Iterative Sequential Action Control for Stable, Model-Based Control of Nonlinear Systems

Emmanouil Tzorakoleftherakis, *Member, IEEE*, and Todd D. Murphey, *Member, IEEE*

***Abstract*—This paper presents iterative Sequential Action Control (iSAC), a receding horizon approach for control of nonlinear systems. The iSAC method has a closed-form open-loop solution, which is iteratively updated between time steps by introducing constant control values applied for short duration. Application of a contractive constraint on the cost is shown to lead to closed-loop asymptotic stability under mild assumptions. The effect of asymptotically decaying disturbances on system trajectories is also examined. To demonstrate the applicability of iSAC, we employ a variety of systems and conditions, including a 13-dimensional quaternion-based quadrotor and NASA's TRACE spacecraft. Each system is tested in different scenarios, ranging from feasible and infeasible trajectory tracking, to setpoint stabilization, with or without the presence of external disturbances. Finally, limitations of this work are discussed.**

I. Introduction

Model-predictive control (MPC) has been widely used over the past twenty years for control of linear and nonlinear systems [1], [2]. The rationale behind MPC is the following: at each time step, an L_2 or alternative variation of the cost function is locally optimized over time to obtain the open-loop control as a function of time, only a small portion of which is actually applied to the system. The time horizon is then shifted, and the process is repeated based on acquired state feedback. Using this approach, it is clear that optimizing with respect to the entire open-loop control, although often effective, might not be the most efficient way to compute the control since most of the optimizer is typically discarded.

In view of this observation, our previous work in [3]–[7] presented Sequential Action Control (SAC), a receding horizon approach for control of nonlinear systems, that exploits elements from hybrid systems theory [8], [9]. In contrast with the aforementioned MPC methods, the open-loop solution in SAC optimizes the needle variation [10] of the cost resulting in a single, constant magnitude action which does not optimize, but rather improves the cost function value relative to applying only a nominal control signal. Since only a single action—obtained in closed-form—is computed at each time step, control calculation is efficient. Our earlier work in [3]–[7] indicates that SAC can drive benchmark and challenging systems—including the cart-pendulum, acrobot and pendubot in [3], hopping and humanoid locomotion in [5] and rotor vehicles with dynamics on Lie groups in [6]—close to a desired equilibrium. Nevertheless, it cannot achieve final stabilization and, as a result, switching to a locally stabilizing controller is necessary [3], [6]. An illustration of this behavior is shown in Fig. 1 for the cart-pendulum inversion system.

This paper presents iterative Sequential Action Control (iSAC), an extension of SAC that addresses one fundamental question that was left open in our previous work—how can SAC achieve consistent stabilizing behavior? We provide new theoretical results that show that our modified method, iSAC, can achieve closed-loop stability, which was not possible using SAC in our previous work. We also present side-by-side the procedural differences between iSAC and SAC and how the modifications allow us to prove closed-loop stability. Simulation results show that, with minimal modifications across examples, iSAC is consistently successful in a variety of control scenarios, ranging from benchmark inversion problems [11] to control of quadrotors performing complex three-dimensional tasks like flips, and constrained maneuvers of spacecrafts. Some of the examined conditions include feasible and infeasible trajectory tracking, setpoint stabilization, and control under external disturbances.

In addition to the aforementioned points, the following novelties of iSAC distinguish this work from alternative MPC methods (see, e.g., [1], [2], [12]–[17] and references therein). In order to solve the open-loop problem, most MPC methods either employ nonlinear programming solvers (see [18] and [19] for a review) or solve a matrix Riccati differential equation as, for example, in [13]–[15]. On the contrary, the solution of the open-loop problem in iSAC has an analytic form which requires only forward simulations of the dynamics and a costate variable, which is computationally efficient and does not depend on black-box optimization routines. Moreover, control saturations can be incorporated without additional computational overhead. Finally, as opposed to many MPC alternatives that utilize discrete-time dynamics [1], [2], [15], [16], [20], iSAC uses continuous-time dynamics. As a result, variable-step integration may be used, which, combined with the previous points, significantly speeds-up the solution process.

This paper is structured as follows: Section II provides a description of our previous work on SAC; Section III describes iSAC and the features that differentiate the proposed method from SAC. In Section IV we discuss the global stability properties of iSAC. Section V demonstrates applicability of iSAC to a variety of systems and control scenarios while conclusive remarks are given in Section VI. Finally, all proofs are presented in the Appendix.

Authors are with the Neuroscience and Robotics Laboratory (N×R) at the Department of Mechanical Engineering, Northwestern University, Evanston, IL. Email: man7therakis@u.northwestern.edu, t-murphey@northwestern.edu.

II. Preliminaries - Sequential Action Control

For convenience, we briefly summarize the work presented in [3]–[6]. SAC is a receding horizon method that enables real-time, closed-loop constrained control synthesis by following the cyclic diagram in Fig. 2. We shall consider nonlinear systems with input constraints such that

$$\dot{x} = f(t,x,u) = g(t,x) + h(t,x)\,u \quad \forall t \tag{1}$$

$$\text{with } \; u \in \mathcal{U} \; \text{ and}$$

$$\mathcal{U} := \Big\{ u \in \mathbb{R}^p : u_{min} \le u \le u_{max},\; u_{min} \le 0 \le u_{max} \Big\},$$

i.e., systems that can be nonlinear with respect to the state vector, $x : \mathbb{R} \to \mathcal{X}$, but are assumed to be linear with respect to the control vector, $u : \mathbb{R} \to \mathcal{U}$. State constraints may be added in the form of penalty terms in the cost. The state will sometimes be denoted as $t \mapsto x\big(t; x(t_i), u(\cdot)\big)$ when we want to make explicit the dependence on the initial state (and time), and corresponding control signal.

The SAC method uses objectives of the form

$$J\big(x(\cdot)\big) = \int_{t_i}^{t_i+T} l\big(t, x(t)\big)\, dt + m\big(t_i + T, x(t_i + T)\big), \tag{2}$$

with incremental cost $l\big(t, x(t)\big)$, terminal cost $m\big(t_i + T, x(t_i + T)\big)$ and time horizon T. Although (2) lacks a norm on control effort, this norm is included in one of the subsequent steps as shown in (6). The following definition is necessary before introducing the open-loop problem solved in SAC.

Definition 1: An action A is defined by the triplet consisting of a constant control value, $u_A \in \mathcal{U}$, application duration, $\lambda_A \in \mathbb{R}^+$ and application time, $\tau_A \in \mathbb{R}$, such that $A := \{u_A, \lambda_A, \tau_A\}$.

The open-loop problem in SAC calculates controls that *improve* (not optimize) the objective (2) relative to applying only a nominal control signal. Specifically, the open-loop problem $\mathcal{P}$ in SAC is defined as follows

$$\mathcal{P}(t_i,\, x_i) : \tag{3}$$

Find action A such that

$$J\big(x(t; x_i, u_i^*(\cdot))\big) < J\big(x(t; x_i, u_i^{nom}(\cdot))\big) \tag{4}$$

subject to

$$u_i^*(t) = \begin{cases} u_A & \tau_A \le t \le \tau_A + \lambda_A \\ u_i^{nom}(t) & \text{else} \end{cases},$$

$$\tau_A >= t_i, \;\; \tau_A + \lambda_A <= t_i + T,$$

and (1) with $t \in [t_i, t_i + T]$ and $x(t_i) = x_i$

where u_i^{nom} is a nominal control signal (see Remark 1). The subscript i denotes the i-th time step, starting from $i = 0$ and will be used for the remainder of the paper in the same way.

The solution $u_i^*(t)$ of problem $\mathcal{P}$ includes a switch to the calculated action A for $t \in [\tau_A, \tau_A + \lambda_A] \subseteq [t_i, t_i + T]$ (Fig. 4a) and thus, it is piecewise continuous in t. When applied to (1), $u_i^*(t)$ generates a (discontinuous) switch of the same duration λ_A in the dynamics from $f\big(t, x, u_i^{nom}(\cdot)\big)$ to $f(t, x, u_A)$. The condition in (4) highlights a key feature of SAC, e.g., rather than optimize the objective (2), SAC actions *improve* (2) relative to only applying nominal control (see Remark 1). As the receding horizon strategy progresses, $\mathcal{P}(t_i, x_i)$ is solved from the current time t_i using the measured state x_i, and the calculated control $u_i^*(t)$—corresponding to $x_i^*(t)$—is applied for t_s seconds (sampling time) to drive the system from x_i to $x\big(t_i + t_s; t_i, x_i, u_i^*(\cdot)\big)$. The process is then repeated at the next sampling instance, i.e., $t_i \leftarrow t_i + t_s$ and $i \leftarrow i + 1$. The final result of the closed-loop receding horizon strategy is a sequence of actions, forming a piecewise continuous control signal $\overline{u}_{cl}(t)$ with state response $\overline{x}_{cl}(t)$.

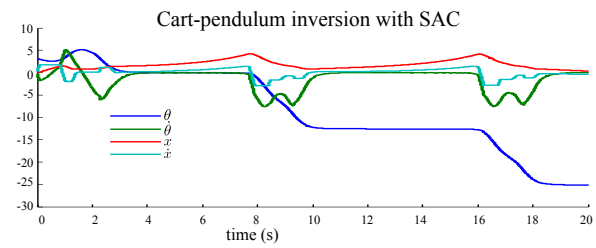

Fig. 1. Cart-pendulum inversion with SAC. While SAC can drive the system close to the upward equilibrium, it does not achieve final stabilization.

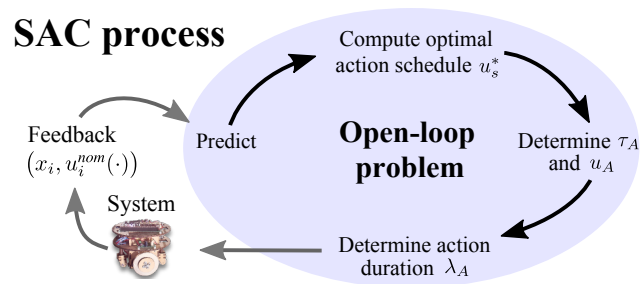

Fig. 2. An overview of the SAC control process. In order to solve (3), the SAC method follows the four steps shown above.

Unlike alternative MPC methods [1], [2], [16], [17], [21], [22], the open-loop problem in SAC can be solved in closed form without employing nonlinear programming solvers (see [18] and [19] for a review). In order to solve (3), the SAC method follows four steps as illustrated in Fig. 2. Beginning with prediction, the steps of the SAC process are described in the following subsections.

Remark 1: Nominal control $u^{nom} : \mathbb{R} \to \mathcal{U}$ is piecewise continuous in t, and is used as a basis when calculating the open-loop solution. It is often $u^{nom}(\cdot) \equiv 0$ so that problem $\mathcal{P}$ outputs the optimal action relative to doing nothing (allowing the system to drift). Alternatively, $u^{nom}(\cdot)$ may be an optimized feedforward or state-feedback controller, e.g., regulating an unstable equilibrium. For purposes of evaluation, the nominal control throughout this paper is considered constant or zero. The system trajectory corresponding to application of nominal control will be denoted as $x\big(t; x(t_i), u^{nom}(\cdot)\big)$ or $x^{nom}(\cdot)$ for brevity.

