

Adaptive Piecewise–Affine Inverse Modeling of Hybrid Dynamical Systems^{*}

Ehsan Elhamifar^{*} Samuel A. Burden^{*} S. Shankar Sastry^{*}

^{*} *Department of Electrical Engineering and Computer Sciences,
University of California, Berkeley, CA 94720 USA
(e-mail: ehsan,sburden,sastry@eecs.berkeley.edu).*

Abstract: Motivated by the study of complex motor control systems, we consider the identification and control of PieceWise Affine (PWA) systems and propose a novel data-driven framework that adaptively inverts the dynamics of such systems using noisy sampled data. First, we propose a novel PWA identification algorithm based on convex optimization applicable to both state–space and input/output models. Instead of considering the non–convex problem of searching over the entire parameter space of all submodels while assigning data to submodels, we propose to estimate a set of candidate submodels first and then select the best few submodels that best explain the given observations. Unlike the state–of–the–art clustering–based PWA identification algorithms, our method is robust to outliers, i.e., outlying submodel estimates will not affect the result of our proposed scheme. Given a PWA model of the dynamics obtained from the identification algorithm, we consider the control of the resulting hybrid system where our goal is to find an input that reproduces a given reference trajectory or that extremizes a performance criterion. To overcome computational limitations of state–of–the–art solutions to control of PWA systems, we cast the two problems of trajectory tracking and performance extremization in an optimization framework, and seek a scalable numerical algorithm. We demonstrate the effectiveness of our proposed framework on a model of a jumping robot.

Keywords: hybrid systems, piecewise affine, system identification, optimal control, inverse dynamic problem, model approximation, physical models

1. INTRODUCTION

To move through and interact with its environment, a *complex motor control system* (e.g. an animal or robot) must intermittently contact objects or terrain. The dynamics of such systems are complicated, being generally high–dimensional due to the presence of many mechanical degrees–of–freedom and piecewise–defined (or *hybrid*) due to impact mechanics. Though careful investigations into the design of such control systems has been undertaken at least since Bernstein (1967), it remains intractable to study their behavior analytically. We propose a data–driven framework for the identification and control of hybrid systems that adaptively inverts the dynamics of such systems using noisy sampled data. Our procedure is based on a novel identification method and novel extensions to established control algorithms, which help to accommodate the discontinuous nature of the dynamics and deal with real-world challenges, such as the presence of corrupted observations.

Hybrid systems, which are heterogeneous dynamical systems arising from the interaction of continuous and discrete dynamics, have attracted increasing attention, due to their rich modeling capabilities. An important class of hybrid systems are PieceWise Affine (PWA) models,

which consist of affine submodels between which the system switches as a consequence of the state/input or the input/output signal (Sontag (1981)). PWA models can describe real phenomena whose discrete dynamics arise from logic devices or from different phases of a process/activity, such as footfall during walking. Moreover, PWA models can be used to approximate nonlinear dynamical systems with arbitrary accuracy. In fact, a broad class of hybrid systems admit a PWA description, since PWA models provide an equivalent representation for Mixed Logic Dynamical (MLD) models (Bemporad and Morari (1999)), Linear Complementarity (LC) models (Heemels and J. Schumacher (2000); der Schaf and Schumacher (1998)), Extended Linear Complementarity (ELC) models (De Schutter (2000)) and Max-Min-Plus-Scaling (MMPS) models (De Schutter and Van den Boom (2001)).

1.1 Identification of PWA models

In order to perform analysis, verification and control of PWA systems, one needs to have a model for the process at hand. However, it is often extremely difficult or even impossible to derive models using first modeling principles. Identification techniques provide effective tools for obtaining such models from experimental observations.

The identification of PWA models can be performed in the input/output or the state–space form. However, the majority of the literature have addressed the identification in the case of the input/output form. While one can often find a

^{*} S. Burden supported by a Graduate Research Fellowship from the US NSF. This work was partially supported by US ARL Agreement W911NF–08–2–0004.

state–space realization from the input/output model, care must be taken to determine whether such a state–space realization exists. Moreover, as studied by Verdult and Verhaegen (2004), since the state–space parameters are always recovered up to an arbitrary change of basis, one needs to determine the appropriate basis for each identified submodel in the state–space form, in order to have continuous state transitions along the switching boundaries.

