Superlinear Convergence Using Controls Based on Second-Order Needle Variations

Giorgos Mamakoukas Malcolm A. MacIver Todd D. Murphey

Abstract— **This paper investigates the convergence performance of second-order needle variation methods for nonlinear control-affine systems. Control solutions have a closed-form expression that is derived from the first- and second-order mode insertion gradients of the objective and are proven to exhibit superlinear convergence near equilibrium. Compared to first-order needle variations, the proposed synthesis scheme exhibits superior convergence at smaller computational cost than alternative nonlinear feedback controllers. Simulation results on the differential drive model verify the analysis and show that second-order needle variations outperform first-order variational methods and iLQR near the optimizer. Last, even when implemented in a closed-loop, receding horizon setting, the proposed algorithm demonstrates superior convergence against the iterative linear quadratic Gaussian (iLQG) controller.**

I. INTRODUCTION

The challenges of nonlinear optimization pose great demands on candidate controllers for general nonlinear systems. Feedback schemes need to perform well throughout the state-space, exhibit robustness to the initial error and actuator limits, and maintain reasonable computational efficiency so that they can be implemented in an online fashion. To address these issues, various methods have been suggested in recent years, including, among others, feedback linearization and backstepping [1], [2], dynamic programming (DP) [3], iterative linear quadratic regulators (iLQR) [4], linear and nonlinear model predictive control (LMPC / NMPC) [5], [6]. However, these methods are either optimal only near the nominal trajectory (LQR, LMPC), fail in the presence of actuator limits (feedback linearization and backstepping), or are computationally rather expensive (DP, NMPC).

A. Needle Variation Methods in Optimal Control

Feedback schemes based on needle variation actions [7], [8], on the other hand, do well in these areas. Because they do not try to minimize the objective, but rather synthesize individual actions that best reduce the cost function with respect to its time evolution, they perform well regardless of the initial error, are computationally fast, and readily apply saturation controls. Additionally, they show robustness to model parameter uncertainties and, when compared to the alternative schemes, they are less sensitive to local minima and have a less complicated representation on Lie groups. Last, they have provable guarantees of descent for certain controllable systems, when considering second-order information. These traits of feedback policies based on needle variation actions are demonstrated on various benchmark problems, such as the cart-pendulum, acrobot, and pendubot [8]–[14].

Needle variation actions in optimal control have so far primarily focused only on the first-order sensitivity of the cost with respect to an infinitesimal (in duration) perturbation in the nominal control. By doing so, they ignore the local curvature of the cost function. As a result, they behave like gradient-descent algorithms and are subject to first-order convergence rates, especially near the equilibrium [15]–[17].

As this paper shows, second-order needle variations improve convergence. By virtue of exploiting curvature information of the approximated cost function, they can be used to compute controls that reduce the cost to first *and* second order. As a result, they improve convergence for applications of finite (but small) duration, regardless of whether first-order methods are non-singular.[1] Moreover, second-order needle variation controls remain computationally less expensive than traditional nonlinear optimization methods.

B. Paper Contribution and Structure

This paper proposes a feedback algorithm for general nonlinear systems that complements needle variation methods with the second order mode insertion gradient in order to improve convergence, while maintaining the aforementioned benefits of needle variation methods in optimal nonlinear control theory. We show that second-order needle variation actions demonstrate superlinear convergence rates, and verify the analysis using the differential drive system. Simulated examples using the cart pendulum further suggest superior convergence rates compared to first-order methods in regions of the state-space that lie away from the optimizer.

The structure of the paper is as follows. In Section II, we discuss the complexity of second-order needle variation controls to alternative nonlinear feedback schemes. In Section III, we analytically show the superlinear convergence rate of second-order needle variation actions and present a feedback synthesis method that is used to compare first- and second-order needle variation actions. In Section IV, we demonstrate

The authors are with the Department of Mechanical Engineering (Mamakoukas, MacIver and Murphey), with the Department of Biomedical Engineering (MacIver), and the Department of Neurobiology (MacIver), Northwestern University, Evanston, IL, 60208 USA. Emails: giorgosmamakoukas@u.northwestern.edu, maciver@northwestern.edu, t-murphey@northwestern.edu

[1] The local shape of the cost function is meant with respect to the duration of needle perturbations in the controls at a certain application time.

convergence, both near and away from the equilibrium, using simulation examples on the 2D differential drive and the cart pendulum system.

II. Comparison to alternative schemes

In comparison to alternative feedback controllers, needle variation methods do not attempt to minimize the objective by constructing a value, cost-to-go, function and computing control sequences throughout a time horizon. Rather, they exploit the time-evolving sensitivity of the objective to infinitesimal switched dynamics in order to compute a single needle perturbation to the nominal control that will optimally improve the cost.

Work in [8] shows that control solutions have a closed-form expression and avoid the computationally expensive iterative procedure of alternative schemes. Further, needle variation actions are shown to exist globally, demonstrate a larger region of attraction and have a less complicated representation on Lie groups [11]. In addition, the suggested second-order needle variation controller has formal guarantees of descent for systems that are controllable with first-order Lie brackets, thus succeeding at tasks (e.g. parallel parking) when gradient descent methods fail [9]. These traits make them suitable for time-sensitive robotic applications that may be subject to large initial error, Euler angle singularities, or fast-evolving (and uncertain) objectives.

In terms of computational effort, second-order needle variation methods are less expensive than existing feedback controllers, such as iLQR or projection-based trajectory optimization methods. For example, in each iteration, the iLQR algorithm solves the Riccati equations to calculate the descent direction and, together with the simulation of the state, it computes three $n \times 1$ and one $n \times n$ differential equations. Projection-based optimization methods solve an additional $n \times n$ set of differential equations for the projection operation onto feasible dynamics [18]. The proposed second-order needle variation approach simulates the state ($n \times 1$), the first- ($n \times 1$), and the second-order ($n \times n$) adjoint to calculate controls and thus comes at lower computational cost to these methods [19]. More importantly, even one iterate of this method is often much closer to the optimizer.

The proposed controller combines the benefits of needle variation methods over alternative feedback schemes, improves convergence over its first-order counterpart, and provides formal guarantees of actionability for nonlinear systems. As a result, it is a promising alternative that could complement existing nonlinear control methods.