Steps for solving the open-loop problem $\mathcal{P}$

The solution process assumes the following:

Assumption 1: The control objective is to steer the state to the origin. This is not a restrictive assumption as most control

scenarios (including trajectory tracking) can satisfy it with a change of coordinates.

Assumption 2: The dynamics f in (1) are continuous in u, piecewise continuous in t and continuously differentiable in x. Also, f is compact, and thus bounded, on compact sets $\mathcal{X}$ and $\mathcal{U}$. Finally, the system is assumed to be controllable and $f(\cdot, 0, 0) = 0$.

Assumption 3: The terminal cost m is positive definite and continuously differentiable. The incremental cost $l(t, x)$ is continuous in t and continuously differentiable in x and $l(\cdot, 0) = 0$. Also, there exists a continuous positive definite, radially unbounded function $M : \mathcal{X} \to \mathbb{R}^+$ such that $l(t, x) \geq M(x)\ \forall t$.

Assumption 4: The trajectory $x_i^*(t) \in \mathcal{X}$ corresponding to the solution $u_i^*(t) \in \mathcal{U}$ of $\mathcal{P}(t_i, x_i)$ is absolutely continuous, and thus bounded, in $[t_i, t_i + T]$.

The open-loop problem $\mathcal{P}$ can then be solved by following the four steps presented below.

1) Predict: The SAC process begins by predicting the evolution of a system model from current state feedback. In this step, the system (1) is simulated from the current state x_i and time t_i, under $u_i^{nom}(t)$ for $t \in [t_i,\ t_i + T]$. The sensitivity of (2) to the state variations along the predicted trajectory $x_i^{nom}(\cdot)$ is provided by an adjoint variable, $\rho_i : [t_i,\ t_i + T] \to \mathbb{R}^n$, also simulated during the prediction step. The adjoint satisfies

$$\dot{\rho}_i = -D_2 l\big(t, x_i^{nom}\big)^T - D_2 f\big(t, x_i^{nom}, u_i^{nom}\big)^T \rho_i \qquad (5)$$
$$\text{subject to } \ \rho_i(t_i + T) = D_2 m\big(t_i + T, x_i^{nom}(t_i + T)\big)^T,$$

where D_i denotes derivative with respect to i-th argument.

The prediction phase completes upon simulation of the state using (1) and the adjoint system (5) under $u_i^{nom}(\cdot)$ control. The resulting trajectories $x_i^{nom}(\cdot)$, $\rho_i(\cdot)$ will be used in the remaining three steps of the solution process.

2) Compute optimal action schedule $u_s^*(\cdot)$: In this step, we compute a schedule $u_s^* : [t_i, t_i + T] \to \mathbb{R}^p$ which contains candidate action values and their corresponding application times, assuming $\lambda \to 0^+$ for all (see Fig. 3 for a sample one-dimensional action schedule). The final u_A and τ_A will be selected from these candidates in step three of the solution process such that $u_A = u_s^*(\tau_A)$, while a finite duration λ_A will be selected in step four. The optimal action schedule $u_s^*(\cdot)$ is calculated by minimizing

$$J_{u_s} = \frac{1}{2} \int_{t_i}^{t_i+T} \left[\frac{dJ}{d\lambda}(t) - \alpha_d\right]^2 + \|u_s(t) - u_i^{nom}(t)\|_R^2 \, dt, \qquad (6)$$

$$\frac{dJ}{d\lambda}(t) = \rho_i(t)^T \left[f\big(t, x_i^{nom}(t), u_s(t)\big) - f\big(t, x_i^{nom}(t), u_i^{nom}(t)\big)\right]$$

where the quantity $\frac{dJ}{d\lambda}(\cdot)$, called mode insertion gradient (see [8], [9]), denotes the rate of change of the cost with respect to a switch of infinitesimal duration λ in the dynamics of the system. In this case, $\frac{dJ}{d\lambda}(\cdot)$ shows how the cost will change if we introduce infinitesimal switches from $f\big(t, x_i^{nom}(t), u_i^{nom}(t)\big)$ to $f\big(t, x_i^{nom}(t), u_s(t)\big)$ at any time $t \in [t_i, t_i + T]$. Intuitively, minimization of (6) is driving $\frac{dJ}{d\lambda}(\cdot)$ to a negative value $\alpha_d \in \mathbb{R}^-$. As a result, each switch/action value in $u_s^*(t)$ is the single choice that improves (2) (relative to only applying nominal control) if applied for $\lambda \to 0^+$ at its corresponding application time. The design parameter α_d determines how much the cost is improved by each infinitesimal action in the schedule $u_s^*(t)$.

Remark 2: Equations (5) and the mode insertion gradient in (6) can alternatively be explained as the adjoint equation and the variation of the Hamiltonian in Pontryagin's Maximum Principle (see, e.g., [23]). However, note that our method uses the variation of the Hamiltonian away from the optimizer, which can be verified by observing that (5) depends on nominal state and control trajectories which are known a priori at each time step.

Based on the simulation of the dynamics (1), and (5) completed in the prediction step (Section II-1), minimization of (6) leads to the following closed-form expression for the optimal action schedule:

$$u_s^*(t) = u_i^{nom}(t) + (\Lambda + R^T)^{-1} \, h\big(t, x_i^{nom}(t)\big)^T \rho_i(t) \, \alpha_d, \quad (7)$$

where $\Lambda \triangleq h\big(t, x_i^{nom}(t)\big)^T \rho_i(t) \rho_i(t)^T h\big(t, x_i^{nom}(t)\big)$. Derivation of (7) is a straightforward extension of the proof of Theorem 1 in [3]. The only difference is that, here, we added the term $u_i^{nom}(t)$ in the control norm in (6), to get a more compact form for the solution (7).

The infinitesimal action schedule can then be directly saturated to satisfy any min/max control constraints of the form $u_{min,k} \leq 0 \leq u_{max,k}\ \forall k \in \{1, \ldots, m\}$ such that $u_s^* \in \mathcal{U}$ without additional computational overhead (see [3] for proof). This is possible because the SAC method does not optimize but rather improves (2), and as such, even if the schedule is directly saturated, each action in the schedule will still point to a descent direction of the cost.

The following assumption is used in conjunction with Proposition 1 to ensure that (7) leads to negative mode insertion gradient, and thus, decreases the cost (2).

Assumption 5: It is assumed that $h\big(t, x(t)\big)^T \rho(t) \neq 0^{p \times 1}$ (related to controllability—see [24]). Also, we assume that u_i^{nom} is not an optimizer of (2) in the current time step (usually selected as constant or zero—see Remark 1), and that $J\big(x_i^*(t)\big) \neq 0$, i.e., system trajectories have not already converged to the desired equilibrium.

3) Determine application time τ_A (and thus u_A value): The SAC method optimizes a decision variable not normally included in control calculations—the choice of *when* to act. As opposed to always acting at the current time, i.e., $\tau_A = t_i$, the application time of an action is allowed to take values in $\tau_A \in [t_i,\ t_i + T]$.

Recall that the curve $u_s^*(\cdot)$ in the previous step provides the values and application times of possible infinitesimal actions that SAC could take at different times to improve (2) from that time. In this step the SAC method chooses one of these actions to apply, i.e., chooses the application time τ_A and thus an action value u_A such that $u_A = u_s^*(\tau_A)$. To do that, $u_s^*(\cdot)$

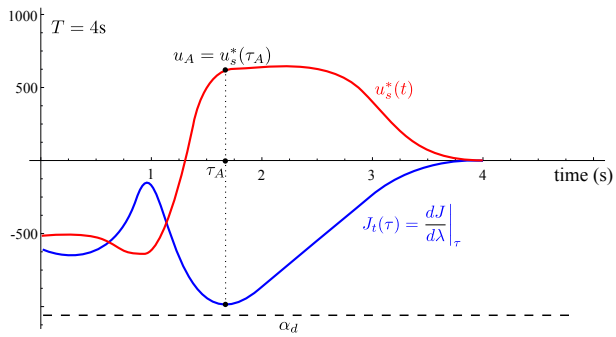

Fig. 3. A sample, one-dimensional ($m = 1$) action schedule $u_s^*(t)$ and the corresponding cost (8) used to calculate the application time τ_A for the hypothetical time window $[0, 4]$s. Every point on $u_s^*(t)$ corresponds to an action of infinitesimal duration, with value and application time as determined by the curve. Because $u_s^*(t)$ is calculated from minimization of (6), all actions in $u_s^*(t)$ would improve (2) relative to only applying nominal control if applied for $\lambda \to 0^+$. Each point on the mode insertion gradient curve approximates the change in cost (2) achievable by infinitesimal application of the corresponding action in $u_s^*(t)$. By choosing the application time that minimizes this curve, we pick the action the generates the greatest cost reduction in the current time window.

is searched for a time τ_A that minimizes

$$J_t(\tau) = \frac{dJ}{d\lambda}\bigg|_\tau, \tag{8}$$

$$\frac{dJ}{d\lambda}\bigg|_\tau = \rho_i(\tau)^T \left[f\big(\tau, x_i^{nom}(\tau), u_s^*(\tau)\big) - f\big(\tau, x_i^{nom}(\tau), u_i^{nom}(\tau)\big)\right]$$

$$\text{subject to } \tau \in [t_i, t_i + T].$$

Notice that the cost (8) is the mode insertion gradient evaluated at the optimal schedule $u_s^*(\cdot)$. Thus, minimization of (8) is equivalent to selecting the infinitesimal action from $u_s^*(\cdot)$ that will generate the greatest cost reduction relative to only applying nominal control. For a sample $J_t(\tau)$ see Fig. 3.

4) Determine control duration λ_A: So far, τ_A and u_A (see Definition 1) have been selected from a schedule of possible *infinitesimal* actions, $u_s^*(\cdot)$. The final step in synthesizing a SAC action is to choose how long to act, i.e., a finite control duration λ_A, such that the condition in (4) is satisfied. The following assumption and proposition will facilitate the analysis in the sequel.

Proposition 1: For a choice of $\alpha_d < 0$ in (6), an infinitesimal control action $u_s^*(\tau)$ that is selected according to Assumption 5 will result in $\frac{dJ}{d\lambda}(\cdot) < 0$.

Proof: See Appendix. ■

Proposition 1 proves that if $\alpha_d < 0$, then the infinitesimal $u_s^*(\tau_A)$ will lead to a negative $\frac{dJ}{d\lambda}(\cdot)$. From [9], [25], there is a non-zero neighborhood around $\lambda \to 0^+$ where the mode insertion gradient models the change in cost in (4) to first order, and thus, a finite duration λ_A exists that satisfies (4). In particular, for *finite* durations λ in this neighborhood we can write

$$J\big(x(t; x_i, u_i^*(\cdot))\big) - J\big(x(t; x_i, u_i^{nom}(\cdot))\big) = \Delta J \approx \frac{dJ}{d\lambda}\bigg|_{\tau_A} \lambda. \tag{9}$$

Thus, from Proposition 1 and (9), the condition in (4) is feasible. Then, a finite action duration λ_A can be calculated by employing a *line search* process [25]. Starting with an initial duration λ_0 we simulate the effect of the control action using (1) and (2). If the simulated action satisfies (4), the duration is selected. If this is not the case, the duration is reduced and the process is repeated. By continuity, the final duration will produce a change in cost within tolerance of that predicted from (9).