Ferrari-Trecate et al. (2003); Ferrari-Trecate and Schinkel (2003) proposed an identification algorithm by clustering parameter vectors, each of which is locally estimated using the nearest neighbors of each data point. Ragot et al. (2003) proposed a method for identifying the parameters of submodels by choosing an adapted weighting function, which allows one to select the data for which each submodel is active. Nakada et al. (2005) proposed a method based on statistical clustering of measured data via a Gaussian mixture model and support vector classifiers. Vidal et al. (2003) proposed an algebraic geometric approach, based on polynomial factorization, in order to estimate the parameters and the number of submodels. Roll et al. (2004) reduced the identification problem of a special class of PWA models to a mixed-integer programming problem. Such methods suffer from challenges such as having a nonconvex formulation, being computationally demanding or not being able to deal with outlying estimates.

1.2 Control of PWA models

Given a PWA approximation for the dynamics of a complex motor control system, our goal is to find an input that reproduces a given reference trajectory as in Crawford and Sastry (1995) and Remy and Thelen (2009) or that extremizes a performance criterion as in Srinivasan and Ruina (2005) and Aguilar et al. (2012). Sontag (1981) analytically characterized the solution to such *inverse modeling* problems for PWA systems, but computationally–tractable solutions are lacking. Therefore we cast both problems—trajectory tracking and performance extremization—in an optimization framework, and seek a scalable numerical algorithm.

Previous work on optimal control of hybrid systems offers analytical characterizations and numerical algorithms subject to various restrictions. Sussmann (1999) and Riedinger et al. (2003) derived first–order necessary conditions for local extrema based on the *maximum principle* and Branicky et al. (1998) characterized global optima using quasi-variational inequalities for a general class of hybrid optimal control problems. For discrete–time hybrid systems, Baotic et al. (2006) derived an analytical characterization and numerical algorithm for extremization of piecewise–linear performance indices. Vasudevan et al. (2013) proved consistent approximation of an algorithm for optimization of *switched* systems, i.e. where discrete mode transitions are triggered solely by the control input. Passenberg et al. (2013) derived a two–stage algorithm for continuous–time systems with partitioned state–space that first chooses an optimal discrete mode sequence and subsequently computes the corresponding optimal continuous input. Motivated by the study of complex motor control systems, we extend prior scalable algorithms to allow *autonomous* transitions that introduce discontinuities in the hybrid system execution.

1.3 Paper Contributions

In this paper, we propose robust and efficient algorithms for the identification and control of PWA models. First, in Section 2, we propose a novel PWA identification algorithm based on convex optimization that can deal with both state–space and input/output models. Instead of searching over the entire parameter space of all submodels while assigning data to submodels, which leads to a highly non-convex formulation, we propose to estimate a set of candidate submodels first and then select a few submodels that best explain the given observations. We formulate a convex program to achieve this goal. Thus, unlike the state-of-the-art PWA identification algorithms that are non-convex, rely on greedy suboptimal algorithms, or are initialization dependent, our proposed formulation can find the global optimal solution efficiently. Not only the solution of our algorithm finds a few relevant submodels, but at the same time it provides the assignment of observations to the selected models. Hence, we can find the partitioning of the state–input domain (in the state–space form) or the regressor domain (in the input/output form) using the output of our algorithm. Since we select a few models that can best describe the observations, unlike clustering-based algorithms (Ferrari-Trecate et al. (2003); Nakada et al. (2005)), outlying submodel estimates will not affect our solution. This is particularly important for PWA approximations of physical phenomena obtained from sampled data.

For the control of PWA models, in Section 3, we use a first–order algorithm to approximate local extrema of a performance criterion in a nonlinear programming framework. Although this scheme will generally not yield global optima, its scalability makes it an attractive approach to control of the high–degree–of–freedom mechanical systems characteristic of complex motor control systems.

Finally, in Section 4, we demonstrate the effectiveness of our proposed framework on a mechanical model of an isolated limb intermittently impacting terrain.

2. PIECEWISE–AFFINE IDENTIFICATION

In this section, we propose a robust and efficient algorithm for the identification of PieceWise Affine (PWA) models in both state–space and input/output models, given the input $\{\mathbf{u}(t) \in \mathbb{R}^p\}_{t=1}^T$ and the output $\{\mathbf{y}(t) \in \mathbb{R}^q\}_{t=1}^T$ data.