III. Second order needle variation actions

Consider a system with state $x : \mathbb{R} \mapsto \mathbb{R}^{N\times 1}$ and control $u : \mathbb{R} \mapsto \mathbb{R}^{M\times 1}$ with control-affine dynamics of the form

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

where $g(x(t),t)$ is the drift vector field. Further consider a time horizon T that describes an interval $[t_o, t_f]$, where $t_f = t_o + T$. Given default control $v(t)$, a needle-like single-action control is inserted at an application time τ for duration λ, such that

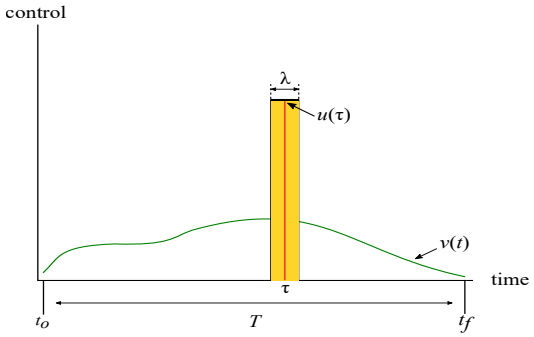

Fig. 1: While the default control is allowed to vary with respect to time, the inserted single-action control is fixed.

$$u_{v,\tau,\lambda}(t) = \begin{cases} v(t), & t \notin [\tau, \tau+\lambda] \\ u(\tau), & t \in [\tau, \tau+\lambda], \end{cases}$$

giving rise to dynamics described by

$$\dot{x}(t) = \begin{cases} f_1, & t_0 \le t < \tau \\ f_2, & \tau \le t < \tau+\lambda \\ f_1, & \tau+\lambda \le t \le t_f, \end{cases} \tag{2}$$

where f_1 and f_2 are associated with default and inserted control v and u, respectively:

$$f_1 \triangleq f(x(t),v(t),t) \qquad \text{and} \qquad f_2 \triangleq f(x(t),u(\tau),t).$$

Note that dynamics f_2 consider fixed control $u(\tau)$, whereas default dynamics f_1 allow for time-varying input $v(t)$.

Feedback controllers based on needle variation methods typically consider objectives without a control cost term [8], [11]. In this work, we derive for the first time feedback policies for cost functions with a control term, of the form

$$J(x(t),u(t)) = \int_{t_o}^{t_f} l(x(t),u(t))\,\mathrm{d}t + m(x(t_f)), \tag{3}$$

where $l(x(t),u(t))$ is the running cost and $m(x(t))$ the terminal cost. The optimization problems used in this paper rest on a series of assumptions that ensure all variables are well-defined.

Assumption 1. *The elements of vector dynamics f_1 and f_2 are real, bounded, $\mathscr{C}^2$ in x, and $\mathscr{C}^0$ in u and t.*

Assumption 2. *The incremental and terminal cost terms—$l(x,u)$ and $m(x)$, respectively—are real and $\mathscr{C}^2$ in x.*

Assumption 3. *Default and inserted controls v and u are real, bounded, and $\mathscr{C}^0$ in t.*

Throughout the rest of this work, we consider a quadratic cost function that satisfies these assumptions with

$$l(x(t),u(t)) = \frac{1}{2}\|\vec{x}(t) - \vec{x}_d(t)\|_Q^2 + \frac{1}{2}\|u(t)\|_R^2$$
$$m(x(t_f)) = \frac{1}{2}\|\vec{x}(t_f) - \vec{x}_d(t_f)\|_{P_1}^2,$$

where Q and P_1 are weight matrices for the running and terminal state errors, respectively, and R the metric on control effort. For such cost functions, the mode insertion gradient (MIG) and Hessian (MIH) are given by

$$\frac{dJ}{d\lambda_+} = \rho^T(f_2 - f_1) + \frac{1}{2}\|u(t)\|_R^2 - \frac{1}{2}\|v(t)\|_R^2, \tag{4}$$

$$\frac{d^2J}{d\lambda_+^2} = (f_2 - f_1)^T \Omega (f_2 - f_1) + \rho^T (D_x f_2 \cdot f_2 + D_x f_1 \cdot f_1 - 2D_x f_1 \cdot f_2) - D_x l \cdot (f_2 - f_1), \tag{5}$$

where $\rho : \mathbb{R} \mapsto \mathbb{R}^{N\times 1}$ and $\Omega : \mathbb{R} \mapsto \mathbb{R}^{N\times N}$ are the first- and second-order adjoints (co-states). We use λ_+ to indicate that a certain term is considered after taking the limit $\lambda \to 0^+$. For the derivation of (4) and (5), the reader can refer to the Appendix. These quantities are calculated using the default dynamics f_1 and are given by

$$\dot{\rho} = -D_x l^T - D_x f_1^T \rho$$

$$\dot{\Omega} = -D_x {f_1}^T \Omega - \Omega D_x f_1 - D_x^2 l - \sum_{i=1}^{N} \rho_i D_x^2 f_1^i,$$

that are subject to

$$\rho(t_f) = D_x m(x(t_f))^T \quad \text{and} \quad \Omega(t_f) = D_x^2 m(x(t_f))^T.$$

The superscript i in the dynamics f_1 refers to the i^{th} element of the vector and is used to avoid confusion against default and inserted dynamics f_1 and f_2, respectively. The operator D_x denotes a partial derivative with respect to x. The MIH expression is derived by the authors. Due to space constraints, the derivation is not included in this paper, but similar analysis can be found in [20].

Note that the MIG and MIH terms have, to the best of our knowledge, only been calculated with respect to a cost function that does not include a metric on control effort [21]–[23]. Expressions (4) and (5) have been modified to correspond to the cost function shown in (3).