After computing the duration λ_A, the control action A is fully specified (it has a value, an application time and a duration) and thus the solution $u_i^*(t)$ of problem $\mathcal{P}$ has been determined. By iterating on this process (Section II-1 until Section II-4), SAC synthesizes piecewise continuous, constrained control laws for nonlinear systems. For more information about SAC, the reader is encouraged to consult [3], [5], [6].

III. From SAC to iSAC

In this section we will describe the differences between SAC and iSAC, and reformulate the open-loop problem that is being solved. We will focus on the two key modifications introduced in iSAC, i.e., a) the iterative update of the open-loop solution $u_i^*(t)$ across time steps, and b) the application of a contractive constraint on the cost. These two features are illustrated in Fig. 4 and are discussed in more detail in the following paragraphs. For convenience, the corresponding SAC behavior is also shown in the same figure.

A. Iterative update of $u_i^*(t)$

Figure 4a illustrates how the iSAC method stores actions calculated at previous time steps and modifies $u_i^*(t)$ by introducing a new action at every time step. On the contrary, SAC only keeps a single action in $u_i^*(t)$, regardless of whether that action lies in the current application window $[t_i, t_i + t_s]$ or not. In light of this, the following definition is necessary to distinguish between the nominal control (zero or constant in this paper) from the open-loop problem $\mathcal{P}$ in SAC, and the default control that iSAC uses to calculate actions.

Definition 2: Default control $u_i^{def} : [t_i, t_i + T] \to \mathcal{U}$, is piecewise continuous in t and is defined as

$$u_i^{def}(t) = \begin{cases} u_{i-1}^*(t) & t_i \le t \le t_i + T - t_s \\ u_i^{nom}(t) & t_i + T - t_s < t \le t_i + T \end{cases}, \tag{10}$$

with $u_0^{def}(\cdot) \equiv u_0^{nom}(\cdot)$. In expression (10), $u_{i-1}^* : [t_{i-1}, t_{i-1} + T] \to \mathcal{U}$ is the output of $\mathcal{P}(t_{i-1}, x_{i-1})$ from the previous time step $i - 1$, and $t_s = t_i - t_{i-1}$ is the sampling period.

As a result, actions calculated in iSAC improve the cost (2) with respect to applying default control $u_i^{def}(\cdot)$. Specifically, as shown in Fig. 4d, iSAC actions establish a decreasing behavior[1] for (2) from time step to time step. As will be explained in the following paragraph, this is accomplished by applying a contractive constraint on the cost, while the iterative nature of $u_i^*(t)$ has a major role in the formulation of this constraint.

[1]In Section IV we make use of this fact and consider (2) as a candidate Lyapunov function.

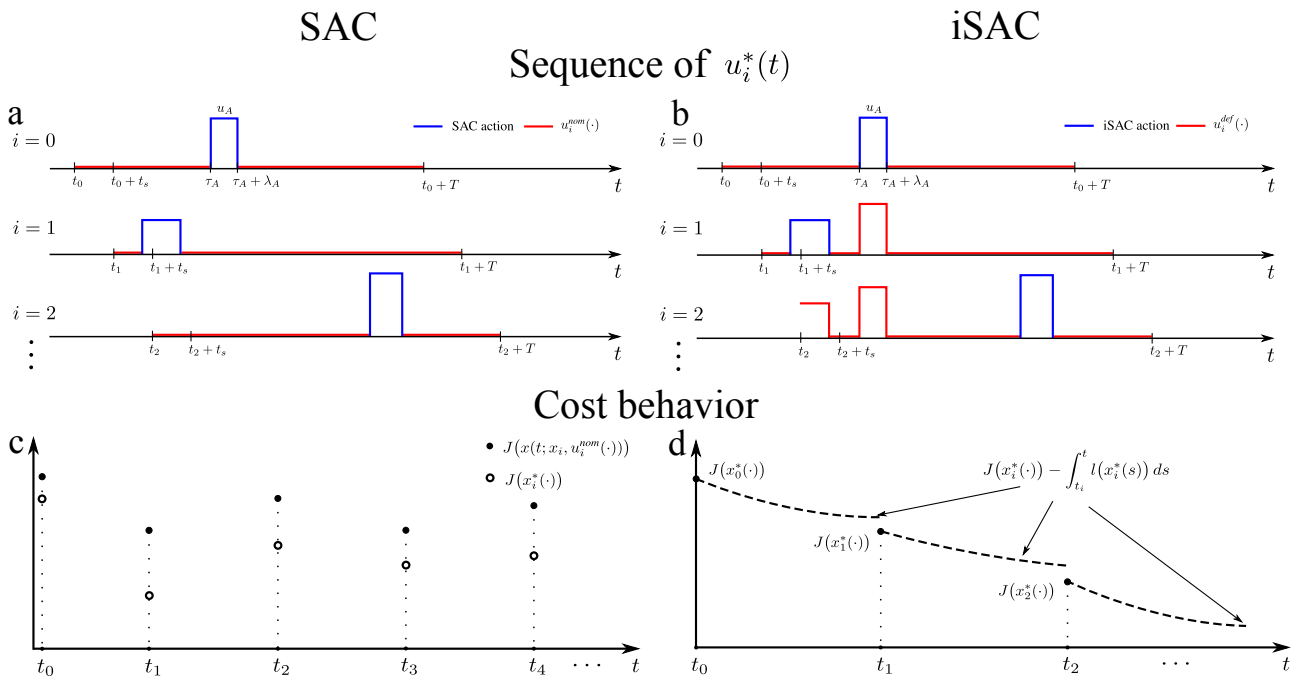

Fig. 4. Summary of differences between SAC and iSAC. Panels a and b underline the fact that iSAC stores previously calculated actions and iteratively improves the open-loop solution $u_i^*(t)$ (assuming zero nominal control). As a result, a contractive constraint can be employed in iSAC to establish a sufficient decrease condition for the cost as shown in panel d. Panel c shows the corresponding graph in SAC as dictated by (4).

B. Contractive constraint on cost

In SAC, each action is calculated such that (4) is satisfied. However, in general, this does not ensure that the cost will follow a decreasing trend. The explanation lies in the fact that (4) only guarantees that the cost will improve relative to a nominal input in the current time step, but not necessarily across time steps. As an example, Fig. 4c shows a general scenario where condition (4) is met. In this case, the system does not meet the desired specifications encoded in the cost, since the latter is not decreasing with time. Thus, even if (4) is satisfied, closed-loop stability cannot be established in SAC.

A solution to this problem is to modify condition (4) such that (2) sufficiently "contracts" between time steps [26]–[29]. Contractive constraints have been widely used in the MPC literature to show closed-loop stability as an alternative to methods relying on a terminal (region) constraint [1], [2], [16], [17], [21], [22]. A disadvantage of the latter is that they require the computation of a terminal region, which has to be calculated separately for each system of interest [30]–[32]. On the contrary, the contractive constraint approach can be applied to a variety of examples without modification and is easier to implement.

Specifically, instead of improving the cost relative to only applying nominal control in (4), we apply

$$J\big(x_i^*(\cdot)\big) - J\big(x_{i-1}^*(\cdot)\big) \leq -\int_{t_{i-1}}^{t_i} l\big(t, x_{i-1}^*(t)\big)\, dt, \qquad (11)$$

as a contractive constraint. Conditions similar to (11) also appear in terminal region methods, either in continuous or in discrete time, as an intermediate step used to prove closed-loop stability.

One problem that arises with (11) in SAC is that the quantities $J\big(x_i^*(\cdot)\big)$ and $J\big(x_{i-1}^*(\cdot)\big)$ cannot be related through the mode insertion gradient, unlike, e.g., $J\big(x_i^*(\cdot)\big)$ and $J\big(x_i^{nom}(\cdot)\big)$ that appear in the original condition (4) and can be related through (9). Thus, Proposition 1 can no longer be used to ensure that (11) is feasible in SAC. On the contrary, in iSAC, we can use the iterative nature of $u_i^*(t)$ to ensure feasibility of (11) through Proposition 1 for sufficiently small sampling time t_s. In particular, in iSAC we can transform (11) to an equation that is similar to (4) and includes the terms $J\big(x_i^*(\cdot)\big)$ and $J\big(x_i^{def}(\cdot)\big)$, which can be related through the mode insertion gradient $\frac{dJ}{d\lambda}$, as in (9).

By definition, (10) leads to $x_i^{def}(t) \equiv x_{i-1}^*(t)$ for $t \in [t_i, t_{i-1}+T]$ (see also Fig. 4b). We can then write

$$\begin{aligned} J\big(x_{i-1}^*(\cdot)\big) - J\big(x_i^{def}(\cdot)\big) &= \int_{t_{i-1}}^{t_i} l\big(t, x_{i-1}^*(\cdot)\big)\, dt \\ &+ m\big(t_{i-1}+T, x_{i-1}^*(t_{i-1}+T)\big) - \int_{t_{i-1}+T}^{t_i+T} l\big(t, x_i^{def}(t)\big)\, dt \\ &- m\big(t_i+T, x_i^{def}(t_i+T)\big). \end{aligned} \qquad (12)$$

Combining (12) with (11) and using (10) we get

$$\begin{aligned} J\big(x_i^*(\cdot)\big) - J\big(x_i^{def}(\cdot)\big) &\leq m\big(t_{i-1}+T, x_{i-1}^*(t_{i-1}+T)\big) \\ &- \int_{t_{i-1}+T}^{t_i+T} l\big(t, x_i^{def}(t)\big)\, dt - m\big(t_i+T, x_i^{def}(t_i+T)\big) \end{aligned} \qquad (13)$$

or equivalently

$$\begin{aligned} &J\big(x_i^*(\cdot)\big) - J\big(x_i^{def}(\cdot)\big) \leq \\ &\quad - \int_{t_{i-1}+T}^{t_{i-1}+t_s+T} l\big(t, x_i^{def}(t)\big) + \dot{m}\big(t, x_i^{def}(t)\big)\, dt = C. \end{aligned} \qquad (14)$$

Thus, we were able to transform (11) into a sufficient decrease condition similar to (4), i.e., a condition that involves $J\big(x_i^*(\cdot)\big)$ and $J\big(x_i^{def}(\cdot)\big)$. In terminal region methods, the integrand quantity in (14) is often required to be negative in some region of the state space and for some nominal control signal (see e.g. [1], [21]) to achieve closed-loop stability. Applying such a requirement in our case would make satisfaction of (14) trivial as the right-hand side of (14) would be positive and the left-hand side would be negative (Proposition 1). However, calculation of the terminal region and control is not straightforward and is also not necessary for iSAC. The following Proposition can be used in conjunction with Proposition 1 to ensure that the new condition (14) is (recursively) feasible.

Proposition 2: For $\alpha_d < 0$ there exists a sufficiently small sampling time t_s such that the sufficient cost decrease required by the contractive constraint (14) is attainable.

Proof: See Appendix. ■

The contractive constraint (14) can be applied in the line search process that determines the duration of an action (Section II-4) in lieu of (4). By applying this condition at each time step, we can guarantee that the cost value will decrease across time steps, and using this fact we can prove closed-loop stability (Section IV).

