ \quad -\frac{1}{2} f_{\pi(\hat{t}+\bar{\lambda})}\big(x(\hat{t}), u(\hat{t})\big) + \\ \quad -\frac{1}{2} f_{\pi(\hat{t}-\bar{\lambda})}\big(x(\hat{t}), u(\hat{t})\big) & \text{o.w.} \end{cases} \quad (19)$$

where $\bar{\lambda}$ is as in Equation (17). We now consider the change of the state trajectory with respect to our insertion.

Proposition 4: *The directional derivative of $x^{(\lambda)}$ for λ positive, evaluated at zero, is:*

$$D^{(\xi,\eta)}x(t) = \begin{cases} \Phi(t,\hat{t})\Delta f(x^{(\xi)}, u, \eta) & \text{if } t \in [\hat{t}, \mu_f], \\ 0 & \text{otherwise}, \end{cases} \quad (20)$$

where $\Phi : [0,\infty) \times [0,\infty) \to \mathbb{R}^{n\times n}$ is the solution of the matrix differential equation:

$$\frac{dX(t,\hat{t})}{dt} = \frac{\partial f_{\pi(t)}}{\partial x}\big(x^{(\xi)}(t), u(t)\big) X(t,\hat{t}),\ X(\hat{t},\hat{t}) = I. \quad (21)$$

Proof. This result is identical to Proposition 6 in [7]. □

Given this variation of the state trajectory, we consider variations of the standard cost and constraint functions, in order to define our optimality function.

Proposition 5: *Let J be the standard cost function as defined in Equation* (7)*. The directional derivative of $J(\rho(\lambda;\xi,\eta))$ evaluated at $\lambda = 0$ is*

$$D^{(\xi,\eta)}J = \sum_{w=1}^{W}\left[\frac{\partial \phi_i}{\partial x}\big(x^{(\xi)}(\mu_{w(i)})\big)\ D^{(\xi,\eta)}x(\mu_{w(i)})\right] + \\ + \big(p^{(\xi)}(\hat{t})\big)^T \Delta f(\xi,\eta) + \frac{\partial L}{\partial u}\big(x^{(\xi)}(\hat{t}), u(\hat{t})\big)\big[\hat{u} - u(\hat{t})\big], \quad (22)$$

where $p^{(\xi)}$ is the solution of

$$-\dot{p}(t) = \frac{\partial f_{\pi(t)}}{\partial x}^T\big(x^{(\xi)}(t), u(t)\big)p(t) + \frac{\partial L}{\partial x}^T\big(x^{(\xi)}(t), u(t)\big), \\ p(\mu_f) = 0. \quad (23)$$

Proof. For the derivative of the final and running cost with respect to the variation, the result follows from Proposition 7 from [7]. The other terms are the result of applying the chain rule: $D^{(\xi,\eta)}\phi_i\big(x(t)\big) = \frac{\partial \phi_i}{\partial x}^T\big(x^{(\xi)}(t)\big) D^{(\xi,\eta)}x(t)$. □

Proposition 6: *Let ψ be the constraint function defined in* (8)*. The directional derivative of $\psi\big(\rho(\lambda;\xi,\eta)\big)$ evaluated at $\lambda = 0$ is*

$$D^{(\xi,\eta)}\psi = \max_{(j,t)\in\mathcal{A}_\xi} \frac{\partial h_j}{\partial x}\big(x^{(\xi)}(t)\big)\ D^{(\xi,\eta)}x(t) \quad (24)$$

where $\mathcal{A}_\xi = \big\{(j,t) \in \mathcal{J} \times [0,\mu_f] \mid h_j\big(x^{(\xi)}(t)\big) = \psi(\xi)\big\}$.

Proof. This follows from Proposition 8 in [7]. □

We prove one last technical result that is used in our proof of the convergence of our algorithm.

