



Offline state estimation for hybrid systems via nonsmooth variable projection[☆]



Jize Zhang^a, Andrew M. Pace^b, Samuel A. Burden^b, Aleksandr Aravkin^{a,*}

^a Department of Applied Mathematics, University of Washington, Seattle, United States of America

^b Department of Electrical & Computer Engineering, University of Washington, Seattle, United States of America

ARTICLE INFO

Article history:

Received 25 May 2019

Received in revised form 31 October 2019

Accepted 16 January 2020

Available online 18 February 2020

Keywords:

State estimation

Hybrid systems

Nonlinear systems

Mechanical systems

Optimization

ABSTRACT

We propose an offline algorithm that simultaneously estimates discrete and continuous components of a hybrid system's state. We formulate state estimation as a continuous optimization problem by relaxing the discrete component and using a robust loss function to accommodate large changes in the continuous component during switching events. Subsequently, we develop a novel nonsmooth variable projection algorithm with Gauss–Newton updates to solve the state estimation problem and prove the algorithm's global convergence to stationary points. We demonstrate the effectiveness of our approach by comparing it to a state-of-the-art filter bank method, and by applying it to simple piecewise-linear and -nonlinear mechanical systems undergoing intermittent impact.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

This paper considers the problem of using noisy measurements from a piecewise-continuous trajectory to estimate a hybrid system's state. A *hybrid* dynamical system switches between dynamic regimes at time- or state-triggered events. The state estimation problem has been extensively studied in *classical* dynamical systems whose states evolve according to one (possibly time-varying) smooth model. This problem is fundamentally more challenging for hybrid systems since the set of discrete state¹ sequences generally grows combinatorially in time.

When the discrete state sequence and switching times are known *a priori* or directly measured, only the continuous state needs to be estimated, yielding a classical state estimation problem; this approach has been applied to piecewise-linear systems (Stengel, 1994, Chap. 4.5) and to nonlinear mechanical

systems undergoing impacts (Menini & Tornambe, 2001). When the discrete state is not known or measured, estimating both the discrete and continuous states simultaneously improves estimation performance. One approach uses a bank of filters, each tuned to one discrete state, and selects the discrete states as the filter with the lowest residual (Balluchi, Benvenuti, Di Benedetto, & Sangiovanni-Vincentelli, 2002, §4.1). This filter bank method has been applied to hybrid systems with linear dynamics (Balluchi, Benvenuti, Di Benedetto, & Sangiovanni-Vincentelli, 2003, §4.1) (Gómez-Gutiérrez, Čelikovský, Ramírez-Treviño, Ruiz-Léon, & Gennaro, 2011), nonlinear dynamics (Barhoumi, Msahli, Djemai, & Busawon, 2012), and jumps in the continuous state when the discrete state changes (Balluchi, Benvenuti, Di Benedetto, & Sangiovanni-Vincentelli, 2013). Likewise, particle filter methods for hybrid systems (Blom & Bloem, 2007; Doucet, Gordon, & Krishnamurthy, 2001; Seah & Hwang, 2009) use a collection of filters, identified as particles, and are applicable to more general nonlinear process dynamics. Particle filters and filter banks are effective when the number of discrete states and dimension of continuous state spaces are small.

Another approach formulates a moving-horizon estimator over both the continuous and discrete states, resulting in a mixed-integer optimization problem (Bemporad, Mignone, & Morari, 1999). The inherently discrete nature of the problem formulation enables estimation of the exact sample when the discrete state switches, at the expense of combinatorial growth of the set of discrete decision variables as the horizon increases. Multiple methods have been developed to mitigate the challenge posed by this combinatorial complexity. One approach entails summarizing past measurements and state estimates with a penalty term in

[☆] This material is based on work supported in part by the Washington Research Foundation, United States of America Data Science Professorship, in part by the U. S. Army Research Laboratory and the U. S. Army Research Office under grant #W911NF-16-1-0158, and in part by the National Science Foundation Cyber-Physical Systems program award #1565529. The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Antonio Vicino under the direction of Editor Torsten Söderström.

* Corresponding author.

E-mail addresses: jizez@uw.edu (J. Zhang), apace2@uw.edu (A.M. Pace), sburden@uw.edu (S.A. Burden), saravkin@uw.edu (A. Aravkin).

¹ The state of the hybrid system is specified by the discrete and continuous components. We refer to the discrete component of the hybrid system state as the *discrete state*, and refer to the continuous component as the *continuous state*.

the objective function (Ferrari-Trecate, Mignone, & Morari, 2002). Another approach, applicable to systems with bounded noise, entails restricting the set of possible discrete state sequences using *a priori* knowledge of the system (Alessandri, Baglietto, & Battistelli, 2005, 2007).

An alternative approach to circumventing the combinatorial challenge entailed by exactly estimating the discrete state sequence involves *relaxing* the discrete state estimate to take on continuous values as in Bako and Lecoeuche (2013) and Johnson (2016). The latter reference uses a sparsity-promoting convex program whose objective incorporates a nonsmooth penalty across all possible discrete state sequences, and guarantees the estimate converges to the true continuous and discrete states. Both approaches are formulated for piecewise-linear systems whose continuous states do not jump when switching between subsystems; in the language of hybrid systems, the continuous states are *reset* using the identity function.

Our approach and contributions

We propose an offline algorithm for estimating the state of hybrid systems with nonlinear dynamics, non-identity resets, and noisy process and observation models. Although prior work accommodates aspects of our problem formulation, to the best of our knowledge no work simultaneously allows nonlinear dynamics and non-identity resets: Balluchi et al. (2013) do not allow nonlinear dynamics, Blom and Bloem (2007) and Ferrari-Trecate, Mignone, and Morari (2002) do not allow non-identity reset, and Bako and Lecoeuche (2013) do not allow either nonlinear dynamics nor non-identity resets. Our starting point is the optimization perspective on generalized and robust state estimation (Aravkin, Burke, Ljung, Lozano, & Pillonetto, 2017; Aravkin, Burke, & Pillonetto, 2012). To formulate state estimation as a continuous optimization problem, we relax the discrete state to take on continuous values as in prior work. Unlike prior work on state estimation for hybrid systems, we model process noise using the Student's t distribution, which allows large innovations and makes the method applicable to systems with non-identity resets.

In combination, these elements yield a nonsmooth nonconvex continuous optimization formulation for offline state estimation (Section 2). We develop a Gauss–Newton type algorithm to solve this problem and prove the algorithm globally converges to stationary points (Section 3). The algorithm is compared to a class of state-of-the-art algorithms (Section 5) and evaluated on piecewise-linear and -nonlinear hybrid system models (Section 6).

2. Problem formulation

We consider observational data periodically sampled from a continuous-time hybrid dynamical system (Goebel, Sanfelice, & Teel, 2009) that undergoes occasional jumps in continuous state, such as a mechanical system undergoing intermittent impacts (Johnson, Burden, & Koditschek, 2016). We utilize a *discrete-time switched system* as the process model for this sampled data. The process model is chosen to capture the salient features of a hybrid dynamical system model, e.g. the continuous-time dynamics differing between discrete states, while shifting the challenge of non-identity resets to the process noise. As we explain below, combining this process model with a Student's t distribution for the process noise captures the salient features of the underlying system dynamics while enabling our derivation of a computationally efficient state estimation algorithm.

2.1. Process and observation models

We use a *discrete-time switched system*

$$\begin{aligned} x_{t+1} &= \sum_{m=1}^M \mathcal{F}_m(x_t) w_t[m] + \sigma_t \\ y_t &= \mathcal{H}_t(x_t) + \delta_t \end{aligned} \quad (1)$$

where $m \in \{1, \dots, M\}$ indexes the continuously-differentiable process model $\mathcal{F}_m: \mathbb{R}^n \rightarrow \mathbb{R}^n$, $M \in \mathbb{N}$ is the number of process models, $\mathcal{H}_t: \mathbb{R}^n \rightarrow \mathbb{R}^d$ is the continuously-differentiable observation model that generates observations $y_t \in \mathbb{R}^d$ of the hidden continuous state $x_t \in \mathbb{R}^n$, σ_t, δ_t are process and measurement noises, and $w_t \in \mathcal{D}^M$ is a *one-hot* vector² that indicates which process model is active at time t . Note that the observation model does not depend explicitly on the active model \mathcal{F}_m , which must be inferred from measurements of the continuous state x_t .

The model \mathcal{F}_m that is active during each time step may be determined by an exogenous signal, prescribed as a function of time or state, or some combination thereof. Thus, the equation in (1) can represent the process and observation models of a wide variety of hybrid systems. Appendix provides an overview of the construction of a switched system by sampling a general hybrid dynamical system. We are motivated theoretically and experimentally to focus on cases where the active model \mathcal{F}_m is constant for many time steps, only occasionally switching to a new model. When the sampling rate of a continuous-time hybrid dynamical system is much faster than the *dwelt-time* (Hespanha & Morse, 1999), consecutive measurements will often be from the hybrid system in the same discrete state.

The problem of when measurements from a switched-system as in (1) with no process noise $\sigma_t \sim 0$, and no measurement noise $\delta_t \sim 0$, can reconstruct the true discrete and continuous state (i.e. when is the system is observable) is well studied continuous time switched linear systems (Vidal, Chiuso, Soatto, & Sastry, 2003) (Johnson, 2016, Chpt. 2). For the more general linear hybrid system, when the continuous state undergoes occasional jumps, observability tests with particular assumptions have been proposed (Balluchi et al., 2003). To the best of our knowledge there is not a general observability test that applies to nonlinear hybrid systems with non-identity resets; a class of hybrid systems considered in this paper.