A. Convergence Analysis

Given dynamics in (2), the cost function (3) depends on the duration λ of dynamics f_2. The sensitivity of the objective to infinitesimal switched modes typically considers the mode insertion gradient to locally approximate the variation of J as a function of λ. This relationship explicitly depends on the control u and reveals how different dynamic modes locally affect the cost function. Work in [8] and [24] shows that the local structure of the cost with respect to perturbations of the nominal control (and, by extension, state, co-state, and trajectory) can be approximated with a Taylor expansion around the duration of inserted control to first-order as

$$J(u,\lambda) \approx J_o + \left.\frac{dJ}{d\lambda}\right|_{\lim_{\lambda\to 0^+}} \lambda, \tag{6}$$

where $J_o \triangleq J(v(t))$ is the default cost that corresponds to no inserted control ($\lambda = 0$). The mode insertion gradient term indicates the sensitivity of the cost at an application time τ and changes as a function of time. As a result, the local structure of the objective around switched mode dynamics also evolves with time (see Fig. 2).

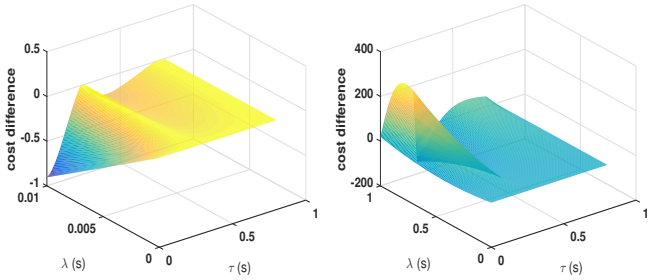

Fig. 2: Approximated structure of the local sensitivity of the objective on the application of switched mode dynamics of duration λ. The structure is constructed using (7) (without J_o for easier visualization) with controls from (8). As the values of the MIG and MIH change with respect to time, so will the local approximation shape. The two figures are plotted over different ranges of λ. They aim to illustrate how λ acts as the trade-off factor between first- and second-order information in a way that can dramatically change control solutions. The figures serve as an instructive example and were created on the cart pendulum system.

With the MIH expression, we can better approximate the local structure of the cost function (3) around the duration of switched dynamics by including curvature information in the control synthesis procedure. The second-order approximation of the sensitivity of the cost function (3) to needle perturbations in nominal control enables us to compute solutions that consider a more accurate approximation of the cost function. Considering how the objective changes as a function of λ to higher order, control solutions are likely to remain optimal for longer, non-infinitesimal durations of control that lie further away from the reference point ($\lambda = 0$).

Next, we use the first- and second-order mode insertion gradients to model the sensitivity of the objective to needle perturbations in control. As one would expect, the structure depends on the needle control u, its duration λ, and the application time τ. We express this relationship as

$$\tilde{J}(\lambda,u) \triangleq J(u_{v,\tau,\lambda}(t)) \approx J_o + \frac{dJ}{d\lambda_+}\lambda + \frac{d^2J}{d\lambda_+^2}\frac{\lambda^2}{2}, \tag{7}$$

where λ_+ is used instead of λ to indicate that the associated terms are considered after taking the limit $\lambda \to 0^+$. That is,

$$\frac{dJ}{d\lambda_+} \triangleq \left.\frac{dJ}{d\lambda}\right|_{\lim_{\lambda\to 0^+}} \quad \text{and} \quad \frac{d^2J}{d\lambda_+^2} \triangleq \left.\frac{d^2J}{d\lambda^2}\right|_{\lim_{\lambda\to 0^+}}.$$

Considering arbitrary control duration $\lambda = \lambda_o$, and using the MIG and MIH, the above expression can be written as a function only of the control u and the application time τ. For simplicity purposes, the time dependence is omitted. Then, consider the following update step on the approximated function with respect to the control u

$$u^{k+1} = u^k - \left(\nabla_u^2 \tilde{J}(u)\right)^{-1} \nabla_u \tilde{J}(u)\Big|_{u=u_k} \tag{8}$$

where $\tilde{J}$ is the second-order approximation of J seen in (7) with $\nabla_u \tilde{J}(u) \in \mathbb{R} \mapsto \mathbb{R}^{M\times 1}$ and $\nabla_u^2 \tilde{J}(u) \in \mathbb{R} \mapsto \mathbb{R}^{M\times M}$. Each

control u^{k+1} is considered part of the default control v in the following iteration. This update equation is similar to, but not the same as, a Newton step on the cost function J. Rather, we consider a different differentiation operator where we first differentiate the cost function with respect to λ and then with respect to the control u. By means of the chain rule, this operation allows us to locally approximate the dependency of the cost function on controls u.

Let $h_i : \mathbb{R} \mapsto \mathbb{R}^{N\times 1}$ be the column control vectors that compose $h : \mathbb{R} \mapsto \mathbb{R}^{N\times M}$ in (1) and $u_i \in \mathbb{R}$ be the individual control inputs. Given (4), (5), and (7), the control update policy $u^k - u^{k+1}$ has a closed-form solution (8) given by

$$\begin{aligned} &= \Big(\frac{\lambda_o}{2}\big(h^T(\Omega^T+\Omega)h + h^T\big(\sum_{k=1}^{n}(D_x h_k)\rho_k\big)^T + \big(\sum_{k=1}^{n}(D_x h_k)\rho_k\big)\cdot h\big) + R\Big)^{-1} \\ &\quad \cdot \Big(h^T\rho + Ru^k + \frac{\lambda_o}{2}\big(\mathscr{L}_{h,h}\cdot u^k + \mathscr{L}_{g,h} - h^T D_x l^T\big)\Big), \end{aligned} \tag{9}$$

where

$$\mathscr{L}_{g,h} \triangleq \begin{pmatrix} \rho^T[g,h_1] \\ \vdots \\ \rho^T[g,h_M] \end{pmatrix}, \mathscr{L}_{h,h} \triangleq \begin{pmatrix} \rho^T[h_1,h_1] & \dots & \rho^T[h_M,h_1] \\ \vdots & \ddots & \vdots \\ \rho^T[h_1,h_M] & \dots & \rho^T[h_M,h_M] \end{pmatrix}.$$

The Lie bracket terms guarantee that a descent direction exists, even when first-order needle variation methods fail (i.e. $h^T\rho = 0$). Given this setup, and following the analysis in [3], we can say that, in a neighborhood around the equilibrium where $\nabla_u^2(\tilde{J}) \succ 0$, the sequence $\{u^k\}$ generated by the iteration (9) converges to u^*. In addition, $\{\|u^k - u^*\|\}$ converges superlinearly.