Proposition 7: *Consider the function ζ, defined in Equation* (13)*. If $\zeta(\xi,\eta) < 0$, then $\lambda^{(\xi,\eta)}$, as defined in Equations* (14) *and* (15)*, is strictly positive.*

Proof. First, assume $\psi(\xi) \le 0$. Note that as $k \to \infty$,

$$\frac{1}{\beta^k}\left[J\big(\rho(\beta^k;\xi,\eta)\big) - J(\xi)\right] \to D^{(\xi,\eta)}J. \quad (25)$$

Since $\zeta(\xi,\eta) \ge D^{(\xi,\eta)}J$ and $\zeta(\xi,\eta) < 0$, it follows that $\alpha\zeta(\xi,\eta) > D^{(\xi,\eta)}J$. Hence, for k large enough,

$$\frac{1}{\beta^k}\left[J\big(\rho(\beta^k;\xi,\eta)\big) - J(\xi)\right] < \alpha\zeta(\xi,\eta). \quad (26)$$

Also, if $\psi(\xi) < 0$ then clearly for k large enough $\psi\big(\rho(\beta^k;\xi,\eta)\big) \le 0$, and if $\psi(\xi) = 0$ note that the definition of ζ implies that $D^{(\xi,\eta)}\psi < 0$, hence, using the same argument, for k large enough

$$\psi\big(\rho(\beta^k;\xi,\eta)\big) = \psi\big(\rho(\beta^k;\xi,\eta)\big) - \psi(\xi) < 0 \quad (27)$$

Therefore there exists $k_0 \in \mathbb{N}$ such that $\lambda^{(\xi,\eta)} = \beta^{k_0}$.

The case when $\psi(\xi) > 0$ follows using the same argument. □

Finally, we can prove that our algorithm has the sufficient descent property.

Theorem 1: *Let $\mathcal{F}$ denote the set of feasible points in $\mathcal{X}$. Algorithm a, as defined in Step 3 of Algorithm 1, has the sufficient descent property with respect to θ for the cost function J and feasible set $\mathcal{F}$, and for each $\xi \in \mathcal{F}$ we have $a(\xi) \in \mathcal{F}$. Moreover, if $\psi(\xi) > 0$ then Algorithm a has the sufficient descent property with respect to θ for the cost function ψ and feasible set $\mathcal{X}$.*

Proof. First note that $a(\xi) \in \mathcal{F}$ whenever $\xi \in \mathcal{F}$. We need to show that for each $\xi \in \mathcal{X} \cap \mathcal{F}$ such that $\theta(\xi) < 0$, there exists $\delta_\xi < 0$ and a neighborhood of ξ, denoted by $U_\xi \subset \mathcal{X}$, such that

$$J\big(a(\xi')\big) - J\big(\xi'\big) \leq \delta_\xi, \quad \forall \xi' \in U_\xi \cap \mathcal{F}. \tag{28}$$

Let $b : \mathcal{X} \to \mathcal{X}$ be defined by $b(\sigma, s, u, w) = (\sigma, s', u', w)$, where $(s', u') = \hat{a}(s, u)$. Since $\hat{a}$ is a descent algorithm by Assumption 3, $J\big(b(\xi)\big) \leq J(\xi)$ for each $\xi \in \mathcal{F}$. Using this definition, note that a can be defined as $a(\xi) = b\big(\rho(\lambda^{(\xi,\hat{\eta})}; \xi, \hat{\eta})\big)$, where $\lambda^{(\xi,\hat{\eta})}$ is defined as in Equations (14) and (15), and $\hat{\eta} \in \mathcal{H}_\xi$ is such that $\zeta(\xi, \hat{\eta}) < 0$, as defined in Step (3a) of Algorithm 1. Let

$$\delta_\xi = \frac{1}{2}\left[J\big(\rho(\lambda^{(\xi,\hat{\eta})}; \xi, \hat{\eta})\big) - J(\xi)\right], \tag{29}$$

then clearly $\delta_\xi < 0$, and let

$$U_\xi = \left\{\xi' \in \mathcal{X} \mid J\big(\rho(\lambda^{(\xi,\hat{\eta})}; \xi', \hat{\eta})\big) - J(\xi') < \delta_\xi\right\}, \tag{30}$$

which is open in $\mathcal{X}$ since both J and ρ are continuous with respect to ξ. Then, given $\xi' \in U_\xi \cap \mathcal{F}$,

$$J\big(a(\xi')\big) - J(\xi') \leq J\big(\rho(\lambda; \xi', \hat{\eta})\big) - J(\xi') < \delta_\xi. \tag{31}$$