Open-loop problem in iSAC

Based on the above, the open-loop problem solved by iSAC at each time step is given by

$$\mathcal{B}(t_i,\, x_i): \tag{15}$$

Find action A such that (14) is satisfied

subject to

$$u_i^*(t) = \begin{cases} u_A & \tau_A \le t \le \tau_A + \lambda_A \\ u_i^{def}(t) & \text{else} \end{cases},$$

$\tau_A >= t_i,\ \tau_A + \lambda_A <= t_i + T,$

and (1) with $t \in [t_i, t_i + T]$ and $x(t_i) = x_i$.

Similar to the open-loop problem $\mathcal{P}$ in SAC, the solution $u_i^*(t)$ of $\mathcal{B}$ can be obtained in closed form by following the four steps in Fig. 2 and Section II, without relying on nonlinear programming solvers. The only difference is that in the solution process of $\mathcal{B}$, the superscripts *nom* are replaced by *def*. Our final result in this section utilizes the aforementioned assumptions and propositions to ensure that the open-loop problem $\mathcal{B}$ has a solution:

Proposition 3 (Existence of solution to $\mathcal{B}$): For sufficiently small sampling time t_s, the solution $u_i^*(t)$ to the open-loop problem $\mathcal{B}(t_i,\, x_i)$ exists for any x_i, t_i.

Proof: See Appendix. ■

IV. Global Stability Analysis

In this section, we provide global stability results for iSAC and discuss how iSAC can be used under a special case of disturbances and to track infeasible trajectories, i.e., trajectories that do not satisfy the dynamics (1).

A. Nominal Case

In this section we are considering nominal stability (not robust stability), i.e., the trajectories of the plant are assumed to coincide with the trajectories predicted by the model. Recall that the iSAC method does not optimize the cost (2) but rather improves it across time steps. It is often the case that minimization of the objective function in the open-loop optimization problem is not required to achieve stability of a model predictive controller. Sometimes, a decrease in the cost at every iteration is sufficient to guarantee stability [20], [21], [33], [34]. In view of these observations, we now present the following Theorem:

Theorem 1 (Asymptotic Stability): For sufficiently small sampling time t_s, the closed-loop system resulting from applying iSAC is asymptotically stable in the sense that $||\overline{x}_{cl}(t)|| \to 0$ as $t \to \infty$.

Proof: See Appendix. ■

Theorem 1 does not imply Lyapunov stability, but rather establishes the usual notion of attractiveness. The importance of t_s in establishing closed-loop stability is also underlined; if the assumptions of the Theorem hold, by appropriate selection of t_s, we can ensure that the contractive constraint is feasible, and thus, asymptotic stability.

Cost as a Lyapunov function

Now that asymptotic stability has been proved, we will show that, under certain assumptions, the objective (2) is a Lyapunov function for the closed-loop system that decreases at intervals of prediction horizons.

Theorem 2 (Objective Function as a discrete-time Lyapunov Function): Assume that f is bounded such that $||f|| \le \xi ||x||$ for some finite constant $\xi > 0$, and that the cost (2) is of the form

$$J\big(t_i, x(\cdot)\big) = \frac{1}{2}\int_{t_i}^{t_i+T} ||x(t)||_Q^2\, dt + \frac{1}{2}||x(t_i+T)||_{P_1}^2, \tag{16}$$

with t_i taking values in a discrete set for $i \in \mathbb{N}$. Then, for sufficiently small t_s, (16) is a discrete-time Lyapunov function for the closed-loop system.

Proof: The proof can be found in the Appendix. ■

Remark 3: Theorem 2 proves that a quadratic cost may be used as a discrete-time Lyapunov function in our method. The Lyapunov function does not have to be continuously decreasing along solution trajectories, thus only establishing attractiveness similar to Theorem 1. Additionally, even though (16) is continuously differentiable, there is no assumption on the continuity with respect to the state. These points are important, since they allow application of this theorem to problems that potentially do not admit a continuous-time Lyapunov function which is also continuous in the state, e.g. nonholonomic systems [35].

B. Asymptotically Decaying Disturbances

This section provides stability results for a special case of disturbances. Here, we assume that the plant is described by the following differential equations (instead of (1))

$$\dot{x}^p = f(t, x^p, u) + \eta(t) = g(t, x^p) + h(t, x^p)\, u + \eta(t), \tag{17}$$

where $x^p(t)$ represents the trajectory resulting from applying control u and the effect of an unknown additive disturbance $\eta(t)$. The disturbance is bounded and asymptotically decaying; in particular, we assume the following:

Assumption 6: The disturbance $\eta(t)$ is bounded, i.e., $||\eta_i(t)|| \leq \delta_i < \infty$ for all $t \in [t_i, t_i + T]$ and all $i > 0$. Furthermore, $\eta(t)$ is asymptotically decaying, i.e., $\delta_i \to 0$ as $i \to \infty$.

Closed-loop stability under $\eta(t)$ is provided by the following theorem.

Theorem 3 (Asymptotic Stability under Asymptotically Decaying Disturbances): For sufficiently small t_s, the closed-loop system under the disturbance $\eta(t)$ is asymptotically stable in the sense that $||\overline{x}_{cl}(t)|| \to 0$ as $t \to \infty$.

Proof: The proof can be found in the Appendix. ■

As seen in the proof, for as long as the disturbance is acting, the cost is allowed to increase. As time progresses, the disturbance attenuates and the contractive constraint becomes feasible again.

C. Remarks on tracking of non-admissible trajectories

When designing a control task, there are often cases where asymptotic stability/convergence to the desired equilibrium cannot be achieved. For example, in trajectory tracking, it is not always feasible to identify trajectories that satisfy the dynamics of the system of interest, unless there is special structure one can exploit, e.g., differential flatness [36]. The control goal in this case is to ensure that the deviation of the generated (feasible) open-loop trajectory from the desired one is bounded, i.e., $||x_i^*(t) - x^d(t)|| \leq \Delta_i$ for $t \in [t_i, t_i + T]$ with $\Delta_i \in [0, \infty)$. Unlike the decaying disturbance case, the upper bound Δ_i does not have any structure that can be exploited. Thus there is no guarantee that $\overline{x}_{cl}(t) \to x^d(t)$ as $t \to \infty$. Nevertheless, even for this scenario, our simulation results show that iSAC keeps the deviation from the desired infeasible trajectories bounded.

V. Simulation Results

In this section we demonstrate the flexibility and versatility of iSAC through application to a variety of challenging systems. For each example, we include literature review on related control methods, and we also compare the performance of iSAC on NASA's TRACE spacecraft [37] to pseudospectral infinite-horizon control [38], [39]. It is important to note that unlike other system-specific methods mentioned in this section, our approach requires no modifications across examples; only the dynamic model that is being controlled and the relevant parameters of iSAC are changed for each example. Also, all the examples presented here run much faster than real time, indicating applicability to hardware implementations. For better visualization of the results, the reader is encouraged to visit https://vimeo.com/219702474.

The control objective is encoded using a quadratic cost throughout this section (Assumption 3 is thus satisfied). In the TRACE example, we also incorporate state constraints by introducing penalty terms in the cost. The examples also include cases of feasible and infeasible tracking, as well as disturbance rejection. The sampling time t_s for each example was appropriately selected to achieve closed-loop stability (Theorems 1 and 3) when feasible. Finally, a summary of iSAC-specific parameters for each example is given in Table I, while Fig. 5 shows the resulting trajectories for the examples described below. Our method is shown to be successful in these scenarios, leading to asymptotic stabilization when feasible.

A. Cart-pendulum inversion

The cart-pendulum inversion is a benchmark problem in control theory [11], [40]. The dynamics of the system and the parameters used in the simulations that follow can be found in [41]. The state vector for the system includes the angle and angular velocity of the pendulum and the horizontal position and velocity of the cart, $[\theta, \dot{\theta}, x_c, \dot{x}_c]$. The control input is the horizontal acceleration of the cart with zero u^{nom}. The desired setpoint in this example was $x^d = [0, 0, 1, 0]$. The cost weights were selected as $Q = Diag[20\,, 0\,, 5\,, 0]$, $P = Diag[0.1\,, 0\,, 5\,, 0]$ and $R = 0.3$.

Comparing the cart-pendulum response in Fig. 5 with Fig. 1, it is clear that iSAC, unlike SAC, can achieve final stabilization. The proposed method is able to solve the inversion problem by pumping energy into the system without explicitly encoding this behavior in the objective. In contrast, alternative methods switch between separately derived controllers [42] to achieve the same result. Also, many methods (see [43], [44]) rely on energy-based objectives to solve the swing-up problem for the acrobot and pendulum systems, primarily because state-tracking objectives result in many local minima that prevent convergence to desirable trajectories. It is noteworthy that, iSAC is able to invert the system using a state-tracking objective while bypassing local minima. The corresponding cost for this example is shown in Fig. 6.

B. Car-like vehicle

The dynamics used in the simulations are

$$\dot{x}_c = v\cos\theta,\ \dot{y}_c = v\sin\theta,\ \dot{v} = u_1,\ \dot{\theta} = u_2, \qquad (18)$$

where x_c, y_c denote the position of the car, θ is the angle with respect to the horizontal axis and v is the forward velocity. The iSAC method directly controls the acceleration and the angular velocity of the car. This nonholonomic system violates Brockett's necessary condition for smooth or even continuous stabilization [45], which makes the control design problem challenging. Since iSAC automatically generates a discontinuous control law, it is not subject to this condition.

1) Parallel parking:

The desired setpoint for the parallel parking problem was $x^d = [0, 0, 0, 0]$ and the cost weights were selected as $Q = Diag[1\,, 15\,, 0.8\,, 0.8]$, $P = Diag[0\,, 25\,, 0\,, 0]$ and $R = Diag[0.1\,, 0.1]$. As seen in Fig. 5, iSAC successfully drives the system to the origin. Nominal control u^{nom} was set to zero for both inputs. The corresponding cost is in Fig. 6.