Proposition 1. *Given application time τ, consider a function $\nabla_u\tilde{J}(u(t)) \in \mathbb{R} \mapsto \mathbb{R}^M$, and a vector $u^*(\tau)$ such that $\nabla_u\tilde{J}(u^*(\tau)) = 0$ and $\nabla_u^2\tilde{J}(u^*(\tau)) \succ 0$. For $\delta > 0$, let S_δ denote the sphere $\{u(\tau)|\|u(\tau) - u^*(\tau)\| \le \delta\}$. Assume continuity of ρ, g, h, v, and Ω in a neighborhood of $\tau : [\tau, \tau+\lambda]$ and that $\nabla_u^2\tilde{J}(u^*(\tau))$ is invertible. Then, there exists $\delta > 0$ such that if $u^0(\tau) \in S_\delta$, the sequence $\{u^k(\tau)\}$ generated by the iteration*

$$u^{k+1}(\tau) = u^k(\tau) - \Big(\nabla_u^2\tilde{J}(u^k(\tau))\Big)^{-1}\nabla_u\tilde{J}(u^k(\tau))$$

is defined, belongs to S_δ, and converges to $u^*(\tau)$. In addition, $\{\|u^k(\tau) - u^*(\tau)\|\}$ converges superlinearly.

Proof. Let $d(u(\tau)) = \nabla_u\tilde{J}(u(\tau))$, such that $\nabla d(u(\tau)) = \nabla_u^2\tilde{J}(u(\tau))$. The proof follows the analysis by Bertsekas [3] and rests on the assumptions that a) $d(u(\tau))$ is continuously differentiable (such that $\nabla d(u(\tau))$ exists) and that b) $\nabla d(u^*(\tau))$ is invertible. Under continuity assumptions of ρ, g, h, and v that are discussed in [8], $d(u(\tau))$ exists in a neighborhood of τ: $[\tau, \tau+\lambda]$. Moreover, under continuity assumptions of Ω, $\nabla d(u(\tau))$ exists in a neighborhood of τ: $[\tau, \tau+\lambda]$, which meets assumption a. Further, $\nabla d(u(\tau))$, for continuity purposes, remains invertible for some $\delta > 0$ (where $\|u(\tau) - u^*(\tau)\| \le \delta$) and also near a small neighborhood of τ: $[\tau, \tau+\lambda]$, which meets assumption b. Therefore, the sequence $\{u^k(\tau)\}$, starting at $u^0(\tau) \in S_\delta$, converges to $u^*(\tau)$ superlinearly. □

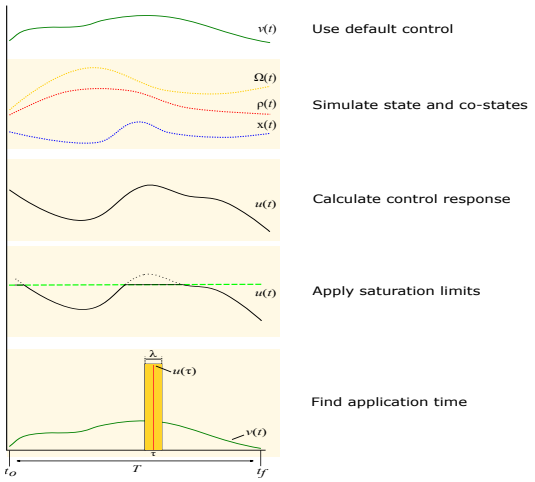

Fig. 3: The algorithmic steps of the single-action controller, as presented in Algorithm 1.

We should stress that the update policy in (8) is meant with respect to a single application of inserted control u at a specific application time. Contrary to optimization schemes that iterate on the entire curve of actions to minimize the objective, the proposed update sequence allows one to converge to $u^*(\tau)$, for which the local quadratic approximation of J with respect to needle perturbations in control is at a minimum for a given duration λ_o. In practice, the application time between iterations may vary, so that different sequences of controls (e.g. $\{u(\tau_1)\} \neq u(\tau_2)\}$) will be updated. This implies that the evolution of the cost function may not demonstrate overall superlinear convergence. Rather, in each iteration i, controls will converge superlinearly to their optimal value when examined separately at application time τ_i. For receding horizon tasks, this limitation becomes of lesser importance, since the objective continuously changes and the convergence rate of each individual action is what matters. When compared to first-order needle variation actions iteration by iteration, convergence of second-order actions will be superior.

In addition, the convergence rate assumes no saturation limits on the control inputs. That said, near the equilibrium control updates are expected to be small and unlikely to saturate. As a result, the superlinear convergence rate, exhibited primarily near a neighborhood of the equilibrium, is not expected to deteriorate considerably in the presence of actuation constraints.

B. Algorithmic Procedure

First- and second-order needle variation actions are compared using the same algorithmic procedure, that is outlined in Algorithm 1 and illustrated in Fig. 3.

For first-order actions, simulation is required only for the states and the first-order co-states. Computed controls minimize the first-order Taylor expansion of the objective (6) and have a control update policy given by

$$u^{k+1} = u^k - R^{-1}(h^T\rho + Ru^k). \tag{10}$$

Algorithm 1

1: Simulate states and co-state $[t_o, t_o + T]$ with default control $v(t)$
2: Compute controls $u(t)$ that minimize $\tilde{J}(u(t))$
3: Saturate controls $u(t)$
4: Determine application time τ
5: Determine update step to decrease the cost function (3)
6: Store actions in $[\tau - \lambda/2, \tau + \lambda/2]$

The application time is found by minimizing (6). Second-order actions need to additionally compute the second-order co-states in Step 1 of Algorithm 1, while control solutions and the application time minimize (7). In the last step, the update is determined with an Armijo line search [25].

For the examples that are near the equilibrium, the line search was performed with respect to the magnitude of the control update step $u^{k+1} = u^k - \gamma d^k$, where d^k is the descent direction. For all other examples, we implemented a line search with respect to the duration λ of the control $[\tau - \gamma\lambda/2, \tau + \gamma\lambda/2]$ so as to take advantage of possible larger time steps. We should note that the MIG and the MIH could be used to find the optimal control duration and skip the line search in each iteration. For the purposes of a more accurate comparison between solutions of first- and second-order methods, we do not exploit this option in this paper. In Section IV, Algorithm 1 refers to first-order actions and Algorithm 2 to second-order ones.