If $\psi(\xi) > 0$, then the proof of a having sufficient descent with respect to θ for the cost function ψ and feasible set $\mathcal{X}$ follows similarly. □

We have that our algorithm is a *Phase I/Phase II* algorithm and converges to points that satisfy our optimality condition as desired. Indeed, the algorithm stops only if the variation ρ cannot make a significant decrease in either the cost (for feasible points) or the infeasibility (for infeasible points), or if the decrease in the standard cost J is smaller than the increase in $\widetilde{J}$ due to the insertion of new modes, which is exactly the definition of our optimality condition.

V. Implementation

In this section, we describe the numerical implementation of Algorithm 1. First, we describe how to reformulate Stage 1 in the Bi-Level Optimization Scheme via a transformation into a canonical optimal control problem. Second, we discuss the implementation of our optimality function.

Given a $\xi \in \mathcal{X}$, we discuss how to solve Stage 1 in the Bi-Level Optimization Scheme by transforming our problem into one where the optimization over the switching instances and continuous control becomes an optimization over the initial condition and the continuous control. Algorithms to solve this type of formulation while satisfying Assumption 3 have been extensively studied in the literature [11]. We introduce functions $\gamma_k : [0,1] \to \mathbb{R}$ and $z_k : [0,1] \to \mathbb{R}^n$ for $k = 1, \ldots, \#\sigma$ which are solutions to:

$$\begin{aligned} &\dot{\gamma}_k(t) = s(k) L(z_k(t), \bar{u}_k(t)), \ \gamma_k(0) = 0, \\ &\dot{z}_k(t) = s(k) f_{\sigma(k)}(z_k(t), \bar{u}_k(t)), \ z_k(0) = z_{k-1}(1), \end{aligned} \tag{32}$$

where we let $z_0(1) = x_0$ and $\bar{u}_k(t) = u\big(t \cdot s(k) + \mu(k-1)\big)$ for all t in $[0,1]$ and $k = 1, \ldots, \#\sigma$. Given these definitions, we construct new state variables, $\omega_k : [0,1] \to \mathbb{R}^{n+2}$ and define a new optimal control problem whose solution is a transformed version of the solution to our problem:

$$\min_{\substack{\{s(k)\}_{k=1}^{\#\sigma} \subset \mathbb{R}_+ \\ \{\bar{u}_k\}_{k=1}^{\#\sigma} \subset \mathcal{U}}} \sum_{k=1}^{\#\sigma} \gamma_k(1) + \sum_{i=1}^{W} \phi_i\big(z_{w(i)}(1)\big) \tag{33}$$

subject to:

$$\begin{aligned} &\dot{\omega}_k(t) = \begin{bmatrix} s(k) f_{\sigma(k)}\big(z_k(t), \bar{u}_k(t)\big) \\ 0 \\ s(k) L\big(z_k(t), \bar{u}_k(t)\big) \end{bmatrix}, \ \omega_k(0) = \begin{bmatrix} z_{k-1}(1) \\ s(k) \\ 0 \end{bmatrix}, \\ &h_j\big(z_k(t)\big) \leq 0, \ \forall j \in \mathcal{J}, \ \forall\, t \in [0,1], \ \forall k = 1, \ldots, \#\sigma \end{aligned} \tag{34}$$

The solution to this problem is tractable and equivalent to the solution of Stage 1. In practice, the time is discretized in order to approximate each of the $k = 1, \ldots, \#\sigma$ differential equations. Since we must evaluate the state at each of the objectives in order to solve Stage 1, by associating each of the objectives with an element of σ, as we do with the objective mapping space, we can guarantee that this evaluation is possible. If the objective mapping space was instead not employed and the objectives were allowed to move arbitrarily in time, their evaluation maybe required at a time step where a discrete sample is unavailable.