When the discrete state changes in a hybrid system, the continuous state may change abruptly according to a *reset* map. As an example, the velocity of a rigid mass changes abruptly when it impacts a rigid surface (Lötstedt, 1982). Empirically, these discrete reset dynamics are much more poorly characterized than their continuous counterparts. For instance, whereas the ballistic trajectory of a rigid mass is well-approximated by Newton's laws, the abrupt change in velocity that occurs at impact is not consistent with any established impact law (Fazeli, Zapolsky, Drumwright, & Rodriguez, 2017). Including such a reset in the system model (1) will introduce bias into the state estimate because the model will generate erroneous predictions at resets, diminishing the accuracy of estimated states at nearby times. This observation motivates us in the next section to account for the effect of unknown resets as part of the process noise.

2.2. Process noise and observation noise models

Instead of incorporating continuous state resets explicitly into the model (1), we introduce a distributional assumption on the process noise σ_t that accepts large instantaneous changes in the

² $w \in \mathbb{R}^M$ is *one-hot* if $w[i] \in \{0, 1\}$ for all $i \in \{1, \dots, M\}$ and $1^T w = 1$; $\mathcal{D}^M \subset \mathbb{R}^M$ denotes the set of one-hot vectors.

continuous state estimate. Specifically, we assume that process noise σ_t follows a Student's t distribution. However, we emphasize that this is a modeling assumption. It does not imply that process noise from real hybrid system has to follow this distribution. Compared with the commonly-used Gaussian distribution, the *heavy-tailed* Student's t is tolerant to large deviations in the estimate of the hidden continuous state x_t (Aravkin, Burke, & Pillonetto, 2014). Hence, the Student's t error model allows an instantaneous change in the state that is consistent with (1) before and after the change. The negative log-likelihood of the Student's t (as a function of σ_t) is given by

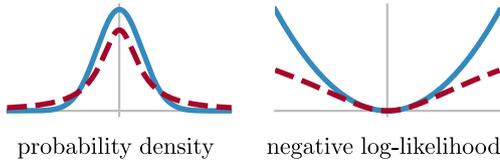
$$r \log \left(r + \|Q^{-1/2} \sigma_t\|^2 \right) - C(r), \quad (2)$$

where r is the degrees-of-freedom parameter of the Student's t , and Q is the covariance matrix, and $C(r)$ is a term independent of σ_t .

If the continuous state x_t was known, then any residual between the predicted observations $\mathcal{H}_t(x_t)$ and actual measurements y_t at time t is due to measurement noise; in particular, the residual does not exhibit large deviations due to continuous state resets at switching times. Thus, we assume the measurement noise δ_t follows the usual Gaussian distribution, with negative log-likelihood

$$\frac{1}{2} \|R^{-1/2} \delta_t\|^2, \quad (3)$$

where R is the covariance matrix. The plots below provide a comparison between the probability density (left) and the negative log-likelihood (right) for the scalar Gaussian (solid blue) and Student's t distributions (dashed red; degree-of-freedom $r = 1$).



2.3. State estimation problem formulation

We derive the objective function for estimating states of (1) using *maximum a posteriori* (MAP) likelihood. Including the constraint on w , we obtain the optimization problem

$$\min_{x_t \in \mathbb{R}^n, w_t \in \mathcal{D}^M} \sum_{t=0}^{T-1} l_{\text{meas}}(x_t, y_t) + l_{\text{proc}}(x_t, y_t, w_t) \quad (4)$$

where

$$l_{\text{meas}}(x_t, y_t) = \frac{1}{2} \|R^{-1/2} (y_t - \mathcal{H}_t(x_t))\|^2$$

and

$$l_{\text{proc}}(x_t, y_t, w_t) = r \log \left(r + \left\| Q^{-1/2} \left(x_{t+1} - \sum_{m=1}^M \mathcal{F}_m(x_t) w_t[m] \right) \right\|^2 \right).$$

Problem (4) is a nonlinear mixed-integer program with respect to both the continuous (x_t) and discrete (w_t) decision variables, with the discrete variable constrained to be a one-hot vector ($w_t \in \mathcal{D}^M$). We can significantly simplify the structure by establishing the following lemma.

Lemma 1 (Formulation Equivalence). Given $w \in \mathcal{D}^M$, any vectors x_1, x_2 , models \mathcal{F}_i , and any penalty functional g , we have

$$\min_{w \in \mathcal{D}^M} g \left(x_2 - \sum_{m=1}^M w[m] \mathcal{F}_m(x_1) \right)$$

$$= \min_{w \in \mathcal{D}^M} \sum_{m=1}^M w[m] g(x_2 - \mathcal{F}_m(x_1))$$

and

$$\begin{aligned} & \operatorname{argmin}_{w \in \mathcal{D}^M} g \left(x_2 - \sum_{m=1}^M w[m] \mathcal{F}_m(x_1) \right) \\ &= \operatorname{argmin}_{w \in \mathcal{D}^M} \sum_{m=1}^M w[m] g(x_2 - \mathcal{F}_m(x_1)). \end{aligned}$$

Proof. Since $w \in \mathcal{D}^M$ for both problems, there are only M possible values for both objective functions, i.e.

$$g(x_2 - \mathcal{F}_1(x_1)), \quad g(x_2 - \mathcal{F}_2(x_1)), \quad \dots, \quad g(x_2 - \mathcal{F}_M(x_1)).$$

Hence, the minimum objective value for both problems will be $\min_m g(x_2 - \mathcal{F}_m(x_1))$ and every minimizer is a one-hot vector that selects a minimum value. \square

Based on Lemma 1, an equivalent formulation to (4) is given by

$$\begin{aligned} & \min_{x_t \in \mathbb{R}^n, w_t \in \mathcal{D}^M} \sum_{t=0}^{T-1} \left(\frac{1}{2} \|R^{-1/2} (y_t - \mathcal{H}_t(x_t))\|^2 + \right. \\ & \left. \sum_{m=1}^M w_t[m] r \log \left(r + \|Q^{-1/2} (x_{t+1} - \mathcal{F}_m(x_t))\|^2 \right) \right). \end{aligned} \quad (5)$$

Although still a mixed-integer program, this reformulation exhibits linear coupling between the discrete variables w_t and continuous variables x_t . We will leverage this linear coupling when we develop our estimation algorithm based on the relaxed problem formulation introduced in the next section.

2.4. Relaxed state estimation problem formulation

Ultimately, the discrete state estimate will be specified as a one-hot vector, $w_t \in \mathcal{D}^M \subset \mathbb{R}^M$. To formulate a continuous optimization problem that approximates the mixed-integer problem formulated in the previous section, we relax the decision variable w_t to take values in the convex hull Δ^M of \mathcal{D}^M . We use $\Delta^M := \{w \in [0, 1]^M : 1^T w = 1\}$ to denote the simplex in \mathbb{R}^M . The optimal relaxed w_t will generally lie on the interior of the simplex, so we project the result from our relaxed optimization problem to return the one-hot discrete state estimate. Since this relaxation-optimization-projection process tends to induce frequent changes in the discrete state estimate, we introduce a smoothing term on w_t ,

$$v \|w_{t+1} - w_t\|_2^2,$$

yielding the continuous relaxation of (5) given by

$$\begin{aligned} & \min_{x_t \in \mathbb{R}^n, w_t \in \Delta^M} f(x, w) := \sum_{t=0}^{T-1} \left(\frac{1}{2} \|R^{-1/2} (y_t - \mathcal{H}_t(x_t))\|^2 \right. \\ & \left. + \sum_{m=1}^M w_t[m] r \log \left(r + \|Q^{-1/2} (x_{t+1} - \mathcal{F}_m(x_t))\|^2 \right) \right. \\ & \left. + v \|w_{t+1} - w_t\|_2^2 \right), \end{aligned} \quad (6)$$

where x is the concatenated variable containing all x_t , w is the concatenated variable containing all w_t , and v is a parameter controlling the strength of smoothing. The optimal relaxed discrete state estimate $w_t \in \Delta^M$ is projected onto \mathcal{D}^M by choosing the (unique) one-hot vector whose $\operatorname{argmax}_m w_t[m]$ component is equal to 1.

3. State estimation algorithm

In this section, we derive an algorithm to solve the relaxed state estimation problem formulated in (6) using two key ideas:

- (1) nonsmooth variable projection;
- (2) Gauss–Newton descent with Student’s t penalties.

These two ideas are explained in the next two subsections, followed by a convergence analysis in the third subsection.