IV. Simulation Results

We demonstrate the convergence performance of second-order needle variation methods in various settings. Using the example of a differential drive and the cart-pendulum system, we examine convergence rates of second-order needle variation actions near equilibrium. We also illustrate convergence performance for a receding-horizon task of inverting the cart pendulum. In the receding horizon examples, the application time is re-calculated in each iteration, and the line search is with respect to the duration in order to achieve as large of a cost reduction as possible.

A. Convergence near equilibrium

1) Differential Drive: The system has states $s = [x, y, \theta]^T$ and dynamics

$$f = r \begin{bmatrix} cos(\theta) & cos(\theta) \\ sin(\theta) & sin(\theta) \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} u_R \\ u_L \end{bmatrix}, \tag{11}$$

where $r = 3.6$ cm is the radius of each wheel, $L = 25.8$ cm is the distance between them, and u_R, u_L are the right and left wheel control angular velocities, respectively. Convergence near equilibrium was tested with iLQR, first- and second-order needle variation methods and the results are presented in Fig. 4 and Fig. 5. Initial conditions are $s_o = [0,0,0]^T$, desired state $s_d = [10 \text{ mm}, 0, 0]^T$, and parameters used are $Q = \text{diag}(10,10,10)$, $P_1 = \text{diag}(0,0,0)$, $R = \text{diag}(10^{-4}, 10^{-4})$, $T = 2.5$ s, and $\lambda_o = 10^{-2}$.

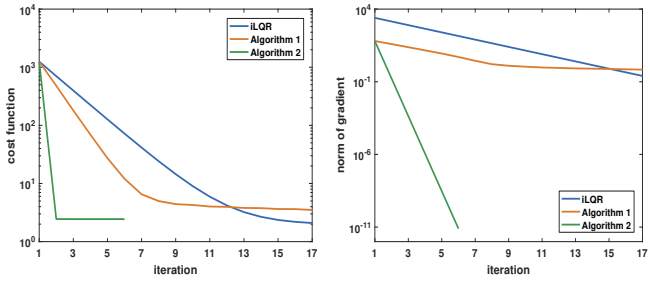

Fig. 4: Convergence of iLQR, first- and second-order needle variation methods near equilibrium. Algorithm 2 shows improved (superlinear) convergence compared to iLQR and Algorithm 1, which demonstrate linear and sublinear rates, respectively.

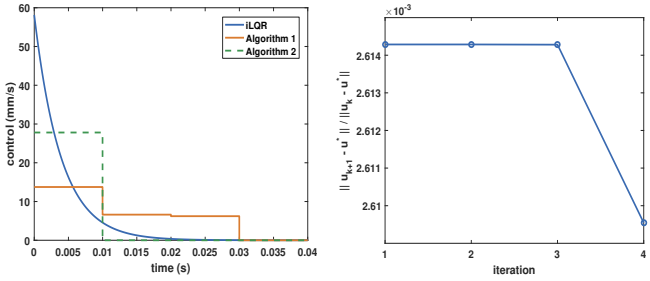

Fig. 5: The left figure shows the control solutions of the three methods after six iterations. The right plot shows the convergence rate for the sequence of controls generated with Algorithm 2, which converges after five iterations. For u^*, we use the final value to which the algorithm converges. The application time of each action happens to be the same $(t = 0)$ so that convergence rates of the sequence $\{u(t = 0)\}$ verify the superlinear rate predicted in the analysis.

It is important to note that, for an accurate measure of the convergence rate of the needle variation based algorithms, control actions are applied for duration λ_o as is used in the calculation of solutions (8) and (10). In addition, the time horizon of the cost $[t_o, t_f]$ was kept fixed. In this simulation, the application time was the same $(t = 0)$ in each iteration for Algorithm 2. This fact allows the results to be consistent with the convergence analysis, in the sense that the generated control sequence is with respect to the same application time. Algorithm 2 converges considerably faster than the other schemes. Without any penalty on control effort $(R = 0)$, Algorithm 2 converges in one iteration, whereas the other two methods fail due to infinite control. The cost function given these parameters is convex with respect to the inserted control, which explains the result.

A Monte Carlo simulation that investigates convergence rates near equilibrium is presented in Fig. 6 for 100 trials. To create a convex optimization problem, we sampled only over the x-coordinate of the system: $s_o = [U(-100, 100), 0, 0]^T$. The desired state used is $s_d = [0,0,0]^T$ and we excluded trials with initial coordinates that lied within 10 mm from the target. Simulation parameters were $Q = \text{diag}(0,0,0)$, $P_1 = \text{diag}(10,10,10)$, $R = \text{diag}(10^{-4}, 10^{-4})$, $T = 2.5$ s, and $\lambda_o = 10^{-2}$. All samples converged within five iterations, fewer than what iLQR or first-order methods needed given

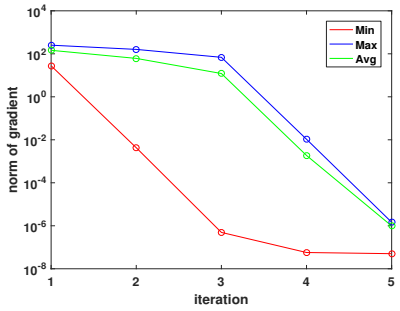

Fig. 6: Monte Carlo simulation (100 trials) for the differential drive. Fast convergence rates occur once the norm of the gradient falls approximately below 100. The stopping criterion for convergence is that the gradient norm is no larger than 10^{-5}. The calculation of the average information for a given iteration is collected only from the samples that have not converged for that iteration.