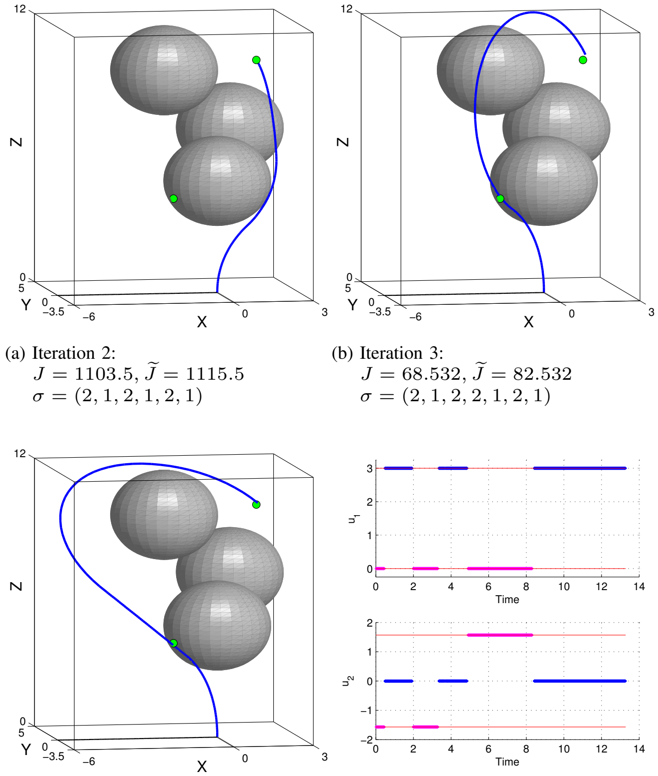
Next, we discuss the implementation of our optimality function. In Algorithm 1, given a $\xi \in \mathcal{X}$, we check to see if $\theta(\xi) = 0$. If $\theta(\xi) < 0$, we find any point $\eta \in \mathcal{H}_\xi$ such that $\zeta(\xi, \eta) < 0$. Since $\theta(\xi)$ is a non-convex min-max optimization problem, it can be implemented either applying a min-max optimization algorithm (similar to those presented in Section 2.5 of [11]) or employing the epigraph transformation to obtain a standard constrained minimization problem. Again, the time is discretized in order to approximate the differential equation; hence, the computation of $\theta(\xi)$ is done by solving an optimization problem over $\hat{u} \in U$ for each $\hat{\alpha} \in \mathcal{Q}$ and each $\hat{t}$ in the set of discretized times.

VI. Examples

In this section, we apply Algorithm 1 to calculate an optimal control for two examples.

A. Bevel-Tip Flexible Needle

Bevel-tip flexible needles are asymmetric needles that move along curved trajectories when a forward pushing force is applied. The 3D dynamics of such needles has been described in [10] and the path planning in the presence of obstacles has been heuristically considered in [4]. Letting the origin be the point of entry of the needle, x, y, z be the



(a) Iteration 2:
$J = 1103.5$, $\widetilde{J} = 1115.5$
$\sigma = (2, 1, 2, 1, 2, 1)$

(b) Iteration 3:
$J = 68.532$, $\widetilde{J} = 82.532$
$\sigma = (2, 1, 2, 2, 1, 2, 1)$

(c) Iteration 4:
$J = 15.818$, $\widetilde{J} = 31.818$
$\sigma = (2, 1, 2, 2, 1, 1, 2, 1)$

(d) Input Iteration 4

Fig. 1: Optimal trajectories (drawn in blue) on the top row and bottom left in an environment with three obstacles (drawn in gray) and objectives (drawn in green) and inputs on the bottom right for the fourth iteration with forward mode (called 1, drawn in blue), turn mode (called 2, drawn in pink), and restrictions (drawn in red).

position of the needle relative to this coordinate system, β_y be the yaw of the needle in the plane, β_p be the pitch of the needle out of the plane, and β_r be the roll of the needle in the plane, the equations of motion are given as:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \\ \dot{\beta}_y(t) \\ \dot{\beta}_p(t) \\ \dot{\beta}_r(t) \end{bmatrix} = \begin{bmatrix} \sin\left(\beta_p(t)\right) & 0 \\ -\cos\left(\beta_p(t)\right)\sin\left(\beta_y(t)\right) & 0 \\ \cos\left(\beta_y(t)\right)\cos\left(\beta_p(t)\right) & 0 \\ \kappa\cos\left(\beta_r(t)\right)\sec\left(\beta_p(t)\right) & 0 \\ \kappa\sin\left(\beta_r(t)\right) & 0 \\ -\kappa\cos\left(\beta_r(t)\right)\tan\left(\beta_p(t)\right) & 1 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \tag{35}$$