3.1. Nonsmooth variable projection

The first idea is to pass to the *value function*, projecting out (partially minimizing over) the w variables, so as to reduce the number of variables to optimize over. Define

$$v(x) := \min_w f(x, w) \quad (7)$$

with $f(x, w)$ as in (6). The objective $f(x, w)$ is convex in w , but not strictly convex. To guarantee differentiability of $v(x)$, we add a smoothing term and consider

$$v_\beta(x) := \min_w f(x, w) + \frac{\beta}{2} \|w\|^2. \quad (8)$$

where β is usually taken to be a very small number (e.g. 10^{-4} or smaller) so that the added term has minimal effect on the original value function. (The minimizer of v_β is different from that of v .) The function $v_\beta(x)$ is a *Moreau envelope* (Rockafellar & Wets, 1998, Def 1.22) of the true value function v ; we refer the interested reader to Aravkin, Drusvyatskiy, and van Leeuwen (2016) for details and examples concerning the Moreau envelope specifically (and nonsmooth variable projection more broadly). The unique minimizer $w(x)$ can be found quickly and accurately since the minimization problem with respect to w is strongly convex: projected gradient descent converges linearly and can be accelerated using the Fast Iterative Shrinkage-Thresholding Algorithm (FISTA) (Beck & Teboulle, 2009). With the minimizer $w(x)$, the gradient of v_β is readily computed as

$$\nabla v_\beta(x) = \partial_x f(x, w)|_{w=w(x)}. \quad (9)$$

Plugging $w(x)$ back into (6) we obtain the problem

$$\begin{aligned} \min_x v_\beta(x) = & \frac{1}{2} \sum_{t=0}^{T-1} \|y_t - \mathcal{H}(x_t)\|_{R^{-1}}^2 + v \|w_{t+1}(x) - w_t(x)\|_2^2 \\ & + \sum_{m=1}^M w_{t,m}(x) r \log \left(1 + \frac{\|x_{t+1} - \mathcal{F}_m(x_t)\|_{Q^{-1}}^2}{r} \right) \\ & + \frac{\beta}{2} \|w(x)\|^2, \end{aligned} \quad (10)$$

where $w_{t,m}(x) \equiv w_t[m](x)$.

3.2. Gauss–Newton descent with Student’s t penalties

We derive a Gauss–Newton descent algorithm to solve (10) based on a line search method first proposed in Burke (1985) for convex composite problems. To apply the method we first cast the objective in (10) into a convex composite function, let $v_\beta = \rho \circ F$, where

$$F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \end{pmatrix}$$

with

$$f_1(x) = \frac{1}{2} \sum_{t=0}^{T-1} \sum_{m=1}^M w_{t,i}(x) r \log \left(1 + \frac{\|x_{t+1} - \mathcal{F}_m(x_t)\|_{Q^{-1}}^2}{r} \right)$$

$$+ v \|w_{t+1}(x) - w_t(x)\|_2^2 + \frac{\beta}{2} \|w(x)\|^2$$

$$f_2(x) = \mathcal{H}(x) - y$$

and

$$\rho \begin{pmatrix} c \\ u \end{pmatrix} = c + \frac{1}{2} \|u\|_{R^{-1}}^2 + \delta_{[0,+\infty]}(c).$$

At each iteration, we choose a search direction $d^*(x)$ that

$$\begin{aligned} d^* \in & \operatorname{argmin}_d \rho(F(x) + F^{(1)}(x)d) + \frac{1}{2} d^T U(x) d \\ \in & \operatorname{argmin}_d f_1(x) + \nabla f_1(x)d + \frac{1}{2} \|f_2(x) + \nabla f_2(x)d\|_{R^{-1}}^2 \\ & + \frac{1}{2} d^T U(x) d \\ \in & \operatorname{argmin}_d \frac{1}{2} d^T (U(x) + \nabla \mathcal{H}(x)^T R^{-1} \nabla \mathcal{H}(x)) d \\ & + \nabla v_\beta(x)^T d \end{aligned} \quad (11)$$

where the equivalence is obtained by dropping terms independent of d . In general $U(x)$ can be any positive semidefinite matrix that varies continuously with respect to x , but for our particular objective function involving Student’s t penalty, $U(x)$ is chosen to be a Hessian approximation of the Student’s t term in $f_1(x)$. Therefore the update can be interpreted as a Gauss–Newton style update. This approximation, proposed in Aravkin et al. (2014, (5.5), (5.6)), is employed here because of its significant computational advantage; it is of the form

$$U = \begin{bmatrix} U_1 & A_2^T & 0 & \\ A_2 & U_2 & A_3^T & 0 \\ 0 & \ddots & \ddots & \ddots \\ 0 & 0 & A_T & U_T \end{bmatrix} \quad (12)$$

with

$$A_t = -r \sum_{m=1}^M w_{t-1,m}(x) \frac{Q^{-1} \nabla \mathcal{F}_m(x_{t-1})}{r + \|x_t - \mathcal{F}_m(x_{t-1})\|_{Q^{-1}}^2},$$

$$\begin{aligned} U_t = & r \sum_{m=1}^M \frac{w_{t,m}(x) \nabla \mathcal{F}_m(x_t)^T Q^{-1} \nabla \mathcal{F}_m(x_t)}{r + \|x_{t+1} - \mathcal{F}_m(x_t)\|_{Q^{-1}}^2} \\ & + \frac{w_{t-1,m}(x) Q^{-1}}{r + \|x_t - \mathcal{F}_m(x_{t-1})\|_{Q^{-1}}^2} \end{aligned}$$

for $1 \leq t \leq T-1$, and

$$U_T = \frac{r w_{T-1,m}(x) Q^{-1}}{r + \|x_T - \mathcal{F}_m(x_{T-1})\|_{Q^{-1}}^2}.$$

We can rewrite $U(x)$ as

$$U(x) = \sum_m \mathcal{F}_m(x)^T \tilde{Q}_m(w(x))^{-1} \mathcal{F}_m(x),$$

where

$$G_m(x) = \begin{bmatrix} I & 0 & 0 & \\ -\nabla \mathcal{F}_m(x_2) & I & 0 & 0 \\ 0 & \ddots & \ddots & \ddots \\ \dots & 0 - \nabla \mathcal{F}_m(x_T) & I & \end{bmatrix}$$

and

$$\begin{aligned} \tilde{Q}_m(w(x))^{-1} &= \operatorname{diag}(\tilde{Q}_{m,t}(w(x))^{-1}) \\ \tilde{Q}_{m,t}(w(x))^{-1} &= \frac{r w_{t-1,m}(x) Q^{-1}}{r + \|x_t - \mathcal{F}_i(x_{t-1})\|_{Q^{-1}}^2}. \end{aligned}$$

Clearly $U(x)$ is positive semidefinite; we show in Lemma 3 that $U(x)$ is actually positive definite, so problem (11) reduces to the block tridiagonal linear system

$$(U(x) + \nabla \mathcal{H}(x)^T R^{-1} \nabla \mathcal{H}(x)) d + \nabla v_\beta(x) = 0.$$

Given $d^*(x)$, the new x^+ is of the form

$$x^+ = x + \delta d^*,$$

where δ is a step size selected using the Armijo-type (Nocedal & Wright, 2006, Sec. 3.1) line search criterion.

$$\delta = \max\{\gamma^l: \rho(F(x) + \gamma^l d^*) \leq \rho(F(x)) + c\gamma^l \Delta(x; d^*) \text{ and } c \in (0, 1)\} \quad (13)$$

with

$$\Delta(x; d) = \rho(F(x) + F^{(1)}(x)d) + \frac{1}{2}d^T U(x)d - \rho(F(x)).$$

When $d = 0$, we have $\Delta(x; 0) = 0$,³ and since we choose the minimizing

$$d^* = \underset{d}{\operatorname{argmin}} \rho(F(x) + F^{(1)}(x)d) + \frac{1}{2}d^T U(x)d,$$

we have $\Delta(x; d^*) \leq 0$. Further,

$$\begin{aligned} \Delta(x; d^*) = 0 &\Leftrightarrow 0 \in \underset{d}{\operatorname{argmin}} \rho(F(x) + F^{(1)}(x)d) + \frac{1}{2}d^T U(x)d \\ &\Leftrightarrow 0 \in \partial \rho(F(x))F^{(1)}(x) \end{aligned}$$

by Burke (1985, Thm. 3.6). In other words, stationarity is achieved when $\Delta(x; d^*) = 0$. When $\Delta(x; d) < 0$, we are guaranteed to have descent

$$\rho(F(x) + F^{(1)}(x)d) < \rho(F(x))$$

since $U(x)$ is positive semidefinite. This condition ensures that the line search step (13) is well-defined (Burke, 1985, Lemma 2.3).

Our approach is summarized in Algorithm 1. The positive parameter ϵ in the algorithm specifies the stopping condition. Finally, we project the relaxed discrete state estimate $w_t \in \Delta^M$ to obtain a discrete state estimate in \mathcal{D}^M as described in Section 2.4.

Algorithm 1 Variable Projection for (6)

Require: $x, w, Q, R, r, v, \beta, \epsilon$

- 1: **for** $k = 1, 2, 3, \dots$ **do**
 - 2: $d^{(k)} \leftarrow$ Gauss–Newton direction for $x^{(k)}$
 - 3: $x^{(k+1)} \leftarrow x^{(k)} + \delta d^{(k)}$
 - 4: $w^{(k+1)} \leftarrow \operatorname{InnerSolver}_{\Pi_t \Delta}(w^{(k)})$
 - 5: $\operatorname{loss}_k \leftarrow f(x^{(k+1)}, w^{(k+1)})$
 - 6: Iterate till $\Delta(x^{(k)}; d^{(k)}) \geq -\epsilon$.
-

3.3. Convergence of state estimation algorithm

In this section we show the convergence of the proposed algorithm. The convergence of Algorithm 1 to a stationary point for a general class of convex composite objective functions is established in Aravkin et al. (2014) and Burke (1985). In particular (Aravkin et al., 2014, Theorem 5.1) establishes the possible outcomes when applying this type of algorithm; informally, either the algorithm converges or the search direction d_k diverges. In the remainder of this section we provide two technical results needed to formalize this intuition and to apply the aforementioned theorem:

³ We overload Δ here to match the notation in Aravkin et al. (2014) and Burke (1985); $\Delta(x; d^*)$ should not be confused with Δ^M , which is used to denote the simplex containing relaxed state estimates.