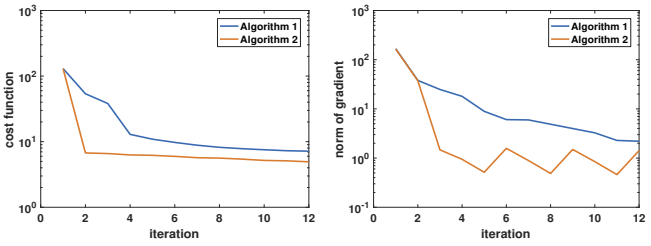

Fig. 7: Convergence of first- and second-order needle variation methods near equilibrium for a fixed-horizon inversion task of the cart pendulum system. Convergence is not consistently quadratic because the application time changes between iterations.

a smaller initial error (see Fig. 4).

2) Cart pendulum: We consider the cart pendulum system because it has been a popular testbed for conventional controllers (e.g. PID, LQR, DDP) [26]–[29]. The system states are $s = [\theta, \dot{\theta}, x, \dot{x}]^T$ and has dynamics

$$f = \begin{bmatrix} \dot{\theta} \\ \frac{g\sin(\theta)}{l} - \frac{b\dot{\theta}}{ml^2} \\ \dot{x} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{cos(\theta)}{l} \\ 0 \\ 1 \end{bmatrix} u_1, \qquad (12)$$

where the parameters $m = 0.2$ kg, $l = 1$ m, $b = 0.01$, $g = 9.81$ m/s^2 are the mass, pendulum length, friction coefficient, and gravity constant, respectively. Near-equilibrium convergence was tested with initial conditions $s_o = [\pi/36, 0, 0, 0]^T$, desired states given by $s_d = [0,0,0,0]^T$, and rest of parameters described by $T = 0.5$ s, $R = 10^{-4}$, $Q = \text{diag}(0,0,0,0)$, $P_1 = \text{diag}(10^4, 10^4, 0, 0)$, $\lambda_o = 0.01$ s, and saturation limits $u_1 \in [-20, +20]$ m/s^2. Results are presented in Fig. 7. It is worth noting that the norm of the gradient for the first two steps are similar between the two algorithms, however, second-order needle variation controls manage a significantly larger cost reduction. This observation exemplifies how the MIH helps appropriately scale control inputs.

B. Receding horizon convergence

We compare the convergence of first- and second-order needle variation methods in a closed-loop, receding horizon

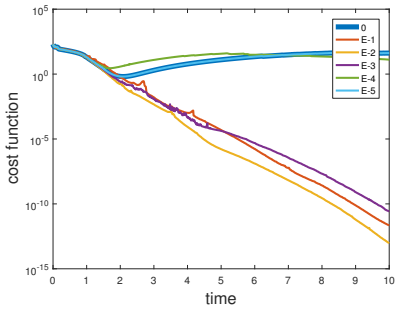

Fig. 8: Cost function with first- and second-order needle variation actions for a receding-horizon pendulum inversion task. Simulation was run for ten seconds at a sampling rate of $t_s = 0.01$ s. The cost function is plotted for different values of λ. As is evident, weighing second-order information too much or too little can be disadvantageous.

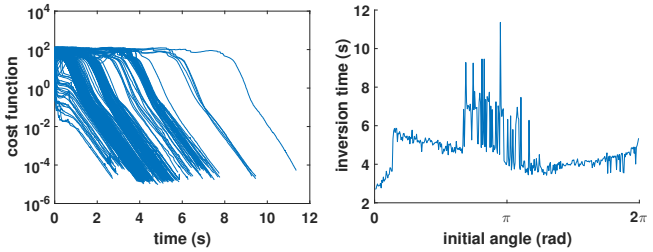

Fig. 9: Convergence of the cart pendulum inversion task over different initial angles using second-order needle variations.

setting using the dynamics of the cart pendulum system (12) [30], [31]. Second-order methods converge in half the time than their first-order counterparts, as shown in Fig. 10 and Fig. 8. Results were generated with $s_o = [\pi, 0, 0, 0]^T$, $s_d = [0,0,0,0]^T$, $T = 1.2$ s, $R = 0.3$, $Q = \text{diag}(20,0,5,0)$, $P_1 = \text{diag}(0.1,0,5,0)$, sampling rate $t_s = 0.01$ s, $\lambda_o = 10^{-3}$ s and $u_1 \in [-20, +20]$ m/s^2. As the screenshots reveal, second-order methods need about half the time to complete the full inversion than their first-order counterparts (see Fig. 8).

We further test convergence of solutions in (9) over different initial conditions (θ_o) and present the results in Fig. 9. Convergence is assumed when each state is less than 0.01. Angles were sampled at increments of 1 degree from 0 to 2π and the rest of simulation parameters are $s_d = [0,0,0,0]^T$, $T = 1.4$ s, $R = 0.1$, $Q = \text{diag}(20,0,5,0)$, $P_1 = \text{diag}(0.1,0,5,0)$, sampling rate $t_s = 0.01$ s, $\lambda_o = 10^{-2}$ s and $u_1 \in [-20, +20]$ m/s^2. All trials succeed within 12 seconds. The symmetry of the inversion time around $\theta_o = \pi$ is expected, with simulations starting closer to the inverted position converging faster. Certain trials are significantly slower, however all runs exhibit similar convergence rate once they are sufficiently close to the equilibrium.

We also compare the convergence of the needle variation methods and iLQG [32] on the differential drive system, using the publicly available software.[2] As Fig. 11 and 12 indicate, second-order needle variation solutions converge faster

[2] Available at http://www.mathworks.com/matlabcentral/fileexchange/52069-ilqg-ddp-trajectory-optimization

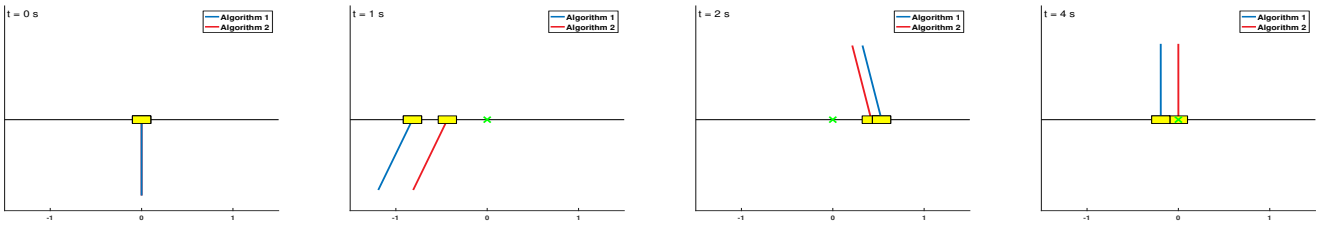

Fig. 10: Snapshots of receding-horizon pendulum inversion with first- and second-order needle variation actions.