In the above, $u_1 \in [0, 3]$ is the insertion speed, $u_2 \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ is the rotation speed of the needle, and κ is the curvature the needle follows and is equal to .22 where all the units are specified according to the centimeter-gram-second system. As suggested by [4], we hybridize the dynamics by introducing two modes: *Forward* and *Turn*. For the *Turn* mode we set $u_1 = 0$ and for the *Forward* mode we set $u_2 = 0$. We consider the optimal control of the needle under the scenario presented in Figure 1 while avoiding three spherical obstacles (drawn in gray) located at $(0\,,\,0\,,\,5)$, $(1\,,\,3\,,\,7)$, and $(-2\,,\,0\,,\,10)$ all with radius 2 under a cost function specified by:

$$\begin{aligned} L\left(x^{(\xi)}(t), u(t)\right) &= u(t)^T \begin{bmatrix} 0.05 & 0 \\ 0 & 0.005 \end{bmatrix} u(t) \\ \phi_1\left(x^{(\xi)}(\mu_{w(1)})\right) &= 50 \cdot \left\| \begin{bmatrix} x(\mu_{w(1)}) \\ y(\mu_{w(1)}) \\ z(\mu_{w(1)}) \end{bmatrix} - \begin{bmatrix} -2 \\ -2 \\ 4.5 \end{bmatrix} \right\|^2 \\ \phi\left(x^{(\xi)}(\mu_f), \mu_f\right) &= 15 \cdot \left\| \begin{bmatrix} x(\mu_f) \\ y(\mu_f) \\ z(\mu_f) \end{bmatrix} - \begin{bmatrix} 2 \\ 3 \\ 10 \end{bmatrix} \right\|^2 + \mu_f. \end{aligned} \tag{36}$$

The cost per number of modes was chosen as $C = 2$, and the parameters inside the optimality function were chosen as $\gamma_1 = 1$ and $\gamma_2 = 10$.

In the example a constraint was added for the final position of the needle to be at most at 0.3 centimeters from the final waypoint. The algorithm was initialized with the mode sequence $\sigma = (2, 1, 2, 1)$, where mode 1 is *Forward* and mode 2 is *Turn*. The first iteration resulted in an infeasible trajectory. The result of the next three iterations, after Step 3 in Algorithm 1, are shown in Figure 1. A fifth iteration was computed ($J = 15.436$, $\widetilde{J} = 35.436$), but it passed the condition in Step 4 of Algorithm 1, as the decrease in J was negligible. On an AMD Opteron, 8 core, 2.2 GHz, 16 GB RAM machine with a MATLAB implementation the total time to compute Stage 1 for all iterations was 148.37 seconds and the total time to compute Stage 2 for all iterations was 156.49 seconds.

B. *Quadrotor Helicopter Control*

Next, we consider the optimal control of a quadrotor helicopter using a two dimensional simplified model. Letting x denote the position along the horizontal axes, z the height above the ground, and θ the roll angle of the helicopter, the equations of motion is given as:

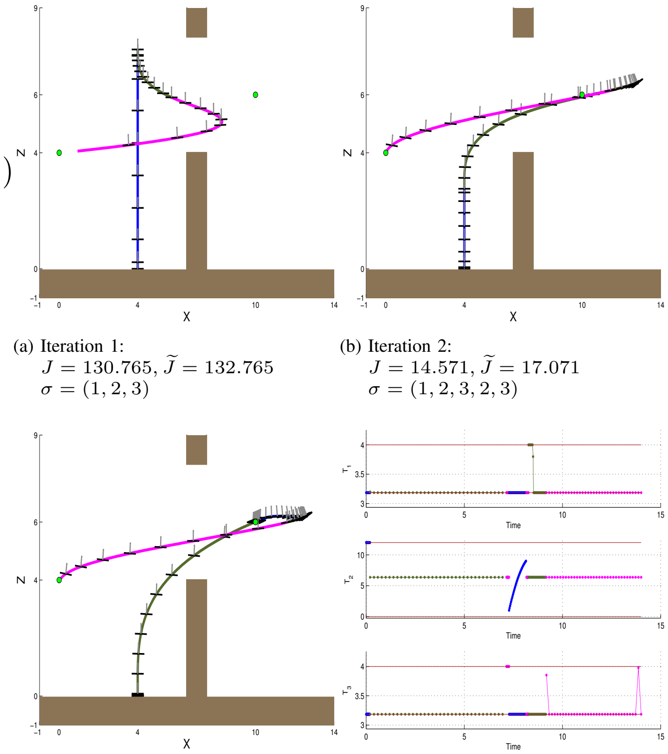
$$\begin{bmatrix} \ddot{x}(t) \\ \ddot{z}(t) \\ \ddot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \frac{\sin\theta(t)}{M} & \frac{\sin\theta(t)}{M} & \frac{\sin\theta(t)}{M} \\ \frac{\cos\theta(t)}{M} & \frac{\cos\theta(t)}{M} & \frac{\cos\theta(t)}{M} \\ \frac{L}{I_y} & 0 & -\frac{L}{I_y} \end{bmatrix} \begin{bmatrix} T_1(t) \\ T_2(t) \\ T_3(t) \end{bmatrix} - \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} \tag{37}$$

where in the above T_1 and T_3 are the thrusts applied at the opposite ends of the quadrotor along the x axis, and T_2 is the sum of the thrusts of the other two rotors at the center of mass of the quadrotor. The parameters M, L, I_y, and g denote the mass, distance from center of mass of each of the rotors T_1 and T_3, moment of inertia about y axis, and gravitational acceleration, respectively. The values of the parameters for this example are taken from the STARMAC experimental platform [9]. We hybridize the dynamics by introducing three modes: *Left*, *Right*, and *Up*. For the *Left* mode we set $T_1 = \frac{1}{4}Mg$, $T_2 = \frac{1}{2}Mg$, and let $T_3 \in [\frac{1}{4}Mg, 4]$. In the *Right* mode, we set $T_3 = \frac{1}{4}Mg$, $T_2 = \frac{1}{2}Mg$, and let $T_1 \in [\frac{1}{4}Mg, 4]$, and in the *Up* mode we set $T_1 = T_3 = \frac{1}{4}Mg$, and let $T_2 \in [0, 12]$. The objective is to reach two waypoints (drawn in green in Figure 2) while avoiding obstacles (drawn in brown in Figure 2), and maintaining a speed between zero and three. We define the cost as

$$L\big(x^{(\xi)}(t),u(t)\big)=\tilde{u}(t)^T\begin{bmatrix}0.03 & 0 & 0\\ 0 & 0.03 & 0\\ 0 & 0 & 0.03\end{bmatrix}\tilde{u}(t)$$
$$\phi_1\big(x^{(\xi)}(\mu_{w(1)})\big)=5\cdot\left\|\begin{bmatrix}x(\mu_{w(1)})\\ z(\mu_{w(1)})\end{bmatrix}-\begin{bmatrix}10\\ 6\end{bmatrix}\right\|^2-\cos\big(\theta(\mu_{w(1)})\big)$$
$$\phi\big(x^{(\xi)}(\mu_f),\mu_f\big)=5\cdot\left\|\begin{bmatrix}x(\mu_f)\\ z(\mu_f)\end{bmatrix}-\begin{bmatrix}0\\ 4\end{bmatrix}\right\|^2-\cos\big(\theta(\mu_f)\big)+ \\ +\dot{x}^2(\mu_f)+\dot{z}^2(\mu_f)+\mu_f \tag{38}$$

where $\tilde{u}(t)=u(t)-u_{ss}$, with $u_{ss}=Mg\cdot(\frac{1}{4},\frac{1}{2},\frac{1}{4})$ is the steady-state input. The cost per number of modes was chosen as $C=\frac{1}{2}$, and the parameters inside the optimality function were chosen as $\gamma_1=1$ and $\gamma_2=10$.