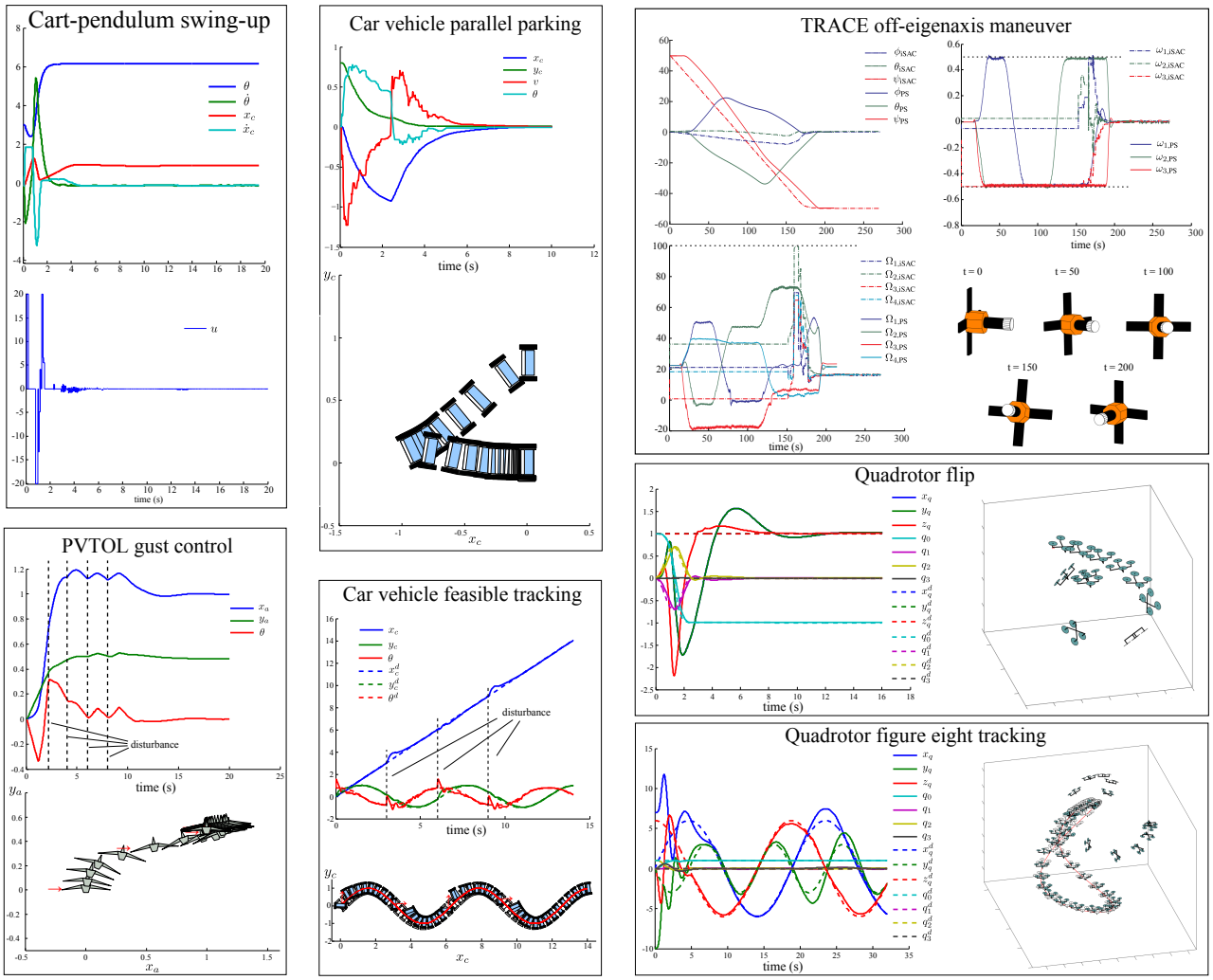

Fig. 5. Summary of results; the iSAC method stabilizes the presented systems successfully. Notice that our method is effective even in the presence of decaying disturbances and, also, keeps the deviation from the desired trajectory bounded in infeasible tracking. For better visualization of the results, the reader is encouraged to visit https://vimeo.com/219702474.

2) Feasible trajectory tracking under disturbance:

The desired trajectory in this example was chosen as $\left(x_c^d(t), y_c^d(t), \theta^d(t)\right) = (t, \sin t, \tan^{-1}\cos t)$ and the cost weights were $Q = Diag[100\,, 100\,, 0\,, 10]$ and $R = Diag[0.1\,, 0.1]$. Nominal control u^{nom} was set to zero for both inputs. External disturbances acted instantaneously on the system on four occasions, i.e., at $t = 0, 3, 6,$ and 9s, each perturbing v and θ by $+3\,\mathrm{m/s}$ and $+\frac{\pi}{4}$ rad respectively. The effect of the disturbance can be seen in the corresponding plot in Fig. 5; while the car was tracking the desired trajectory, the disturbance pushed the system away every time it acted on the system. After the last disturbance attenuated the control successfully steered the system back to the desired trajectory. The corresponding cost is in Fig. 6.

A great deal of work on nonholonomically constrained car-like models was completed in the 1990's using nonsmooth or time-varying control laws aiming to overcome stabilizability limitations traditional techniques [46]–[48]. Alternative methods that achieve similar results in the parking problem include fuzzy controllers as in [49] and dynamic feedback linearization [50]. Other approaches that intrinsically lead to piecewise continuous controls like iSAC include MPC as in [26], [51]. The former paper utilizes contractive constraints on the state and, although successful, results in a straight line path between the initial and final configuration. The latter uses the aforementioned terminal region constraints to achieve stabilization, but, unlike iSAC, the method is designed specifically for the car-vehicle system.

C. Gust control of planar vertical take-off landing (PVTOL) aircraft

The dynamic model used in this example is given in [52]. The state vector for the system includes the position of the aircraft, the angle with respect to the horizontal axis and the corresponding velocities, i.e., $[x_a, \dot{x}_a, y_a, \dot{y}_a, \theta, \dot{\theta}]$. The control inputs of the system are thrust (directed out the bottom of the aircraft) and the rolling moment. Lastly, the coupling parameter ϵ that appears in the model was chosen as $\epsilon = 0.3$.

Note that ϵ couples the rolling moment input with the lateral acceleration of the aircraft; a positive value $\epsilon > 0$ means that applying a (positive) moment to roll left produces an acceleration to the right, making this a non-minimum phase system.

Nominal control u^{nom} was set to 1N for the thrust and zero for the rolling moment. The desired setpoint in this example was $x^d = [1, 0, 0.5, 0, 0, 0]$ and the cost weights were selected as $Q = Diag[15\,, 3\,, 15\,, 3\,, 3\,, 0]$ and $R = Diag[0.1\,, 0.1]$. External disturbances acted instantaneously on the system on five occasions, i.e., at $t = 0, 2, 4, 6$, and 8s, each perturbing $\dot{x}_a$ by $+0.1\,\mathrm{m/s}$. The behavior of the system was similar to the one observed in the car-like vehicle; while the control was steering the system towards the desired setpoint, the disturbance drove it away until the last disturbance.

Figure 5 shows the resulting trajectories. The control of PVTOL aircrafts has been extensively studied using, e.g., gain-scheduling [53], robust control [54] and input-output linearization [52]. In the latter case, the system is decoupled into simpler subsystems to facilitate the control process (a similar approach is commonly followed in quadrotor control). It is important to note that iSAC is able to successfully control the system using the full dynamics and without relying on separate controllers, unlike, e.g. in [53]. Finally, other techniques that have been applied on VTOL aircrafts and unmanned aerial vehicles (UAVs) in general include sliding mode control and backstepping as explained in the following example [55].

D. Quadrotor

For this example, a quaternion-based model is used because it leads to polynomial, singularity-free dynamical equations for the quadrotor. The model can be found in [56] and the model parameters we used are the same as in [6]. The 13-dimensional state vector consists of the position and velocity of the center of mass, the angular orientation with respect to the inertial frame expressed in quaternions and the angular velocity expressed in the body frame, i.e., $[x_q, y_q, z_q, \dot{x}_q, \dot{y}_q, \dot{z}_q, q_0, q_1, q_2, q_3, p, q, r]$. The control inputs are the squared angular velocities of the rotors, which are converted to upward thrust for each rotor when multiplied by an appropriate constant. For both examples that follow, nominal control u^{nom} for each input was set to a constant value such that the net upward thrust is approximately equal to gravity force when the quadrotor is parallel to the ground.

1) Flip:

To perform a flip, the desired quaternion vector was a flip trajectory with the respective Q weights set equal to 1500. After the flip, the desired position for the center of mass of the quadrotor was $x^d = [1, 1, 1]$ with the respective Q weights set equal to 3. Finally, $R = Diag[0.1\,, 0.1\,, 0.1\,, 0.1]$. Figures 5 and 6 show the resulting trajectories and cost.

2) Infeasible three dimensional figure eight tracking:

In this example, the quadrotor starts in an upside-down position and is set to track a three-dimensional figure eight (respective Q weights set equal to 10) while keeping the quaternion states at $\left(q_0^d(t), q_1^d(t), q_2^d(t), q_3^d(t)\right) = [1, 0, 0, 0]$ (respective Q weights set equal to 1000). Thus, the quadrotor is requested to track an aggressive 3-dimensional maneuver without changing its angular orientation, which is infeasible. The weight $R = Diag[0.1\,, 0.1\,, 0.1\,, 0.1]$. As seen in Fig. 5, at the beginning of the simulation the quadrotor reverts its angular orientation and then it starts tracking the desired trajectory keeping the error small. This behavior is automatically generated by our method. The corresponding cost is in Fig. 6.

Many different approaches have been followed for control of quadrotors over the years. Linear control methods like LQ and PID synthesis [57], [58] are popular for their simplicity. They utilize decoupled or simplified dynamics and also exploit differential flatness [36] in order to track feasible trajectories. For example, in [59] a quadrotor flip is implemented using a dedicated attitude controller. In contrast iSAC controls the full nonlinear dynamics of the system, and has good performance even with infeasible trajectories as shown in the simulation examples. No additional steps are required to apply our method to this system, unlike, e.g., feedback linearization [60] or sliding mode control [55]. The simplicity and versatility of iSAC is one of the main points that this section demonstrates; the same exact approach is being applied to a variety of challenging systems in real time, by replacing only the dynamic model that is being controlled.

E. Incorporating State Constraints – TRACE Simulation

In 2010, the first in-orbit, time-optimal maneuver was carried out, onboard the NASA's TRACE spacecraft [37]. The TRACE is a three-axis stabilized, zero-momentum, system that employs a set of four reaction wheels for primary attitude control. The in-orbit maneuvering strategy was calculated by pseudospectral (PS) optimal control theory [38], [39].

For this example, the dynamics and parameters of the system were taken from [37]. The inputs are the torques applied to the reaction wheels. The 11-dimensional state vector consists of the angular orientation expressed as a quaternion vector, the vector of angular body rates and the reaction wheel rates, i.e., $[q_0, q_1, q_2, q_3, \omega_1, \omega_2, \omega_3, \Omega_1, \Omega_2, \Omega_3, \Omega_4]$. The control objective was a rest-to-rest $100°$ time-optimal reorientation about the spacecraft's z-axis. The reaction wheels in the initial configuration are moving at constant rates of 20 rad/s to provide the necessary energy for the desired maneuver.

To introduce saturation constraints on the state of the form $-x_{sat} < x < x_{sat}$ we included bounded, differentiable penalty functions in the cost as

$$B(x) = \frac{\bar{Q}}{1 + e^{\pm a(x \pm x_{sat})}} \tag{19}$$

where $\bar{Q}$ and a are parameters to be selected. These can be directly integrated into the running cost l without violating any of the stated assumptions. For this simulation the only nonzero Q weights corresponded to the quaternion vector and were set equal to 1500 with $R = Diag[0.1\,, 0.1\,, 0.1\,, 0.1]$, thus prioritizing the maneuvering time. The penalty functions were applied at the body rates ω and the reaction wheel rates Ω according to [37] as $|\omega_i| < 0.5$ degrees/s, $i = 1, 2, 3$ and $|\Omega_j| < 100$ rad/s, $j = 1, 2, 3, 4$. Also, $\bar{Q} = 100$ and $a = 80$.