- Lemma 2 establishes a set of sufficient conditions that prevent divergence ($\|d^{(k)}\| \rightarrow \infty$);
- Lemma 3 proves that the sufficient conditions are satisfied.

Lemma 2. Let $\Lambda = \{y | \rho(y) \leq v_\beta(x^0)\}$. If $F^{-1}(\Lambda) = \{x | F(x) \in \Lambda\}$ is bounded and $U(x)$ is positive definite for all $x \in F^{-1}(\Lambda)$, then the hypotheses in Aravkin et al. (2014, Theorem 5.1) are satisfied and the sequence of search directions $\{d^{(k)}\}$ is bounded.

Proof. The hypotheses in Aravkin et al. (2014, Theorem 5.1) require that $F^{(1)}$ to be bounded and uniformly continuous on the set $S = \bar{c} \circ F^{(-1)}(\Lambda)$ where $\bar{c} \circ$ stands for the closed convex hull. $F^{(1)}$ is continuous on S since $f_1^{(1)}$ exists and is continuous by property of Moreau envelope and proximal operator, and $f_2^{(1)}$ is continuous trivially. Further, given that S is closed by definition and bounded by assumption, it is compact. Hence $F^{(1)}$ is bounded and uniformly continuous on S .

Now we need to show that the sequence of search direction is bounded. At any iteration, the search direction d we choose satisfies

$$0 \leq \rho(F(x) + F^{(1)}(x)d) + \frac{1}{2}d^T U(x)d \leq \rho(F(x)) \leq \rho(F(x^0))$$

where the first inequality relies on $\rho \geq 0$ and on the positive semidefinite property of $U(x)$; the second inequality comes from $\Delta(x; d) \leq 0$; the third inequality results from the line search condition that creates a decreasing sequence $\{\rho(F(x^{(k)}))\}$.

Since $\rho(F(x^0))$ is finite, $d^T U(x)d < \infty$ for all iterations. Because Λ is closed by closedness of ρ and F is continuous, $F^{-1}(\Lambda)$ is also closed. Along with its boundedness by assumption, $F^{-1}(\Lambda)$ is compact. Since $x \in F^{-1}(\Lambda) \mapsto \lambda_{\min}(U(x))$ is continuous, its image is bounded, hence given that $U(x)$ is positive definite there exists some $\lambda_{\min} > 0$ for all $x \in F^{-1}(\Lambda)$. Therefore $0 < \lambda_{\min} \|d\|^2 \leq d^T U(x)d < \infty$, which implies that $d^{(k)}$ cannot be unbounded. \square

Lemma 3. $F^{-1}(\Lambda)$ is bounded for problem (10) and $U(x)$ is positive definite for all $x \in F^{-1}(\Lambda)$.

Proof. First note that Λ is bounded by the coercivity of ρ . This implies that for an unbounded sequence $\|x^{(k)}\| \rightarrow \infty$, we still have $f_1(x^{(k)}) < \infty$ and $\|f_2(x^{(k)})\| < \infty$.

If $\|x^{(k)}\| \rightarrow \infty$, then we can find some $t + 1$ and a subsequence J such that $\lim_{k \in J} \|x_{t+1}^{(k)}\| = \infty$. By the definition of f_1 and $f_1(x^{(k)}) < \infty$, $\lim_{k \in J} \|\mathcal{F}_i(x_t^{(k)})\| = \infty$, which further implies that $\lim_{k \in J} \|x_t^{(k)}\| = \infty$. Iteratively this means that $\lim_{k \in J} \|x_t^{(k)}\| = \infty$ for all t , in particular for the given starting point x_0 , but that is not possible.

To show that $U(x)$ in (12) is positive definite, recall that we can rewrite $U(x)$ as

$$U(x) = \sum_m G_m(x)^T \tilde{Q}_m(w(x))^{-1} G_m(x) \succeq 0.$$

If there exists some d such that $d^T U(x)d = 0$, then

$$\begin{aligned} &d^T \left(\sum_m G_m(x)^T \tilde{Q}_m(w(x))^{-1} G_m(x) \right) d \\ &= \sum_m \underbrace{d^T G_m(x)^T}_{z_m(x)^T} \tilde{Q}_m(w(x))^{-1} \underbrace{G_m(x) d}_{z_m(x)} \\ &= \sum_m z_m(x)^T \tilde{Q}_m(w(x))^{-1} z_m(x) = 0, \\ &\Rightarrow z_m(x)^T \tilde{Q}_m(w(x))^{-1} z_m(x) = 0 \quad \forall i \\ &\Rightarrow z_{m,t}(x)^T \tilde{Q}_{m,t}(w(x))^{-1} z_{m,t}(x) = 0 \quad \forall t \quad \forall i \end{aligned}$$

since $\tilde{Q}_m(w(x))^{-1} = \text{diag}(\tilde{Q}_{m,t}(w(x))^{-1})$, and

$$\tilde{Q}_{m,t}(w(x))^{-1} = \frac{rw(x)_{t,m}Q^{-1}}{r + \|x_{t+1} - \mathcal{F}_m(x_t)\|_{Q^{-1}}^2}$$

are positive semidefinite. However because each $w_t \in \Delta$, there has to be some $\tilde{Q}_{m,t}^{-1} > 0$ for each t . Therefore $U(x)$ must be positive definite for all $x \in F^{-1}(\Delta)$. \square

4. Parameter tuning for proposed algorithm

Before we present numerical results, we include a general guidance on parameter tuning for the new algorithm. We discuss both standard parameters (e.g. Q , R) that must be tuned by any algorithm for this application, as well as the parameters ν and r which are specific to our approach. We first give a rough outline of steps we have taken to tune the parameters, followed by more detailed guidelines to tune each individual parameter.

- (1) Start with large r for Student's t , i.e. distribution close to Gaussian.
- (2) If Q and R are unknown, they are tuned such that the smooth part of trajectories can be well approximated.
- (3) Decrease degrees of freedom r of Student's t so that the nonsmooth part of trajectories can be captured.
- (4) Adjust smoothing coefficient ν to reduce number of switches.

For degrees of freedom r , one can start with a large value, meaning that the distribution is close to Gaussian, and decrease it later to capture jumps in the continuous state.

For covariance matrices Q and R , if empirical estimations are available, they can be supplied to the model directly. There is existing literature on estimation methods for noise covariance matrices (Duník, Straka, Kost, & Havlík, 2017). When such estimations are not available, we usually assume the matrices to be diagonal for simplicity, in which case the inverse of diagonal entries can also be interpreted as weights. The diagonal values of R represent variance for measurements. When choosing R , we consider the relative scale of measurements, e.g. measurements with smaller magnitude usually have smaller variance. For choices of diagonal values of Q , we usually assign smaller variance for observed states, e.g. positions in our examples, and larger variance for unobserved states.

The choice of smoothing coefficient ν depends on modeler's belief in frequency of switches. One can start with a small value of ν (i.e. little penalty on frequent switches), and gradually increase it, till the pattern of switches is close to modeler's belief.

We recommend having a short piece of manually labeled trajectories as a training set for the purpose of parameter tuning. After tuning, the user can apply the same parameters on larger dataset collected from similar scenarios.

In terms of sensitivity of estimation results on parameters, we had the following observations when running our experiments:

- The estimation result is not very sensitive to r . We were able to decrease r fairly aggressively during parameter tuning.
- For the diagonals of Q and R , we found that it was important to have values in the correct ranges, but the exact values taken were not crucial.
- For smoothing coefficient ν , we noticed that the switching times were sensitive to ν when ν was very small relative to the diagonal entries of Q^{-1} and R^{-1} . Since we assumed that the discrete states should not change too frequently, we used a slightly larger ν .

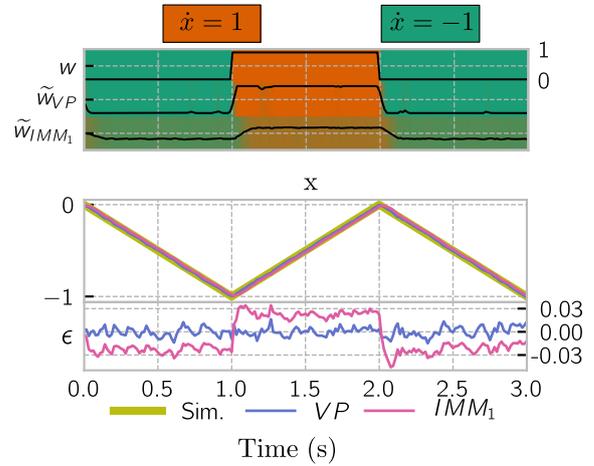


Fig. 1. Algorithm 1 (VP) performs comparably to IMM when the continuous state does not undergo any resets. The top plot shows the true state w and the simplex estimate of the true state from both methods \tilde{w}_{VP} , \tilde{w}_{IMM_1} . The simplex estimate is shown in color and the probability estimate of the discrete state being $w = 1$ is superimposed as a black line. The middle plot shows the actual value of the continuous state of the simulation and the estimates. The bottom plot shows the residual between true continuous state and the estimated continuous state.