The algorithm was initialized with the sequence $\sigma=(1,2,3)$, where mode 1 is *Up*, mode 2 is *Right*, and mode 3 is *Left*. The result of the first three iterations, after Step 3 in Algorithm 1, are shown in Figure 2. A fourth iteration was computed, but passed the condition in Step 4 of Algorithm 1. On an AMD Opteron, 8 core, 2.2 GHz, 16 GB RAM machine with a MATLAB implementation the total time to compute Stage 1 for all iterations was 72.3 seconds and the total time to compute Stage 2 for all iterations was 110.31 seconds.



(a) Iteration 1: $J=130.765$, $\widetilde{J}=132.765$, $\sigma=(1,2,3)$

(b) Iteration 2: $J=14.571$, $\widetilde{J}=17.071$, $\sigma=(1,2,3,2,3)$

(c) Iteration 3: $J=12.708$, $\widetilde{J}=16.208$, $\sigma=(1,2,3,1,3,2,3)$

(d) Input Iteration 3

Fig. 2: Optimal trajectories (the STARMAC is drawn in black and the normal direction to the frame is drawn in gray) on the top row and bottom left in an environment with constraints (drawn in brown) and objectives (drawn in green) and inputs on the bottom right for the third iteration with *Up* mode is drawn in blue (mode 1), the *Right* mode is drawn in green (mode 2), and the *Left* mode is drawn in pink (mode 3), and restrictions (drawn in red).

VII. Conclusion

This paper presents an extension to an algorithm to numerically determine the optimal control for constrained nonlinear switched systems. For such systems, the control parameter has both a discrete component, the sequence of modes, and two continuous components, the duration of each mode and the continuous input. We extend our original solution to this problem in three ways to improve its utility. In practice, the algorithm presented in this paper can be more readily applied to any constrained nonlinear switched system to determine an optimal control when compared to our previous algorithm.

References

[1] 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*, 136(2):167–186, 2008.

[2] M. Branicky, V. Borkar, and S. Mitter. A Unified Framework for Hybrid Control: Model and Optimal Control theory. *IEEE Transactions on Automatic Control*, 43(1):31–45, 1998.

[3] R. Brockett. Stabilization of Motor Networks. In *Proceedings of the 34th IEEE Conference on Decision and Control*, volume 2, 1995.

[4] V. Duindam, R. Alterovitz, S. Sastry, and K. Goldberg. Screw-based motion planning for bevel-tip flexible needles in 3d environments with obstacles. In *IEEE Intl. Conf. on Robot. and Autom*, pages 2483–2488, 2008.

[5] M. Egerstedt, Y. Wardi, and H. Axelsson. Transition-Time Optimization for Switched-Mode Dynamical Systems. *IEEE Transactions on Automatic Control*, 51(1):110–115, 2006.

[6] H. Gonzalez, R. Vasudevan, M. Kamgarpour, S. Sastry, R. Bajcsy, and C. Tomlin. A Descent Algorithm for the Optimal Control of Constrained Nonlinear Switched Dynamical Systems. In *Proceedings of the 13th International Conference on Hybrid Systems: Computation and Control*, 2010.

[7] H. Gonzalez, R. Vasudevan, M. Kamgarpour, S. S. Sastry, R. Bajcsy, and C. Tomlin. A descent algorithm for the optimal control of constrained nonlinear switched dynamical systems: Appendix. Technical Report UCB/EECS-2010-9, EECS Department, University of California, Berkeley, Jan 2010.

[8] S. Hedlund and A. Rantzer. Optimal Control of Hybrid Systems. In *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, Arizona, Dec. 1999.

[9] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin. Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment. In *Proc. of the AIAA Guidance, Navigation, and Control Conference*, 2007.

[10] V. Kallem and N. Cowan. Image-guided Control of Flexible Bevel-Tip Needles. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3015–3020, 2007.

[11] E. Polak. *Optimization: Algorithms and Consistent Approximations*. Springer, 1997.

[12] X. Xu and P. Antsaklis. Results and Perspective on Computational Methods for Optimal Control of Switched Systems. *Lecture Notes in Computer Science*, 2623:540–556, 2003.