TABLE I
iSAC PARAMETER VALUES USED IN SIMULATIONS

Example	V-A	V-B1	V-B2	V-C	V-D1	V-D2	V-E
α_d	$-15\,J\big(x_i^{def}(\cdot)\big)$	$-100\,J\big(x_i^{def}(\cdot)\big)$		$-10\,J\big(x_i^{def}(\cdot)\big)$	-5000		-5000
T	1.2s	1.2s	0.35s	3s	3s	2s	15s
t_s	0.01s	0.02s	0.03s	0.02s	0.02s	0.05s	0.05s
$[u_{min}, u_{max}]$	$[-20, 20]$ m/s^2	$[-10, 10]$ m/s^2 $[-4, 4]$ rad/s		$[0, 5]$ N $[-100, 100]$ N m	$[0, 12]$ rad^2/s^2 for all inputs		$[-0.05, 0.05]$ N m for all inputs

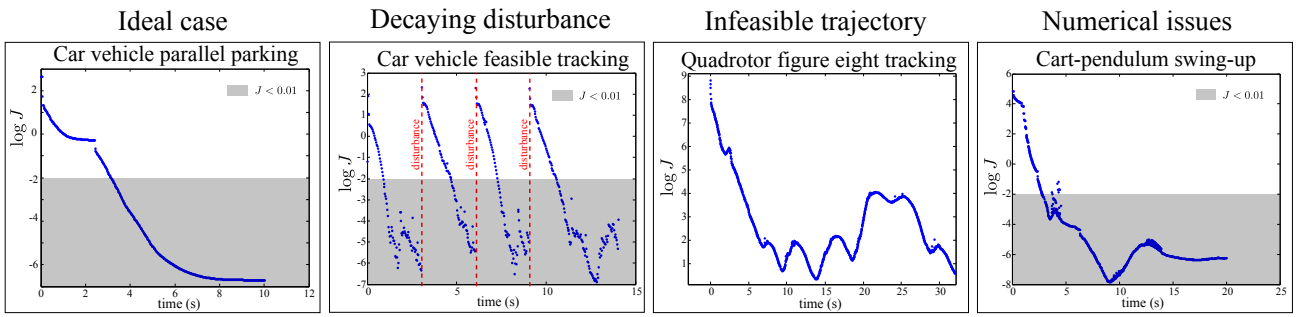

Fig. 6. Sample cost behavior in the presented examples. As explained in Section V-F, due to numerical issues, the contractive constraint may not be satisfied momentarily near the desired equilibrium, leading to a small cost increase. Nevertheless, the system is effectively already stabilized at that point (compare corresponding trajectories from Fig. 5). Columns two and three show the cost behavior when acting under the presence of an external bounded disturbance and tracking infeasible trajectories respectively, where the cost is not expected to continuously decrease.

Finally, nominal control u^{nom} was set to zero for all inputs. The remaining iSAC parameters are given in Table I.

The iSAC simulation results as well as the real flight results that correspond to the PS method (see [61]) are shown in Fig. 5. First, notice that both methods successfully complete the maneuver while satisfying the constraints on the body rates and the wheel rates throughout the simulation. Despite the fact that the PS method directly optimizes the maneuver time in the cost function (we indirectly do so by weighing the state error), it took approximately 170 seconds for iSAC to complete the maneuver, i.e., 10 seconds less than the PS method (181 seconds). Also, the target quaternion vector in the PS method was updated every 1 second, but no relevant discussion was provided for the execution time in [61]. Our simulation with iSAC took less than 40 seconds for the 4-minute trajectory shown in Fig. 5, and computations were efficient enough to be executed every $t_s = 0.05$s.

Additionaly, note that both methods follow an off-eigenaxis rotation, i.e., a rotation about a variable axis. The explanation for this lies in the fact that the body masses were not symmetrically distributed on the spacecraft, and as a result, the shortest-time maneuver did not correspond to the shortest angle path. Nevertheless, ϕ and θ in iSAC deviate much less from zero and the same is true for the respective angular velocities ω_1 and ω_2. Finally, the wheel rates Ω are smaller for almost all time, indicating that iSAC uses less control authority than the PS method over the time horizon.

F. Discussion and Limitations

In general, iSAC performs well in the variety of scenarios presented. When the task is feasible, trajectories are asymptotically stable and the cost is generally decreasing from time step to time step as a result of applying condition (11). Nevertheless, near the desired equilibrium, the behavior of the cost sometimes deviates from what one would expect, even when the control task is feasible. As an example, in the rightmost panel of Fig. 6 the cost is not continuously decreasing near the equilibrium, although the system is effectively stabilized when the unexpected behavior occurs (the cost value is less than 0.01). There are three possible explanations for this. First, in order to speed up the solution process, the line search (Section II-4) in all examples presented in this paper was stopped after 10 iterations. While for the majority of cases 10 iterations were sufficient to satisfy the sufficient descent condition (11), on some occasions more iterations were necessary. As a result, a small increase was observed in the cost in these time steps. Another factor to consider is numerical tolerance. From the simplified model (9), depending on the value of the mode insertion gradient, the required λ value that achieves the sufficient decrease descent could be smaller than the minimum tolerance of numerical integrators. This issue appears often near the equilibrium where systems are more sensitive to the duration of an action. Finally, from Proposition 2, it is possible that the selected sampling time t_s was not sufficiently small to (recursively) satisfy (14). Since the open-loop problem in iSAC can be solved quickly and efficiently, a strategy for selecting t_s is to start with small values and gradually increase t_s until (14) is violated.

VI. Conclusion

In this paper we presented iSAC, a model-predictive approach for control of nonlinear systems. As the time horizon progresses, our method sequences together optimal actions and synthesizes piecewise continuous control laws. Some key

characteristics of iSAC include: a) analytic solution to the open-loop problem—there is no need to rely on nonlinear programming solvers, b) iterative update of the open-loop solution and c) use of continuous dynamics while incorporating control constraints without additional overhead. Due to these points, iSAC leads to computationally efficient solutions. To establish closed-loop stability, we applied a contractive constraint on the cost. Compared to methods relying on terminal region constraints, the contractive constraint alleviates the need to calculate a terminal region. We also investigated different control scenarios ranging from feasible and infeasible trajectory tracking to set point stabilization with or without external disturbances. Finally, we presented simulation examples using a variety of challenging systems to demonstrate the applicability and flexibility of our method.

APPENDIX

A. Proof of Proposition 1

Substituting (7) into the mode insertion gradient formula given in (6) we get (we omit superscripts for brevity)

$$\begin{aligned}\frac{dJ}{d\lambda}(t) &= \Gamma(t)\left(\Gamma(t)^T\Gamma(t)+R^T\right)^{-1}\Gamma(t)^T\,\alpha_d \\ &= \alpha_d\,||\Gamma(t)||^2_{(\Gamma(t)^T\Gamma(t)+R^T)^{-1}} < 0, \end{aligned} \tag{20}$$

where $\Gamma(t)^T = h\big(t,x(t)\big)^T\rho(t)$. The term $\Gamma(t)^T\Gamma(t)$ produces a positive semi-definite matrix, and adding $R>0$ yields a positive definite matrix. Because the inverse of a positive definite matrix is positive definite, the quadratic norm $||\Gamma(t)||^2_{(\Gamma(t)^T\Gamma(t)+R^T)^{-1}}$ is positive for $\Gamma(t)^T \neq 0$ (Assumption 5). Therefore, if $\alpha_d < 0$, $\frac{dJ}{d\lambda}(t) < 0$. ■

B. Proof of Proposition 2

From Proposition 1 and (9), we know that $J\big(x_i^*(\cdot)\big) - J\big(x_i^{def}(\cdot)\big) = \Delta J < 0$ for each λ in a neighborhood around $\lambda \to 0^+$ if $\alpha_d < 0$. By continuity assumptions for m and l, C in (14) is continuous with respect to t_s since t_s is part of the integral limits. Thus, it follows that there exists a sufficiently small t_s such that $\Delta J < C$. ■

C. Proof of Proposition 3

The open-loop problem $\mathcal{B}$ is solved by following the same four sequential steps as in the open-loop problem $\mathcal{P}$ in Section II. The only difference is that in the solution process of $\mathcal{B}$, the superscripts *nom* are replaced by *def*. To show that $u_i^*(t)$ exists, we will show that each of the four solution steps has a solution:

1) The first step in Section II-1 involves calculating the solutions to (1) and (5). Since the default control is in general discontinuous, if solutions are interpreted as sample and hold (CLSS) solutions (see [62]), existence follows directly from Assumptions 2, 3.

2) In the second step our method in Section II-2 calculates the optimal action schedule u_s^* by minimizing (6). Because (6) is convex with a continuous first variation from Assumptions 1-3, solutions (7) exist and are unique, which is also both necessary and sufficient for global optimality of (6).

3) In the third step in Section II-3 the process selects τ_A and u_A by minimizing (8). Because $\tau_A \in [t_i, t_i+T]$ and (8) is in general piecewise continuous (and thus bounded), a solution to this one-dimensional problem exists.

4) Finally, from Proposition 2, the backtracking process in Section II-4 is guaranteed to find a duration that satisfies condition (14) for sufficiently small t_s.

Since all four subproblems have solutions, the open-loop solution $u_i^*(t)$ exists. ■

D. Proof of Theorem 1

The proof has two parts; Lemma 1 shows that the integral $\int_{t_0}^t M\big(x(s)\big)\,ds$ is bounded for $t\to\infty$ (see Assumption 3). The latter is used in conjunction with a well-known lemma found, e.g., in [63], [64], to prove asymptotic convergence in the second part of the proof.

First, define

$$V\big(\alpha,\beta,x(\cdot)\big) = \int_\alpha^\beta l\big(s,x(s)\big)\,ds + m\big(\beta,x(\beta)\big). \tag{21}$$

Consider the horizon interval $[t_i, t_i+T]$. Let $u_i^*(t)$ be the solution to the open-loop problem $\mathcal{B}(t_i,x_i)$ and $x_i^*(t)$ the corresponding state trajectory. Clearly, $V\big(t_i,t_i+T,x_i^*(\cdot)\big) = J\big(x_i^*(\cdot)\big)$. Then, for $t\in[t_i,t_i+T]$

$$\begin{aligned}&V\big(t,t_i+T,x_i^*(\cdot)\big) \\ &\quad = V\big(t_i,t_i+T,x_i^*(\cdot)\big) - \int_{t_i}^t l\big(s,x_i^*(s)\big)\,ds.\end{aligned} \tag{22}$$

Based on the above, we now present the following lemma.

Lemma 1: For small t_s, all $t\in[t_i,t_i+T]$ and all $i\in\mathbb{N}$

$$\begin{aligned}\int_{t_0}^t M\big(\overline{x}(s)\big)ds &\le V\big(t_0,t_0+T,x_0^*(\cdot)\big) \\ &\quad - V\big(t,t_i+T,x_i^*(\cdot)\big),\end{aligned} \tag{23}$$

with

$$\overline{x}(t) = \begin{cases}\overline{x}_{cl}(t) & \text{for } t<t_i \\ x_i^*(t) & \text{else}\end{cases}.$$

Proof: From (11) and Assumption 3 we get that

$$\begin{aligned}&V\big(t_i,t_i+T,x_i^*(\cdot)\big) - V\big(t_{i-1},t_{i-1}+T,x_{i-1}^*(\cdot)\big) \\ &\quad \le -\int_{t_{i-1}}^{t_i} M\big(x_{i-1}^*(s)\big)ds\end{aligned} \tag{24}$$

holds in $[t_i,t_i+T]$ for sufficiently small t_s (Proposition 2). Using the corresponding inequalities from the previous time steps until $[t_0,t_0+T]$ and the fact that the open-loop solution $u_i^*(\cdot)$ is only applied in $[t_i,t_{i+1}]$ we can write

$$\begin{aligned}\int_{t_0}^{t_i} M\big(\overline{x}_{cl}(t)\big)ds &\le V\big(t_0,t_0+T,x_0^*(\cdot)\big) \\ &\quad - V\big(t_i,t_i+T,x_i^*(\cdot)\big).\end{aligned} \tag{25}$$

Also, from (22) we have

$$\begin{aligned}\int_{t_i}^t M\big(x_i^*(s)\big)\,ds &\le V\big(t_i,t_i{+}T,x_i^*(\cdot)\big) \\ &\quad - V\big(t,t_i+T,x_i^*(\cdot)\big).\end{aligned} \tag{26}$$

Adding (25) and (26) leads to (23) and Lemma 1 is proved. ■

From Lemma 1, because $V\big(t, t_i+T, x_i^*(\cdot)\big) \geq 0$ and M is positive definite, we can deduce that $\int_{t_0}^{t} M\big(x(s)\big)\,ds$ is bounded for $t \to \infty$. We also have that $x_i^*(\cdot)$, and thus $\overline{x}_{cl}(\cdot)$, are bounded and from the properties of f, $\dot{x}_i^*(\cdot)$ and $\dot{\overline{x}}_{cl}(\cdot)$ are bounded as well. These facts combine with the following well-known lemma to prove asymptotic convergence.