5. Comparison with the Interacting Multiple Model (IMM) method

We compare the nonsmooth variable projection algorithm (Algorithm 1) with the Interacting Multiple Model (IMM) (Blom & Bar-Shalom, 1988) algorithm implemented in the open-source package filterpy (Labbe, 2014). We consider two examples, in both cases the continuous state x is a scalar, and there are two discrete states. In the first example, the continuous state x undergoes no jumps, i.e. the reset is the identity function. In the second example, the continuous state x undergoes an instantaneous jump when the discrete state changes; i.e. a non-identity reset. The dynamics of the two discrete state process models are:

$$\begin{aligned} \dot{x} &= -1 & \mathcal{F}_{w=1}, \\ \dot{x} &= 1 & \mathcal{F}_{w=2}. \end{aligned}$$

For the second example with non-identity resets, when a discrete state switch occurs, the continuous state decreases by 5. In both examples the discrete state switches at $t = 1$ and $t = 2$. Additionally, the measurement noise has a variance of $R = [.0001]$, which is used as the measurement noise covariance for all models. IMM_1 uses a process noise model of covariance $Q = [.001]$ for both the internal Kalman filters and IMM_2 uses a process normal process noise model with covariance $Q = [.2]$.

In the first example, Algorithm 1 (VP) and IMM perform nearly identically (Fig. 1). Both methods accurately recover the continuous state and discrete state. When the system undergoes instantaneous jumps in the continuous state at discrete state changes, Algorithm 1 outperforms IMM (Fig. 2). For IMM, there is a clear trade-off that exists between recovering the continuous state and recovering the discrete state. When using a process noise model with large covariance, as in the case of IMM_2 , the continuous state can be recovered at the expense of the discrete state. In the top subplot of Fig. 2, \tilde{w}_{IMM_2} is nearly the same value for the duration of the simulation, with slight separation between the two modes. With a smaller covariance, as in IMM_1 , the discrete state can be recovered. From $t = 1$ to near $t = 1.25$, IMM_1 incorrectly identifies the discrete state due to the continuous state jump direction being opposite of the continuous state dynamics for discrete state $w = 2$.

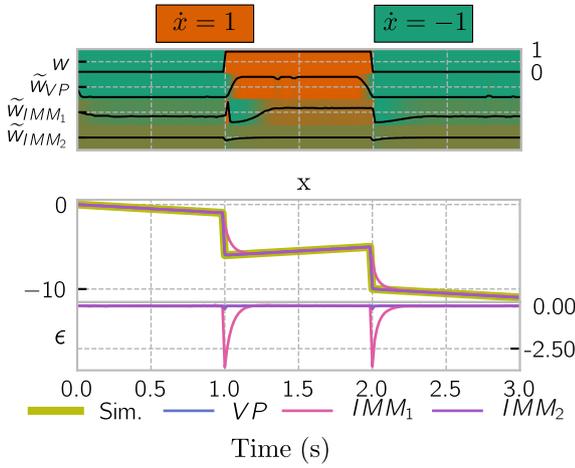


Fig. 2. Algorithm 1 (VP) outperforms the IMM when there are jumps in the continuous state. The plots follow the convention laid out in Fig. 1.

Both Algorithm 1 and IMM require a similar number of parameters from the user. For both methods, covariance matrices for the process error model Q and measurement error model R need to be provided. IMM adjusts the estimated frequency of switching between the discrete states via a probability transition matrix while Algorithm 1 uses the smoothing parameter ν , Section 2.4. Algorithm 1 has one additional parameter r due to the process noise model being Student's t distribution, which is crucial for obtaining accurate estimates with non-identity resets, Section 2.2.

6. Experiments with hybrid system models

To evaluate the proposed approach to state estimation for hybrid systems, we apply our algorithm to linear and nonlinear impact oscillators. In addition to being well-studied (Di Bernardo, Budd, Champneys, & Kowalczyk, 2008, §1.2), (Schatzman, 1998), these mechanical systems were chosen since they are among the simplest physically-relevant models that have non-identity reset maps. The parameter and trajectory regime considered in what follows is representative of a jumping robot constructed from one limb of a commercially-available quadrupedal robot (Kenneally, De, & Koditschek, 2016) and controlled with an event-triggered stiffness adjustment; Fig. 3a contains a photograph of the limb. The jumping robot's hip and foot are constrained to move vertically in a gravitational field, so the rigid pantograph mechanism depicted in Fig. 3b has two mechanical degrees-of-freedom (DOF) coupled through nonlinear pin-joint constraints. These two DOF are preserved, but their nonlinear coupling is neglected, in the piecewise-linear model illustrated in Fig. 3c. The hybrid dynamics of these linear and nonlinear impact oscillators are specified in Section 6.1

We perform two sets of experiments. The first set of experiments in Section 6.2 concern the piecewise-linear model depicted in Fig. 3c and explore the consequences of our modeling assumptions and the efficacy of our proposed algorithm:

- Section 6.2.1 demonstrates the advantage of employing a Student's t distribution for process noise as compared to a Gaussian distribution;
- Section 6.2.2 demonstrates the superior convergence rate yielded by Gauss-Newton descent directions as compared to gradient (steepest) descent;
- Section 6.2.3 demonstrates the advantage of smoothing the relaxed discrete state estimate; and

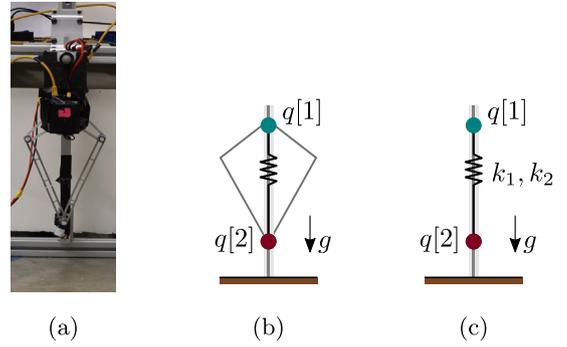


Fig. 3. Jumping robot and impact oscillator hybrid system models (Section 6.1). (a) Photograph of the physical robot (one leg from a Minitaur Kenneally et al., 2016) that inspired the simulation models. (b) Nonlinear model consisting of two masses coupled with a linear spring and a nonlinear pantograph mechanism. (c) Linear model consisting of two masses coupled with a linear spring.

- Section 6.2.4 demonstrates the algorithm's performance when *onboard* measurements are used instead of *offboard* measurements.

The second set of experiments in Section 6.3 evaluate our proposed approach using the nonlinear model depicted in Fig. 3b.

Since this section is devoted to comparing estimated states to ground truth simulation results, and since our approach entails the determination of a relaxed discrete state estimate *en route* to obtaining the discrete state estimate, we now introduce notation that distinguishes these quantities:

- $w_t \in \mathcal{D}^M$ denotes the ground truth discrete state;
- $\tilde{w}_t \in \Delta^M$ denotes the relaxed discrete state estimate;
- $\hat{w}_t \in \mathcal{D}^M$ denotes the discrete state estimate.

This notational distinction was not introduced previously in the interest of readability since there was no ambiguity entailed by overloading notation in the problem formulation and algorithm specification.

6.1. Impact oscillator hybrid system models

The continuous state $x = (q, \dot{q}) \in \mathbb{R}^4$ for the jumping robot hybrid system model consists of the two-dimensional configuration vector $q \in \mathbb{R}^2$ and corresponding velocity $\dot{q} \in \mathbb{R}^2$, where $q[1]$ and $q[2]$ denote the vertical height of the hip and foot, respectively. The foot is not permitted to penetrate the ground, $q[2] \geq 0$, so the first part of the discrete state indicates whether this constraint is active: A (air) if $q[2] > 0$, G (ground) if $q[2] = 0$. To compensate for energy losses at impact, an event-triggered controller stiffens or softens a spring based on which direction the hip is traveling, so the second part of the discrete state indicates the direction of travel for $q[1]$: \uparrow if up, \downarrow if down. With $\ddot{q}_m(q, \dot{q}) \in \mathbb{R}^2$ denoting the acceleration of the hip and foot in discrete state $m \in \{A\downarrow, G\downarrow, G\uparrow, A\uparrow\}$,⁴ formula for this acceleration is given in Table 1. At the moment of impact (when the discrete state changes from $w_t \in \{A\downarrow, A\uparrow\}$ to $w_{t+1} \in \{G\downarrow, G\uparrow\}$) the foot velocity $\dot{q}[2]$ is instantaneously *reset* to 0, corresponding to perfectly *plastic* impact. An example of the jump in continuous state when transitioning from $A\downarrow$ to $G\downarrow$ on the foot velocity $\dot{q}[2]$ is shown in Fig. 4 near time 17.5 s.

⁴ To simplify exposition we identify $m = A\downarrow$ with $m = 1$, $m = G\downarrow$ with $m = 2$, $m = G\uparrow$ with $m = 3$, and $m = A\uparrow$ with $m = 4$.

Table 1

Discrete states and continuous dynamics for impact oscillator hybrid system models (Section 6.1). Note that the continuous dynamics \ddot{q} have the same general form for both the piecewise-linear and -nonlinear models, with the spring law k being a linear or nonlinear function of the continuous state $x = (q, \dot{q})$ depending on which model is considered.