2.1 State–Space PWA Models

A PWA model in the state–space form can be described as

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A}_{\sigma_t} \mathbf{x}(t) + \mathbf{B}_{\sigma_t} \mathbf{u}(t) + \mathbf{g}_{\sigma_t} + \mathbf{w}(t) \\ \mathbf{y}(t) &= \mathbf{C}_{\sigma_t} \mathbf{x}(t) + \mathbf{D}_{\sigma_t} \mathbf{u}(t) + \mathbf{h}_{\sigma_t} + \mathbf{v}(t), \end{aligned} \quad (1)$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the continuous state of the system at time t . The terms $\mathbf{w}(t) \in \mathbb{R}^n$ and $\mathbf{v}(t) \in \mathbb{R}^q$ represent errors, which arise from process, measurement, or modeling errors. The discrete state σ_t describes the affine dynamics at time t and is assumed to take a finite number of values from $\{1, 2, \dots, s\}$, where s is the number of affine submodels. The real vector/matrices $\{\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i, \mathbf{D}_i, \mathbf{g}_i, \mathbf{h}_i\}$, of appropriate dimensions, represent the parameters of the i -th affine submodel, \mathcal{M}_i . The discrete state of a PWA

model in the state–space form is determined according to the rule

$$\sigma_t = i \quad \text{iff} \quad \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{u}(t) \end{bmatrix} \in \Psi_i, \quad i = 1, 2, \dots, s, \quad (2)$$

where $\{\Psi_i\}_{i=1}^s$ denotes a partitioning of the state–input domain $\Psi \subseteq \mathbb{R}^{n+p}$.

Given the input $\{\mathbf{u}(t)\}_{t=1}^T$ and the output $\{\mathbf{y}(t)\}_{t=1}^T$ data as well as the number of submodels s , our goals are: (a) identify the parameters $\{\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i, \mathbf{D}_i, \mathbf{g}_i, \mathbf{h}_i\}_{i=1}^s$ of submodels; (b) find the discrete state $\sigma_t \in \{1, \dots, s\}$ at each time instant $t = 1, \dots, T$; (c) estimate the partitioning $\{\Psi_i\}_{i=1}^s$ of the state–input domain.

2.2 Input/Output PWA Models

A PWA model in the input/output form can be described as

$$\mathbf{y}(t) = \boldsymbol{\theta}_{\sigma_t}^\top \begin{bmatrix} \mathbf{r}(t) \\ 1 \end{bmatrix} + \mathbf{e}(t), \quad (3)$$

where $\sigma_t \in \{1, 2, \dots, s\}$ denotes the discrete state, s is the number of affine submodels, $\boldsymbol{\theta}_i$ denotes the matrix of parameters of the i -th submodel, \mathcal{M}_i , and $\mathbf{e}(t)$ is the error. The vector $\mathbf{r}(t)$ is called the regressor and is defined as

$$\mathbf{r}(t) = [\mathbf{y}(t-1)^\top \dots \mathbf{y}(t-n_a)^\top \mathbf{u}(t)^\top \dots \mathbf{u}(t-n_b)^\top]^\top, \quad (4)$$

for a given $n_a \geq 1$ and $n_b \geq 0$. Similar to the state–space form, the discrete state in the input/output form can be determined according to the rule

$$\sigma_t = i \quad \text{iff} \quad \mathbf{r} \in \Omega_i, \quad i = 1, 2, \dots, s, \quad (5)$$

where $\{\Omega_i\}_{i=1}^s$ denotes a partitioning of the regressor domain $\Omega \subseteq \mathbb{R}^m$ and $m = q \cdot n_a + p \cdot (n_b + 1)$. A PWA model in the form (3) whose switching mechanism is determined according to (5) is called a PieceWise AutoRegressive eXogenous (PWARX) model.

Given the input $\{\mathbf{u}(t)\}_{t=1}^T$ and the output $\{\mathbf{y}(t)\}_{t=1}^T$ data, the number of submodels s , and the parameters n_a and n_b , our goals are: (a) identify the parameters $\{\boldsymbol{\theta}_i\}_{i=1}^s$ of submodels; (b) find the discrete state $\sigma_t \in \{1, \dots, s\}$ at each time instant $t = 1, \dots, T$; (c) estimate the partitioning $\{\Omega_i\}_{i=1}^s$ of the regressor domain.