Lemma 2: Let $x : \mathbb{R}^+ \to \mathcal{X}$ be an absolutely continuous function and $M : \mathcal{X} \to \mathbb{R}^+$ be a continuous, positive definite function ($0 \in \mathcal{X}$). If

$||x(\cdot)||_{L^\infty(\mathbb{R}^+)} < \infty$,
$||\dot{x}(\cdot)||_{L^\infty(\mathbb{R}^+)} < \infty$, and
$\lim_{T\to\infty} \int_0^T M\big(x(t)\big)\,dt < \infty$
then $x(t) \to 0$ as $t \to \infty$.

Proof: The proof can be found, e.g., in [63], [64]. ■

Theorem 1 is proved. ■

E. Proof of Theorem 2

We will show that, for all i, there exist positive constants a, b, c, such that
1. $a||x_i^*||^2 \leq J\big(t_i, x_i^*(\cdot)\big) \leq b||x_i^*||^2$
2. $J\big(t_i, x_i^*(\cdot)\big) - J\big(t_{i-1}, x_{i-1}^*(\cdot)\big) \leq -c||x_{i-1}^*||^2$.

Using Assumption 4, we can find a constant $d > 0$ such that

$$||x_i^*(t)|| \leq d||x_i^*||, \forall t \in [t_i, t_i+T],\ i \in \mathbb{Z}^+ \tag{27}$$

with $x_i^* = x_i^*(t_i)$. Since u is constrained, this is always true except for systems with finite escape times which are already ruled out from Assumption 4.

• *Upper bound on* $J\big(t_i, x_i^*(\cdot)\big)$: From (16) and (27) we have

$$\begin{aligned} J\big(t_i, x_i^*(\cdot)\big) &\leq \frac{1}{2}T\lambda_{\max}(Q)d^2||x_i^*||^2 + \frac{1}{2}\lambda_{\max}(P_1)d^2||x_i^*||^2 \\ &= \frac{d^2}{2}\big(T\lambda_{\max}(Q) + \lambda_{\max}(P_1)\big)||x_i^*||^2 = b||x_i^*||^2. \end{aligned} \tag{28}$$

• *Lower bound on* $J\big(t_i, x_i^*(\cdot)\big)$: Using (27), the reverse triangle inequality and $||f|| \leq \xi||x||$ we have

$$\begin{aligned} ||x_i^*(t)|| &\geq ||x_i^*|| - \int_{t_i}^{t} ||f||\,d\tau \\ &\geq ||x_i^*|| - \int_{t_i}^{t} \xi||x_i^*(\tau)||\,d\tau \\ &\geq [1 - \xi d(t-t_i)]||x_i^*||. \end{aligned}$$

It then follows, for example, that

$$||x_i^*(t)|| \geq \frac{||x_i^*||}{2} \text{ for } t \in \left[t_i, t_i + \frac{1}{2\xi d}\right]. \tag{29}$$

Thus, we have two cases to consider:
1) $t_i + T \leq t_i + \frac{1}{2\xi d}$, or $T \leq \frac{1}{2\xi d}$.
In this case,

$$\begin{aligned} J\big(t_i, x_i^*(\cdot)\big) \geq \frac{1}{2}\int_{t_i}^{t_i+T} \lambda_{\min}(Q)||x_i^*(t)||^2\,dt \\ + \frac{1}{2}\lambda_{\min}(P_1)||x_i^*(t_i+T)||^2 \end{aligned}$$

or

$$J\big(t_i, x_i^*(\cdot)\big) \geq \frac{1}{8}T\lambda_{\min}(Q)||x_i^*||^2. \tag{30}$$

2) $t_i + T \geq t_i + \frac{1}{2\xi d}$, or $T \geq \frac{1}{2\xi d}$.
In this case

$$\begin{aligned} J\big(t_i, x_i^*(\cdot)\big) \geq \frac{1}{2}\int_{t_i}^{t_i+\frac{1}{2\xi d}} \lambda_{\min}(Q)||x_i^*(t)||^2\,dt \\ + \frac{1}{2}\lambda_{\min}(P_1)||x_i^*(t_i+T)||^2 \end{aligned}$$

or

$$J\big(t_i, x_i^*(\cdot)\big) \geq \frac{1}{16\xi d}\lambda_{\min}(Q)||x_i^*||^2. \tag{31}$$

Thus, it follows from (30), (31) that

$$\begin{aligned} J\big(t_i, x_i^*(\cdot)\big) &\geq \min\left\{\frac{1}{8}T\lambda_{\min}(Q), \frac{1}{16\xi d}\lambda_{\min}(Q)\right\}||x_i^*||^2 \\ &= a||x_i^*||^2. \end{aligned} \tag{32}$$

• *Upper bound on* $J\big(t_i, x_i^*(\cdot)\big) - J\big(t_{i-1}, x_{i-1}^*(\cdot)\big)$: From (11) and (29) we have the following two cases to investigate.
1) $t_i \leq t_{i-1} + \frac{1}{2\xi d}$, or $t_s \leq \frac{1}{2\xi d}$.
In this case,

$$\begin{aligned} &J\big(t_i, x_i^*(\cdot)\big) - J\big(t_{i-1}, x_{i-1}^*(\cdot)\big) \\ &\leq -\int_{t_{i-1}}^{t_i} \lambda_{\min}(Q)||x_{i-1}^*(t)||^2\,dt \leq -\frac{1}{4}t_s\lambda_{\min}(Q)||x_{i-1}^*||^2. \end{aligned} \tag{33}$$

2) $t_i \geq t_{i-1} + \frac{1}{2\xi d}$, or $t_s \geq \frac{1}{2\xi d}$.
In this case

$$\begin{aligned} &J\big(t_i, x_i^*(\cdot)\big) - J\big(t_{i-1}, x_{i-1}^*(\cdot)\big) \\ &\qquad \leq -\int_{t_{i-1}}^{t_{i-1}+\frac{1}{2\xi d}} \lambda_{\min}(Q)||x_{i-1}^*(t)||^2\,dt \\ &\qquad \leq -\frac{1}{8\xi d}\lambda_{\min}(Q)||x_{i-1}^*||^2. \end{aligned} \tag{34}$$

Thus, it follows from (33), (34) that

$$\begin{aligned} &J\big(t_i, x_i^*(\cdot)\big) - J\big(t_{i-1}, x_{i-1}^*(\cdot)\big) \\ &\leq -\min\left\{\frac{1}{4}t_s\lambda_{\min}(Q), \frac{1}{8\xi d}\lambda_{\min}(Q)\right\}||x_{i-1}^*||^2 = -c||x_{i-1}^*||^2. \end{aligned} \tag{35}$$

Theorem 2 is proved. ■

F. Proof of Theorem 3

The difference between the dynamics of the plant and the model used by iSAC for the control computation at each time step i and $t \in [t_i, t_i+T]$ is given by

$$\dot{x}_i^p(t) - \dot{x}_i^*(t) \quad = f(t, x^p, u_i^*) - f(t, x_i^*, u_i^*) + \eta_i(t). \tag{36}$$

The states of this model are updated using feedback at every t_i, so $x_i^*(t_i) = x^p(t_i)$ for all i. Then, we can integrate (36) to obtain

$$\begin{aligned} &x_i^p(t) - x_i^*(t) \\ &= \int_{t_i}^{t} [f\big(s, x^p(s), u_i^*(s)\big) - f\big(s, x_i^*(s), u_i^*(s)\big)] + \eta_i(s)\,ds. \end{aligned}$$

Therefore, using Assumption 6, we can write

$$\begin{aligned}
&||x_i^p(t) - x_i^*(t)|| \\
&\leq \int_{t_i}^{t} ||f\big(s, x^p(s), u_i^*(s)\big) - f\big(s, x_i^*(s), u_i^*(s)\big)||\, ds \\
&\qquad + \delta_i T := \Delta_i(t). \qquad (37)
\end{aligned}$$

The terminal cost m and the running cost l are Lipschitz continuous from Assumption 3, so we can use this relationship to find the expected cost discrepancy:

$$\begin{aligned}
&||J\big(x_i^p(t)\big) - J\big(x_i^*(t)\big)|| \\
&\leq \int_{t_i}^{t_i+T} ||l\big(s, x_i^p(s)\big) - l\big(s, x_i^*(s)\big)||\, ds \\
&\quad + ||m\big(t_i+T, x_i^p(t_i+T)\big) - m\big(t_i+T, x_i^*(t_i+T)\big)|| \\
&\leq \int_{t_i}^{t_i+T} L_1 \Delta_i(t_i+T)\, ds + L_2 \Delta_i(t_i+T) \\
&\leq \Delta_i(t_i+T)(L_1 T + L_2). \qquad (38)
\end{aligned}$$

with Lipschitz constants L_1 and L_2. Using (38) one can show that the upper bound on the contractive constraint (11) will inevitably be higher due to the effect of the disturbance. Specifically, from (11) and (38)

$$\begin{aligned}
J\big(x_i^*(\cdot)\big) - J\big(x_{i-1}^*(\cdot)\big) &\leq -\int_{t_{i-1}}^{t_i} l\big(t, x_{i-1}^*(t)\big)\, dt \\
&\quad + (L_1 T + L_2)[\Delta_i(t_i+T) + \Delta_{i-1}(t_{i-1}+T)]. \qquad (39)
\end{aligned}$$

This practically means that the cost is allowed to increase for as long as the disturbance is active.

The following Lemma is the last step before proving the theorem.

Lemma 3: As $t \to \infty$, $\Delta_i(t) \to 0$ in (37).

Proof: From Assumption 6, $\delta_i \to 0$ as $i \to \infty$. Also, using the fact that $x_i^*(t_i) = x^p(t_i)$ and Assumption 2, $x_i^p(t) \to x_i^*(t)$ as $i \to \infty$. ■

We can now follow the steps in the proof of Theorem 1 to show asymptotic stability under the disturbance $\eta(t)$. First, we can use Assumption 3, (39) and Lemma 3 to show that $\int_{t_0}^{t} M\big(x(s)\big)\, ds$ is bounded for $t \to \infty$ by following the same procedure as in Lemma 1. Asymptotic stability then follows directly from Lemma 2. The process is left as an exercise to the reader.