Discrete state w	Icon	$\ddot{q}_w(x)$
$w = A \downarrow$		$\begin{bmatrix} \frac{1}{m_h} (-k_1(q, \dot{q})) - g \\ \frac{1}{m_t} (k_1(q, \dot{q})) - g \end{bmatrix}$
$w = G \downarrow$		$\begin{bmatrix} \frac{1}{m_h} (-k_1(q, \dot{q})) - g \\ 0 \end{bmatrix}$
$w = G \uparrow$		$\begin{bmatrix} \frac{1}{m_h} (-k_2(q, \dot{q})) - g \\ 0 \end{bmatrix}$
$w = A \uparrow$		$\begin{bmatrix} \frac{1}{m_h} (-k_2(q, \dot{q})) - g \\ \frac{1}{m_t} (k_2(q, \dot{q})) - g \end{bmatrix}$

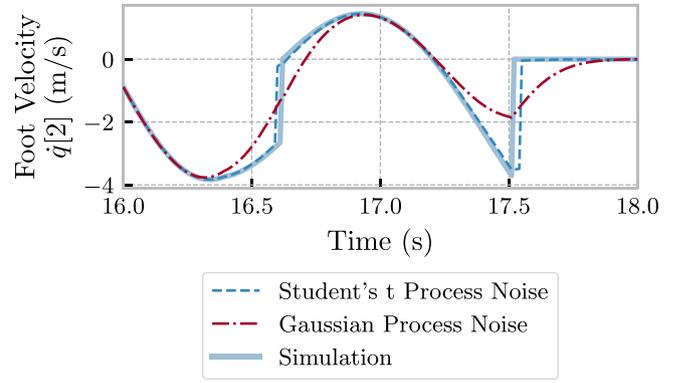


Fig. 4. The Student's t distribution process noise yields better estimates of instantaneous changes in continuous state (Section 6.2.1). In this plot, estimates of the foot velocity are shown near two impacts (≈ 16.6 s, 17.5 s).

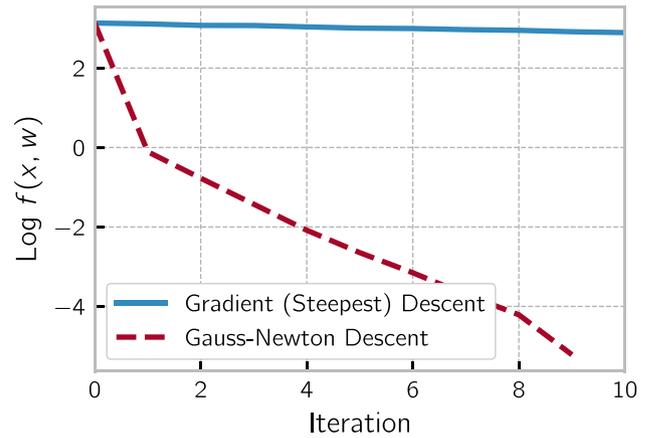


Fig. 5. Gauss–Newton descent directions yield faster convergence than gradient (steepest) descent (Section 6.2.2). In this plot, the discrete state variables w are given and the second line of Algorithm 1 is modified to use either Gauss–Newton descent directions or gradient (steepest) descent to estimate the continuous state variables x by minimizing the relaxed objective function $f(x, w)$ (6).

6.2. Piecewise-linear impact oscillator experiment

In this subsection, we employ the linear spring laws

$$k_1(q, \dot{q}) = 10(q[1] - q[2]) - 3,$$

and

$$k_2(q, \dot{q}) = 15(q[1] - q[2]) - 3,$$

with parameter values $m_h = 3$, $m_t = 1$, $g = 2$.

In our first demonstration the observed states are $q[1]$ and $q[2]$, position of the hip and foot, leaving the velocities unobserved:

$$\mathcal{H}_{\text{pos}}(x) = q. \quad (14)$$

State estimation results for this system are shown in Fig. 7.

In the remainder of this subsection, we demonstrate the effects of the choices we made in our problem formulation (Section 2) and algorithm derivation (Section 3) using the piecewise-linear model as a running example. We also consider a variation where the measurements correspond to the leg length and velocity, which are more representative of the *onboard* measurements available to an autonomous robot operating outside of the laboratory.

6.2.1. Student's t versus Gaussian process noise

Fig. 4 compares the estimation of foot velocity using Student's t with $r = 0.01$ versus using Gaussian for the process noise distribution; in both cases the true discrete state is given. The estimated trajectory for both distributions matches the true simulated trajectory away from jumps, while near jumps, such as around times 16.6s and 17.5s, using the Student's t distribution enables closer tracking of the instantaneous change in the true foot velocity $\dot{q}[2]$ than when using a Gaussian distribution.

6.2.2. Gauss–Newton versus gradient (steepest) descent

We empirically compared convergence rates for continuous state x_t updates obtained using Gauss–Newton and gradient (steepest) descent directions (Algorithm 1, line 2). Fig. 5 shows the log loss versus algorithm iteration for the two methods; the actual discrete state w_t was taken as given to perform this comparison. As expected, the objective value decreases significantly faster when the search direction is determined by the Gauss–Newton scheme as compared to the direction of steepest descent, reaching the stopping criterion in ten times fewer iterations in our tests.

6.2.3. Smoothing the relaxed discrete state versus not

If the continuous states are given, the discrete state estimate returned by our algorithm (skipping lines 2 and 3 of Algorithm 1) is very close to the true discrete state regardless of whether a smoothing term is included in the relaxed problem formulation. When simultaneously estimating both the continuous and discrete states, the smoothing term becomes crucial, as illustrated by comparing the discrete state estimates (\hat{w}_t) in Fig. 6 (without smoothing) and Fig. 7 (with smoothing). In particular, the estimated discrete state switches rapidly without smoothing, whereas with smoothing the discrete state tends to remain constant for many samples and change mostly near ground-truth switching times.

6.2.4. Onboard versus offboard measurements

In the laboratory, the positions of the robot hip and foot can be directly measured *offboard*, e.g. with an external camera system. Outside of the laboratory, only the relative position of the hip and foot can be directly measured *onboard* our robot. Thus, we are motivated by this practical consideration to evaluate our algorithm's performance in the case where only the relative position and velocity of the hip and foot are measured,

$$\mathcal{H}_{\text{relative}}(x) = \begin{bmatrix} q[1] - q[2] \\ \dot{q}[1] - \dot{q}[2] \end{bmatrix}. \quad (15)$$

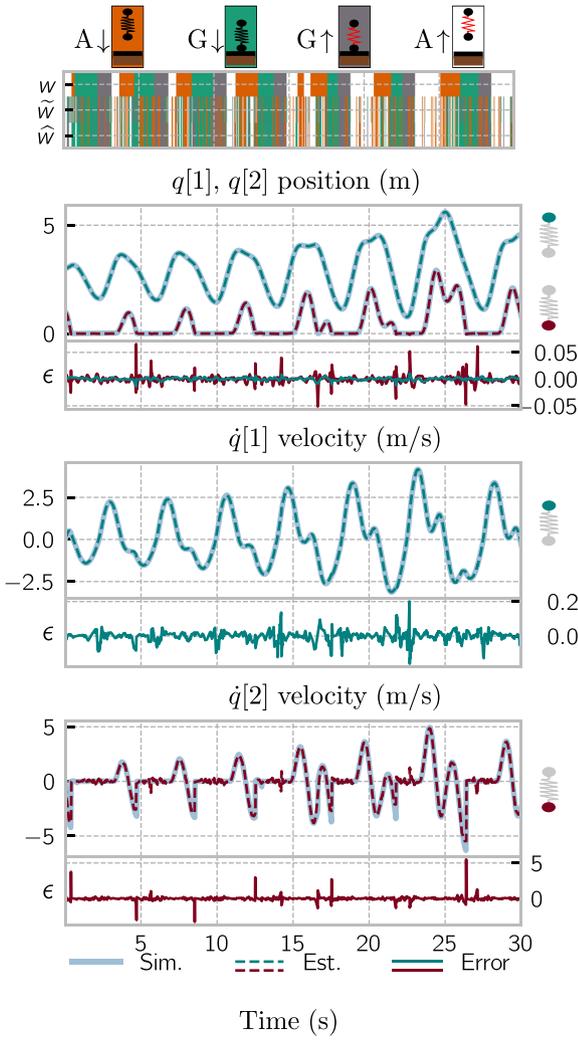


Fig. 6. Without smoothing ($\nu = 0$), the discrete state estimate switches frequently (Section 6.2.3). The top plot shows the true discrete state of the system $w \in D^M$, the relaxed discrete state estimate $\hat{w} \in \Delta^M$, and the discrete state estimate $\tilde{w} \in D^M$ for a simulation of the piecewise-linear system. The subsequent plots show the estimate, simulation, and error ϵ values for position and velocity of the hip $q[1]$ and foot $q[2]$.