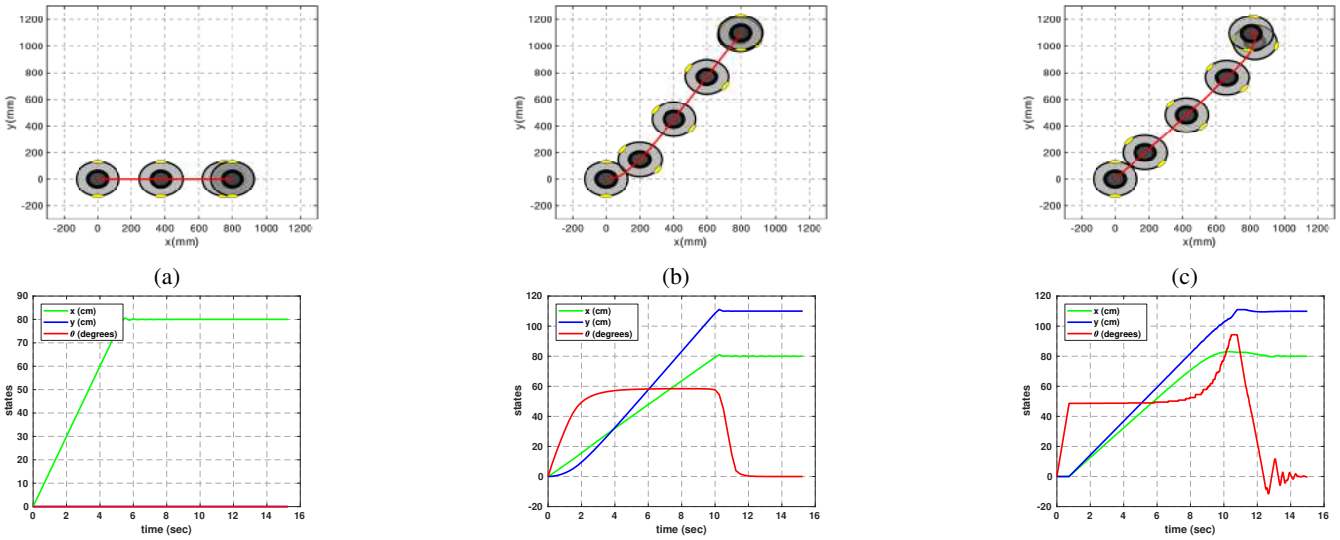

Fig. 11: Differential drive using first- (left), second-order (middle) needle variation actions, and iLQG. Snapshots of the system are shown at $t = 0, 2.5, 5, 7.5, 10$, and 12.5 sec. The target state is $[x_d, y_d, \theta_d] = [800 \text{ mm}, 1100 \text{ mm}, 0]$.

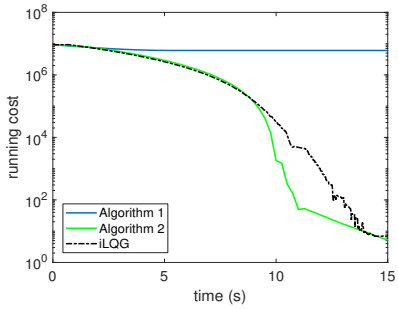

Fig. 12: Comparison of first-, second- order needle variation methods, and iLQG on the differential drive system. The needle variation methods are implemented in a receding horizon setting and compared against the off-line trajectory optimization solution generated by iLQG.

than the off-line solution provided by iLQG. First-order needle variation methods fail to turn the vehicle to the target. This simulation uses initial conditions $s_o = [0,0,0]^T$, desired target $s_d = [800 \text{ mm}, 1100 \text{ mm}, 0]^T$, $Q = \text{diag}(10,10,1000)$, $P_1 = \text{diag}(0,0,0), R = \text{diag}(0.5,0.5), T = 0.5$ s, $\lambda_o = 10^{-1}$.

V. CONCLUSIONS

Including the mode insertion Hessian in the computation of needle variation actions allows us to achieve superlinear convergence with respect to the sequence of actions at a certain application time. Even when applications times are allowed to vary to minimize the objective, the proposed algorithm has improved convergence near and away from the equilibrium, in fixed- and receding-horizon tasks, compared to first-order needle variation methods.

VI. ACKNOWLEDGEMENT

This work was supported by the National Science Foundation under Grant CMMI 1662233. Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the National Science Foundation.

VII. APPENDIX

Proposition 2. *Given dynamics in* (2) *and a cost function* J *of the form* (3)*, the mode insertion gradient and Hessian terms are given by* (4) *and* (5)*.*

Proof. Split (3) into state and control terms, such that

$$J = \int_{t_o}^{t_f} l_1(x(t))\,\mathrm{d}t + m(x(t_f)) + \int_{t_o}^{t_f} \frac{1}{2}\|u(t)\|_R^2\,\mathrm{d}t,$$

where $l_1(x)$ represents a control-independent running cost term. Let J_x and J_u denote the state and control parts,

respectively. Then,

$$\frac{dJ}{d\lambda_+} = \frac{dJ_x}{d\lambda_+} + \frac{dJ_u}{d\lambda_+} \qquad \text{and} \qquad \frac{d^2J}{d\lambda_+^2} = \frac{d^2J_x}{d\lambda_+^2} + \frac{d^2J_u}{d\lambda_+^2}.$$

From [24], [33],

$$\frac{dJ_x}{d\lambda_+} = \rho^T(f_2 - f_1), \tag{13}$$

and, from [9],

$$\frac{d^2J_x}{d\lambda_+^2} = (f_2-f_1)^T\Omega(f_2-f_1) + \rho^T(D_xf_2\cdot f_2 + D_xf_1\cdot f_1 - 2D_xf_1\cdot f_2) - D_xl\cdot(f_2-f_1). \tag{14}$$