Theorem 3 is proved. ■

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under awards CNS-1426961 and DCSD-1662233. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

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

[3] A. R. Ansari and T. D. Murphey, "Sequential action control: Closed-form optimal control for nonlinear and nonsmooth systems," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1196–1214, 2016.

[4] A. Mavrommati, A. Ansari, and T. Murphey, "Optimal control-on-request: An application in real-time assistive balance control," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 5928–5934.

[5] E. Tzorakoleftherakis, A. Ansari, A. Wilson, J. Schultz, and T. D. Murphey, "Model-based reactive control for hybrid and high-dimensional robotic systems," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 431–438, 2016.

[6] T. Fan and T. Murphey, "Online feedback control for input-saturated robotic systems on Lie groups," in *Proceedings of Robotics: Science and Systems*, 2016.

[7] G. Mamakoukas, M. A. MacIver, and T. D. Murphey, "Sequential action control for models of underactuated underwater vehicles in a planar ideal fluid," in *American Control Conference (ACC), 2016*, 2016, pp. 4500–4506.

[8] M. Egerstedt, Y. Wardi, and H. Axelsson, "Optimal control of switching times in hybrid systems," in *IEEE International Conference on Methods and Models in Automation and Robotics*, 2003.

[9] ——, "Transition-time optimization for switched-mode dynamical systems," *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 110–115, 2006.

[10] L. S. Pontryagin, *Mathematical Theory of Optimal Processes*. CRC Press, 1987.

[11] K. J. Åström and K. Furuta, "Swinging up a pendulum by energy control," *Automatica*, vol. 36, no. 2, pp. 287–295, 2000.

[12] F. Lizarralde, J. T. Wen, and L. Hsu, "Feedback stabilization of nonlinear systems: a path space iteration approach," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 1997, pp. 4022–4023.

[13] M. da Silva, Y. Abe, and J. Popović, "Interactive simulation of stylized human locomotion," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, p. 82, 2008.

[14] Y. Tassa, T. Erez, and W. D. Smart, "Receding horizon differential dynamic programming," in *Advances in Neural Information Processing Systems*, 2008, pp. 1465–1472.

[15] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 4906–4913.

[16] H. Chen and F. Allgöwer, "Nonlinear model predictive control schemes with guaranteed stability," in *Nonlinear Model-based Process Control*. Springer, 1998, pp. 465–494.

[17] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. Springer, 2011.

[18] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[19] L. T. Biegler, "A survey on sensitivity-based nonlinear model predictive control," in *Proceedings of the IFAC International Symposium on Dynamics and Control of Process Systems*, 2013, pp. 499–510.

[20] H. Chen and F. Allgower, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," in *European Control Conference (ECC)*, 1997, pp. 1421–1426.

[21] F. A. Fontes, "A general framework to design stabilizing nonlinear model predictive controllers," *Systems & Control Letters*, vol. 42, no. 2, pp. 127–143, 2001.

[22] J. H. Lee, "Model predictive control: review of the three decades of development," *International Journal of Control, Automation and Systems*, vol. 9, no. 3, pp. 415–424, 2011.

[23] M. Hale, Y. Wardi, H. Jaleel, and M. Egerstedt, "Hamiltonian-based algorithm for optimal control," *arXiv preprint arXiv:1603.02747*, 2016.

[24] G. Mamakoukas, M. Maciver, and T. D. Murphey, "Feedback synthesis for underactuated systems using sequential second-order needle variations," *International Journal of Robotics Research*, 2018.

[25] T. M. Caldwell and T. D. Murphey, "Projection-based switched system optimization: Absolute continuity of the line search," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2012, pp. 699–706.

[26] S. L. de Oliveira Kothare and M. Morari, "Contractive model predictive control for constrained nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 6, pp. 1053–1071, 2000.

[27] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Systems*, vol. 22, no. 1, pp. 44–52, 2002.

[28] F. Xie and R. Fierro, "First-state contractive model predictive control of nonholonomic mobile robots," in *American Control Conference (ACC)*, 2008, pp. 3494–3499.

[29] G. Ferrari-Trecate, L. Galbusera, M. P. E. Marciandi, and R. Scattolini, "Model predictive control schemes for consensus in multi-agent systems with single-and double-integrator dynamics," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2560–2572, 2009.

[30] W.-H. Chen, J. O'Reilly, and D. J. Ballance, "On the terminal region of model predictive control for non-linear systems with input/state constraints," *International Journal of Adaptive Control and Signal Processing*, vol. 17, no. 3, pp. 195–207, 2003.

[31] S. Yu, H. Chen, C. Böhm, and F. Allgöwer, "Enlarging the terminal region of NMPC with parameter-dependent terminal control law," in *Nonlinear Model Predictive Control*. Springer, 2009, pp. 69–78.

[32] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Institute of Technology, 2000.

[33] P. O. Scokaert, D. Q. Mayne, and J. B. Rawlings, "Suboptimal model predictive control (feasibility implies stability)," *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 648–654, 1999.

[34] A. Jadbabaie, J. Yu, and J. Hauser, "Unconstrained receding-horizon control of nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 46, no. 5, pp. 776–783, 2001.

[35] F. ACC Fontes, "Discontinuous feedbacks, discontinuous optimal controls, and continuous-time model predictive control," *International Journal of Robust and Nonlinear Control*, vol. 13, no. 3-4, pp. 191–209, 2003.

[36] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 2520–2525.

[37] M. Karpenko, S. Bhatt, N. Bedrossian, A. Fleming, and I. Ross, "Flight implementation of pseudospectral optimal control for the TRACE space telescope," in *AIAA Guidance, Navigation, and Control Conference*, 2011, pp. 65–70.

[38] F. Fahroo and I. M. Ross, "Pseudospectral methods for infinite-horizon nonlinear optimal control problems," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 4, pp. 927–936, 2008.

[39] N. S. Bedrossian, S. Bhatt, W. Kang, and I. M. Ross, "Zero-propellant maneuver guidance," *IEEE Control Systems*, vol. 29, no. 5, 2009.

[40] C.-W. Tao, J.-S. Taur, T. W. Hsieh, and C. Tsai, "Design of a fuzzy controller with fuzzy swing-up and parallel distributed pole assignment schemes for an inverted pendulum and cart system," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1277–1288, 2008.

[41] E. Tzorakoleftherakis and T. D. Murphey, "Controllers as filters: Noise-driven swing-up control based on maxwell's demon," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2015, pp. 4368–4374.

[42] T. Albahkali, R. Mukherjee, and T. Das, "Swing-up control of the pendubot: an impulse–momentum approach," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 975–982, 2009.

[43] X. Xin and T. Yamasaki, "Energy-based swing-up control for a remotely driven acrobot: Theoretical and experimental results," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 4, pp. 1048–1056, 2012.

[44] X. Xin and M. Kaneda, "Analysis of the energy-based swing-up control of the acrobot," *International Journal of Robust and Nonlinear Control*, vol. 17, no. 16, pp. 1503–1524, 2007.

[45] R. W. Brockett *et al.*, "Asymptotic stability and feedback stabilization," *Differential Geometric Control Theory*, vol. 27, no. 1, pp. 181–191, 1983.

[46] C. Samson and K. Ait-Abderrahim, "Feedback control of a nonholonomic wheeled cart in cartesian space," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1991, pp. 1136–1141.

[47] C. C. De Wit and O. Sordalen, "Exponential stabilization of mobile robots with nonholonomic constraints," *IEEE Transactions on Automatic Control*, vol. 37, no. 11, pp. 1791–1797, 1992.

[48] A. Astolfi, "On the stabilization of nonholonomic systems," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 1994, pp. 3481–3486.

[49] C.-S. Chiu, K.-Y. Lian, and P. Liu, "Fuzzy gain scheduling for parallel parking a car-like robot," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 6, pp. 1084–1092, 2005.

[50] A. De Luca, G. Oriolo, and M. Vendittelli, "Stabilization of the unicycle via dynamic feedback linearization," in *IFAC International Symposium on Robot Control*, 2000, pp. 397–402.

[51] D. Gu and H. Hu, "A stabilizing receding horizon regulator for nonholonomic mobile robots," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 1022–1028, 2005.

[52] J. Hauser, S. Sastry, and G. Meyer, "Nonlinear control design for slightly non-minimum phase systems: application to v/stol aircraft," *Automatica*, vol. 28, no. 4, pp. 665–679, 1992.

[53] S.-L. Wu, P.-C. Chen, C.-H. Hsu, and K.-Y. Chang, "Gain-scheduled control of pvtol aircraft dynamics with parameter-dependent disturbance," *Journal of the Franklin Institute*, vol. 345, no. 8, pp. 906–925, 2008.

[54] F. Lin, W. Zhang, and R. D. Brandt, "Robust hovering control of a pvtol aircraft," *IEEE Transactions on Control Systems Technology*, vol. 7, no. 3, pp. 343–351, 1999.

[55] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 2247–2252.

[56] A. E. C. D. Cunha, "Benchmark: Quadrotor attitude control," in *Proceedings of the International Workshop on Applied Verification for Continuous and Hybrid Systems (ARCH)*, vol. 34, 2015, pp. 57–72.

[57] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a large quadrotor robot," *Control Engineering Practice*, vol. 18, no. 7, pp. 691–699, 2010.

[58] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2004, pp. 2451–2456.

[59] E. Fresk and G. Nikolakopoulos, "Full quaternion based attitude control for a quadrotor," in *European Control Conference (ECC)*, 2013, pp. 3864–3869.

[60] D. Lee, H. J. Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of Control, Automation and Systems*, vol. 7, no. 3, pp. 419–428, 2009.

[61] M. Karpenko, S. Bhatt, N. Bedrossian, A. Fleming, and I. Ross, "First flight results on time-optimal spacecraft slews," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 367–376, 2012.

[62] F. H. Clarke, Y. S. Ledyaev, E. D. Sontag, and A. I. Subbotin, "Asymptotic controllability implies feedback stabilization," *IEEE Transactions on Automatic Control*, vol. 42, no. 10, pp. 1394–1407, 1997.

[63] H. Michalska and R. Vinter, "Nonlinear stabilization using discontinuous moving-horizon control," *IMA Journal of Mathematical Control and Information*, vol. 11, no. 4, pp. 321–340, 1994.

[64] I. Barbalat, "Systemes déquations différentielles doscillations non linéaires," *Rev. Math. Pures Appl.*, vol. 4, no. 2, pp. 267–270, 1959.

Emmanouil Tzorakoleftherakis received the joint B.S./M.S. degree in electrical and computer engineering from University of Patras, Greece in 2012, and the M.S. and Ph.D. degrees in mechanical engineering from Northwestern University, Evanston, IL, USA, in 2015 and 2017 respectively. His interests include control and robotics.

Todd D. Murphey received the B.S. degree in mathematics from University of Arizona, Tucson, AZ, USA, and the Ph.D. degree in control and dynamical systems from California Institute of Technology, Pasadena, CA, USA.

He is an Associate Professor of Mechanical Engineering with Northwestern University Evanston, IL, USA. His laboratory is part of the Neuroscience and Robotics Laboratory, and his research interests include robotics, control, and computational methods for biomechanical systems and neuroscience. He has received honors including 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.