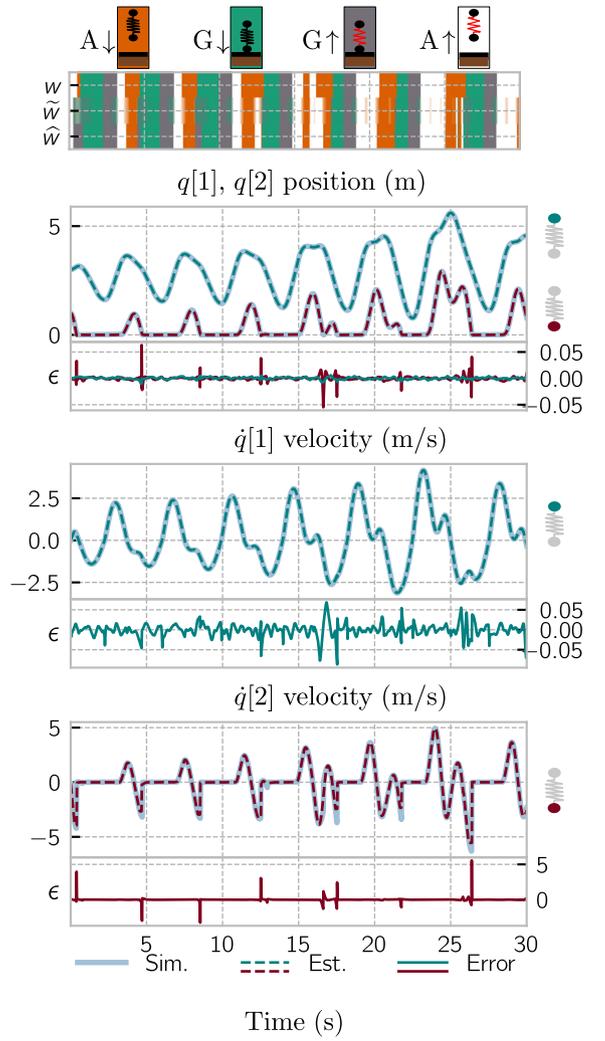


Fig. 7. With smoothing ($\nu > 0$), the discrete state estimate mostly switches near the true switching times (Section 6.2.3). This plot shows results from the piecewise-linear system; the notational and plotting conventions are adopted from Fig. 6.

Although the full hybrid system state is formally unobservable with these relative measurements, our algorithm nevertheless yields good estimates of the discrete state as shown in Fig. 8; due to large errors in the estimate of (unobservable) continuous states, we omit those results from the figure.

6.3. Piecewise-nonlinear impact oscillator experiment

To test Algorithm 1 on a nonlinear model, we included the kinematic constraints depicted in Fig. 3b, resulting in a nonlinear spring force. In this model we set the two spring laws to be the same $k_1 = k_2$, decreasing the number of discrete states from four to two: $w = A$ when $q[2] > 0$ and $w = G$ when $q[2] = 0$. State estimation results compare favorably with the analogous results from the piecewise-linear system when using either absolute position measurements \mathcal{H}_{pos} (14) (compare Fig. 9 with Fig. 7) or relative measurements $\mathcal{H}_{\text{relative}}$ (15) (compare Fig. 10 with Fig. 8).

In Fig. 9 we see that the model can estimate continuous and discrete states in the nonlinear setting. However, we do notice that the estimated trajectories are not as close to ground truth as in the linear case. In particular, when $q[2]$ has a value only slightly

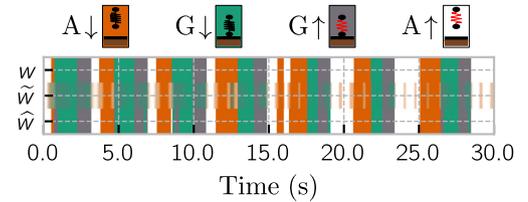


Fig. 8. Estimated discrete state using onboard (relative position and velocity) measurements $\mathcal{H}_{\text{relative}}$ (15) for the piecewise-linear system closely matches true discrete state. (Section 6.2.4). Continuous state estimates are not shown since they are formally unobservable using only onboard measurements (in practice, they drift away from ground truth over time).

greater than 0 (e.g. between times 3 s and 4 s), the algorithm fails to detect the transition between $w = A$ and $w = G$.

7. Conclusion

We proposed a new state estimation algorithm for hybrid systems, analyzed its convergence properties, compared with IMM, and evaluated its performance on piecewise-linear and -nonlinear

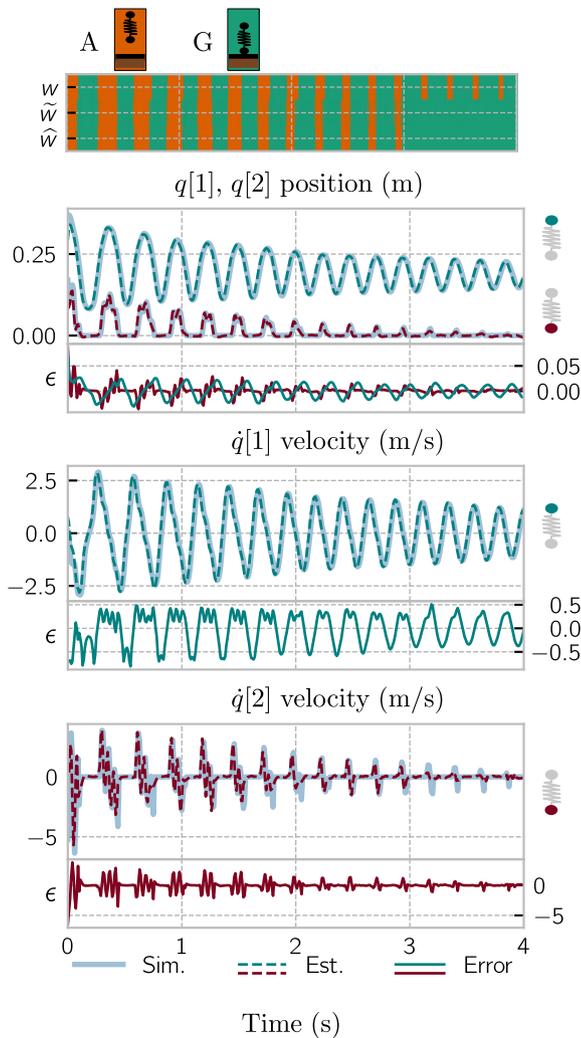


Fig. 9. Continuous and discrete states estimated for the piecewise-nonlinear model (Section 6.3). Notational and plotting conventions are adopted from Fig. 6; note that this model only has two discrete states (Section 6.1).

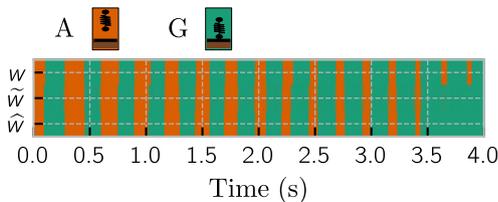


Fig. 10. Estimated discrete state using onboard (relative position and velocity) measurements $\mathcal{H}_{\text{relative}}$ (15) for the piecewise-nonlinear system closely matches true discrete state. (Section 6.2.4). As with Fig. 8, continuous state estimates are not shown since they drift from the true values over time; note that this nonlinear model only has two discrete states (Section 6.1).

hybrid systems with non-identity resets. The algorithm leverages a relaxed state estimation problem formulation where the decision variables corresponding to the discrete state are allowed to take on continuous values. This relaxation yields a continuous optimization problem that can be solved using recently-developed nonsmooth variable projection techniques. The effectiveness of the approach was demonstrated on hybrid system models of mechanical systems undergoing impact.

Appendix. Switched and hybrid dynamical systems

A hybrid dynamical system is a tuple $H = (D, F, G, R)$ (Burden, Revzen, & Sastry, 2015; Goebel et al., 2009) where

$$D = \bigsqcup_{j \in J} D_j, \quad F : D \rightarrow TD, \quad G \subset D, \quad R : G \rightarrow D.$$

With $\phi : [0, \infty) \times D \rightarrow D$ the flow of H , then a discrete-time switched nonlinear system is obtained by sampling H with timestep $\Delta > 0$:

$$x^+ = \phi(\Delta, x).$$

This equation may not immediately appear to be “switched”, but the function ϕ is only piecewise-continuous; the switching structure can be exposed with reference to the flows $\phi_j : [0, \infty) \times D_j \rightarrow D_j$ and time-to-guard $\tau_j : D_j \rightarrow [0, \infty)$, $\tau_{j,k} : D_j \rightarrow [0, \infty)$ functions associated with each discrete state $j \in J$ and pair of discrete states $(j, k) \in J \times J$:

$$x^+ = \begin{cases} \phi_j(\Delta, x), & \tau_j(x) > \Delta; \\ \phi_k(\Delta - \tau_{j,k}(x), R_{j,k}(\phi_j(\tau_{j,k}(x), x))), & \tau_j(x) = \tau_{j,k}(x) \leq \Delta. \end{cases}$$

This piecewise-defined equation, equivalent to (but much more explicit than) $x^+ = \phi(\Delta, x)$, is a discrete-time switched nonlinear system (in particular, each function in the piecewise definition is continuously differentiable) with model set indexed by $M = J \cup (J \times J)$ and switching rule determined as a function of x :

$$m(x) = \begin{cases} j, & \tau_j(x) > \Delta; \\ (j, k), & \tau_j(x) = \tau_{j,k}(x) \leq \Delta. \end{cases}$$