For $u(t) = v(t)\ \forall\ t \notin [\tau, \tau+\lambda]$, J_u can be rewritten as

$$J_u = \frac{1}{2}\int_{t_o}^{\tau}\|v(t)\|_R^2\,\mathrm{d}t + \frac{1}{2}\int_{\tau}^{\tau+\lambda}\|u(t)\|_R^2\,\mathrm{d}t + \frac{1}{2}\int_{\tau+\lambda}^{t_f}\|v(t)\|_R^2\,\mathrm{d}t.$$

Using the Leibniz integral rule,

$$\frac{dJ_u}{d\lambda} = \frac{1}{2}\|u(\tau+\lambda)\|_R^2 - \frac{1}{2}\|v(\tau+\lambda)\|_R^2$$

and taking the limit $\lambda \to 0$,

$$\frac{dJ_u}{d\lambda_+} = \frac{1}{2}\|u(\tau)\|_R^2 - \frac{1}{2}\|v(\tau)\|_R^2. \tag{15}$$

Given that each inserted control is constant with respect to the applied duration λ, (15) becomes

$$\frac{d^2J_1}{d\lambda_+^2} = 0. \tag{16}$$

From (13) and (15), the mode insertion gradient becomes

$$\frac{dJ}{d\lambda_+} = \rho^T(f_2-f_1) + \frac{1}{2}\|u(\tau)\|_R^2 - \frac{1}{2}\|v(\tau)\|_R^2,$$

which matches (4). Similarly, from (14) and (16), the mode insertion Hessian becomes

$$\frac{d^2J}{d\lambda_+^2} = (f_2-f_1)^T\Omega(f_2-f_1) + \rho^T(D_xf_2\cdot f_2 + D_xf_1\cdot f_1 - 2D_xf_1\cdot f_2) - D_xl\cdot(f_2-f_1),$$

which matches (5). □

REFERENCES

[1] H. K. Khalil, *Noninear Systems*. Prentice-Hall, New Jersey, 1996.

[2] P. V. Kokotovic, "The joy of feedback: nonlinear and adaptive," *IEEE Control systems*, vol. 12, no. 3, pp. 7–17, 1992.

[3] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.

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

[5] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming–The explicit solution," *IEEE Transactions on Automatic Control*, vol. 47, no. 12, pp. 1974–1985, 2002.

[6] F. Allgöwer, R. Findeisen, and Z. K. Nagy, "Nonlinear model predictive control: From theory to application," *Journal of the Chinese Institute of Chemical Engineers*, vol. 35, no. 3, pp. 299–315, 2004.

[7] L. S. Pontryagin, *Mathematical theory of optimal processes*. Routledge, 2018.

[8] 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.

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

[10] ——, "Sequential action control for models of underactuated underwater vehicles in a planar ideal fluid," in *American Control Conference (ACC)*, 2016, pp. 4500–4506.

[11] T. Fan and T. Murphey, "Online feedback control for input-saturated robotic systems on Lie groups," in *Robotics: Science and Systems Conference (RSS)*, 2016.

[12] A. Ansari, K. Flaßkamp, and T. D. Murphey, "Sequential action control for tracking of free invariant manifolds," in *Conference on Analysis and Design of Hybrid Systems*, 2015, pp. 335–342.

[13] A. D. Wilson, J. A. Schultz, A. R. Ansari, and T. D. Murphey, "Real-time trajectory synthesis for information maximization using sequential action control and least-squares estimation," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2015, pp. 4935–4940.

[14] 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.

[15] C. T. Kelley, *Iterative methods for optimization*. SIAM, 1999.

[16] D. G. Luenberger and Y. Ye, *Linear and nonlinear programming*. Springer, 2015, vol. 228.

[17] D. Murray and S. Yakowitz, "Differential dynamic programming and Newton's method for discrete optimal control problems," *Journal of Optimization Theory and Applications*, vol. 43, no. 3, pp. 395–414, 1984.

[18] J. Hauser, "A projection operator approach to the optimization of trajectory functionals," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 377–382, 2002.

[19] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems." in *ICINCO (1)*, 2004, pp. 222–229.

[20] T. M. Caldwell and T. D. Murphey, "Switching mode generation and optimal estimation with application to skid-steering," *Automatica*, vol. 47, no. 1, pp. 50–64, 2011.

[21] Y. Wardi, M. Egerstedt, and P. Twu, "A controlled-precision algorithm for mode-switching optimization," in *IEEE Conference on Decision and Control (CDC)*, 2012, pp. 713–718.

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

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

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

[25] L. Armijo, "Minimization of functions having Lipschitz continuous first partial derivatives," *Pacific Journal of mathematics*, vol. 16, no. 1, pp. 1–3, 1966.

[26] R. Fierro, F. L. Lewis, and A. Lowe, "Hybrid control for a class of underactuated mechanical systems," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 29, no. 6, pp. 649–654, 1999.

[27] W. Sun, E. A. Theodorou, and P. Tsiotras, "Game theoretic continuous time differential dynamic programming," in *American Control Conference (ACC), 2015*. IEEE, 2015, pp. 5593–5598.

[28] ——, "Continuous-time differential dynamic programming with terminal constraints," in *Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. IEEE, 2014, pp. 1–6.

[29] A. N. K. Nasir, M. A. Ahmad, and M. F. Rahmat, "Performance comparison between LQR and PID controllers for an inverted pendulum system," in *AIP Conference Proceedings*, vol. 1052, no. 1. AIP, 2008, pp. 124–128.

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

[31] A. M. Bloch, M. Leok, J. E. Marsden, and D. V. Zenkov, "Controlled Lagrangians and stabilization of the discrete cart-pendulum system," in *IEEE Conference on Decision and Control (CDC) and the European Control Conference (ECC)*, 2005, pp. 6579–6584.

[32] E. Todorov and W. Li, "A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *Proceedings of the American Control Conference*, 2005, pp. 300–306.

[33] M. Egerstedt, Y. Wardi, and H. Axelsson, "Optimal control of switching times in hybrid systems," in *9th IEEE International Conference on Methods and Models in Automation and Robotics*, 2003.