References

- Alessandri, A., Baglietto, M., & Battistelli, G. (2005). Receding-horizon estimation for switching discrete-time linear systems. *IEEE Transactions on Automatic Control*, 50(11), 1736–1748.
- Alessandri, Angelo, Baglietto, Marco, & Battistelli, Giorgio (2007). Minimum-distance receding-horizon state estimation for switching discrete-time linear systems. In Rolf Findeisen, Frank Allgöwer, & Lorenz T. Biegler (Eds.), *Lecture notes in control and information sciences: vol. 358, Assessment and future directions of nonlinear model predictive control* (pp. 348–366). Springer-Verlag Berlin Heidelberg.
- Aravkin, Aleksandr, Burke, James V., Ljung, Lennart, Lozano, Aurelie, & Pilonetto, Gianluigi (2017). Generalized Kalman smoothing: Modeling and algorithms. *Automatica*, 86, 63–86.
- Aravkin, Aleksandr, Burke, James V., & Pilonetto, Gianluigi (2012). Robust and Trend-following Kalman Smoothers using student's T. *IFAC Proceedings Volumes*, 45(16), 1215–1220.
- Aravkin, Aleksandr Y., Burke, James V., & Pilonetto, Gianluigi (2014). Robust and trend-following student's t kalman smoothers. *SIAM Journal on Control and Optimization*, 52(5), 2891–2916.
- Aravkin, Aleksandr, Drusvyatskiy, Dmitriy, & van Leeuwen, Tristan (2016). Variable projection without smoothness. arXiv preprint arXiv:1601.05011.
- Bako, Laurent, & Lecoeuche, Stéphane (2013). A sparse optimization approach to state observer design for switched linear systems. *Systems & Control Letters*, 62(2), 143–151.
- Balluchi, Andrea, Benvenuti, Luca, Di Benedetto, Maria D., & Sangiovanni-Vincentelli, Alberto L. (2002). Design of observers for hybrid systems. In *Proceedings of hybrid systems: computation and control (HSCC), Vol. 2289* (pp. 76–89). Springer Berlin Heidelberg.
- Balluchi, A., Benvenuti, L., Di Benedetto, M. D., & Sangiovanni-Vincentelli, A. L. (2003). Observability for hybrid systems. In *Proceedings of the IEEE conference on decision and control, Vol. 2* (pp. 1159–1164). IEEE.
- Balluchi, Andrea, Benvenuti, Luca, Di Benedetto, Maria D., & Sangiovanni-Vincentelli, Alberto (2013). The design of dynamical observers for hybrid systems: Theory and application to an automotive control problem. *Automatica*, 49(4), 915–925.
- Barhoumi, N., Msahli, F., Djemaï, M., & Busawon, K. (2012). Observer design for some classes of uniformly observable nonlinear hybrid systems. *Nonlinear Analysis. Hybrid Systems*, 6(4), 917–929.
- Beck, Amir, & Teboulle, Marc (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1), 183–202.

- Bemporad, A., Mignone, D., & Morari, M. Moving horizon estimation for hybrid systems and fault detection. In *Proceedings of the American control conference*. Vol. 4 (pp. 2471–2475).
- Blom, Henk A. P., & Bar-Shalom, Yaakov (1988). The interacting multiple model algorithm for systems with Markovian switching coefficients. *IEEE Transactions on Automatic Control*, 33(8).
- Blom, H. A. P., & Bloem, E. A. (2007). Exact Bayesian and particle filtering of stochastic hybrid systems. *IEEE Transactions on Aerospace and Electronic Systems*, 43(1), 55–70.
- Burden, Samuel A., Revzen, Shai, & Sastry, S. Shankar (2015). Model reduction near periodic orbits of hybrid dynamical systems. *IEEE Transactions on Automatic Control*, 60(10), 2626–2639.
- Burke, James V. (1985). Descent methods for composite nondifferentiable optimization problems. *Mathematical Programming*, 33(3), 260–279.
- Di Bernardo, M., Budd, C. J., Champneys, A. R., & Kowalczyk, P. (2008). Piecewise-smooth dynamical systems: Theory and applications. *Applied mathematical sciences: vol. 163*. Springer.
- Doucet, A., Gordon, N. J., & Krishnamurthy, V. (2001). Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3), 613–624.
- Duník, Jindřich, Straka, Ondřej, Kost, Oliver, & Havlík, Jindřich (2017). Noise covariance matrices in state-space models: A survey and comparison of estimation methods—Part I. *International Journal of Adaptive Control and Signal Processing*, 31(11), 1505–1543.
- Fazeli, Nima, Zapolsky, Samuel, Drumwright, Evan, & Rodriguez, Alberto (2017). Learning data-efficient rigid-body contact models: Case study of planar impact. In *Proceedings of the conference on robot learning* (p. 10).
- Ferrari-Trecate, G., Mignone, D., & Morari, M. (2002). Moving horizon estimation for hybrid systems. *IEEE Transactions on Automatic Control*, 47(10), 1663–1676.
- Ferrari-Trecate, G., Mignone, D., & Morari, M. (2002). Moving horizon estimation for hybrid systems. *IEEE Transactions on Automatic Control*, 47(10), 1663–1676.
- Goebel, R., Sanfelice, R. G., & Teel, A. (2009). Hybrid dynamical systems. *IEEE Control Systems*, 29(2), 28–93.
- Gómez-Gutiérrez, D., Čelikovský, S., Ramírez-Treviño, A., Ruiz-Léon, J., & Genaro, S. Di (2011). Sliding mode observer for switched linear systems. In *IEEE international conference on automation science and engineering* (pp. 725–730).
- Hespanha, J. P., & Morse, A. S. (1999). Stability of switched systems with average dwell-time. In *Proceedings of the IEEE conference on decision and control*, Vol. 3 (pp. 2655–2660).
- Johnson, Scott C. (2016). *Observability and observer design for switched linear systems*. (PhD thesis), Purdue University.
- Johnson, Aaron M., Burden, Samuel A., & Koditschek, Daniel E. (2016). A hybrid systems model for simple manipulation and self-manipulation systems. *International Journal of Robotics Research*, 35(11), 1354–1392.
- Kenneally, Gavin, De, Avik, & Koditschek, D. E. (2016). Design principles for a family of direct-drive legged robots. *IEEE Robotics and Automation Letters*, 1(2), 900–907.
- Labbe, Roger (2014). Filterpy. [Online].
- Lötstedt, Per (1982). Mechanical systems of rigid bodies subject to unilateral constraints. *SIAM Journal of Applied Mathematics*, 42(2), 281–296.
- Menini, L., & Tornambe, A. (2001). Asymptotic tracking of periodic trajectories for a simple mechanical system subject to nonsmooth impacts. *IEEE Transactions on Automatic Control*, 46(7), 1122–1126.
- Nocedal, Jorge, & Wright, Stephen J. (2006). *Nonlinear equations*. Springer.
- Rockafellar, R. Tyrrell, & Wets, Roger J.-B. (1998). *Variational analysis*. Heidelberg, Berlin, New York: Springer Verlag.
- Schatzman, M. (1998). Uniqueness and continuous dependence on data for one-dimensional impact problems. *Mathematical and Computer Modelling*, 28(4–8), 1–18.
- Seah, C. E., & Hwang, I. (2009). State estimation for stochastic linear hybrid systems with continuous-state-dependent transitions: An IMM approach. *IEEE Transactions on Aerospace and Electronic Systems*, 45(1), 376–392.

Stengel, Robert R. (1994). *Optimal control and estimation*. Dover.

Vidal, René, Chiiso, Alessandro, Soatto, Stefano, & Sastry, Shankar (2003). Observability of linear hybrid systems. In Oded Maler, & Amir Pnueli (Eds.), *Hybrid systems: computation and control*, Vol. 2623 (pp. 526–539). Springer Berlin Heidelberg.



Jize Zhang is a PhD student in Applied Mathematics at the University of Washington. She obtained her BS from Smith College in 2014. She works on optimization research with applications to multiple fields, including finance, engineering, and global health.



Andrew M. Pace earned his B.S. in Engineering from Swarthmore College in 2009. He received his M.S. in Electrical Engineering from the Colorado School of Mines in 2012 where his research focused on the control of wind turbines. He is now a PhD candidate in Electrical Engineering at the University of Washington and works on the application of hybrid dynamical systems to legged robotic locomotion and other areas.



Sam A. Burden earned his BS with Honors in Electrical Engineering from the University of Washington in Seattle in 2008. He earned his PhD in Electrical Engineering and Computer Sciences from the University of California in Berkeley in 2014, where he subsequently spent one year as a Postdoctoral Scholar. In 2015, he returned to UW EE (now ECE) as an Assistant Professor; in 2016, he received a Young Investigator award from the Army Research Office (ARO-YIP). Sam is broadly interested in discovering and formalizing principles of sensorimotor control. Specifically, he focuses on applications in dynamic and dexterous robotics, neuromechanical motor control, and human-cyber-physical systems. In his spare time, he teaches robotics to students of all ages in classrooms and campus events.



Aleksandr Aravkin received B.S. degrees in Mathematics and Computer Science from the University of Washington in 2004. He then received an M.S. in Statistics and a PhD in Mathematics from the University of Washington in 2010. He was a joint postdoctoral fellow in Earth and Ocean Sciences and Computer Science at the University of British Columbia from 2010–2012, and a research staff member at the IBM T.J. Watson Research Center from 2012–2015. During this time he also worked at Columbia as an Adjunct Professor in Computer Science and IEOR. In 2015, Dr. Aravkin joined the faculty at UW Applied Mathematics, where he works on theoretical and practical problems connected to data science, including convex and variational analysis, statistical modeling, and algorithm design.