2.3 A Convex Formulation for PWA Model Identification

In this section, we formulate a convex optimization for the identification of PWA models $\{\mathcal{M}_i\}_{i=1}^s$ in both state–space and input/output forms. To simplify formulation, we denote the parameters of submodels as $\{\boldsymbol{\beta}_i\}_{i=1}^s$, where $\boldsymbol{\beta}_i = \{\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i, \mathbf{D}_i, \mathbf{g}_i, \mathbf{h}_i\}$ in the state–space model and $\boldsymbol{\beta}_i = \boldsymbol{\theta}_i$ in the input/output model.

Notice that we can write the identification problem as

$$\begin{aligned} \min_{\{\boldsymbol{\beta}_i\}, \{z_{it}\}} & \sum_{t=c}^T \sum_{i=1}^s \ell(\mathbf{y}(t), \mathbf{u}(t); \boldsymbol{\beta}_i) z_{it} \\ \text{s. t.} & \sum_{i=1}^s z_{it} = 1, z_{it} \in \{0, 1\}, \forall i, t \end{aligned} \quad (6)$$

where the loss function $\ell(\mathbf{y}(t), \mathbf{u}(t); \boldsymbol{\beta}_i)$ evaluates how well a submodel \mathcal{M}_i with parameters $\boldsymbol{\beta}_i$ can describe the observations $\{\mathbf{y}(t), \mathbf{u}(t)\}$, while the selection variable z_{it}

enforces that each observation can be represented by only one submodel. The starting time index, c , depends on the order of the model n in the state–space form and on n_a and n_b in the input/output form. In fact, the formulation (6) is non-convex and NP-hard to solve due to $\{\boldsymbol{\beta}_i\}$ and $\{z_{it}\}$ being unknown at the same time as well as binary constraints on the variables z_{it} .

In order to efficiently address the identification problem, instead of searching over the entire parameter space, as in (6), we propose to first find a set of candidate parameters $\{\hat{\boldsymbol{\beta}}_k\}_{k=1}^N$, where $N \geq s$, and then select the best s parameters from this set. Finding a set of candidate parameters can be done by taking each regressor and its few nearest neighbors and fitting a model using the least squares, in the case of the input/output model, or by taking the input/output trajectories at different time intervals and fitting a state–space model to each snippet. To select the best parameter estimates, we consider the following optimization program

$$\begin{aligned} \min_{\{z_{kt}\}} & \sum_{t=c}^T \sum_{k=1}^N \ell(\mathbf{y}(t), \mathbf{u}(t); \hat{\boldsymbol{\beta}}_k) z_{kt} \\ \text{s. t.} & \sum_{k=1}^N z_{kt} = 1, z_{kt} \in \{0, 1\}, \forall k, t \\ & \sum_{k=1}^N \|\mathbf{z}_k\|_\infty \leq s, \end{aligned} \quad (7)$$

where $\mathbf{z}_k \triangleq [z_{kc} \dots z_{kT}]^\top$. Notice that the last constraint in (7) imposes that the optimization can select at most s candidate parameters. This comes from the fact that if the parameter $\hat{\boldsymbol{\beta}}_k$ is selected to describe some of the observations $\{\mathbf{y}(t), \mathbf{u}(t)\}$ for some $t \in [c, T]$, then $\|\mathbf{z}_k\|_\infty = 1$, since the nonzero variables z_{kt} will be 1 by the binary constraint. Otherwise, if $\hat{\boldsymbol{\beta}}_k$ is not selected, then $\|\mathbf{z}_k\|_\infty = 0$. Hence, the sum $\sum_{k=1}^N \|\mathbf{z}_k\|_\infty$ corresponds to the number of selected submodels. As a result, the constraints together impose that we select at most s parameters from $\{\hat{\boldsymbol{\beta}}_k\}_{k=1}^N$ while each observation $\{\mathbf{y}(t), \mathbf{u}(t)\}$ is associated with only one of the s selected submodels.

Notice that unlike (6) that searches over the entire parameter space and the binary selection variables at the same time, the formulation in (7) only searches over the selection variables. However, in order to do so, we have increased the number of selection variables from $s \cdot (T - c + 1)$ to $N \cdot (T - c + 1)$. However, the optimization program (7) is still non-convex due to the binary constraints on z_{kt} . Thus, we consider the relaxation $z_{kt} \in [0, 1]$ and propose to solve the following convex program

$$\begin{aligned} \min_{\{z_{kt}\}} & \sum_{t=c}^T \sum_{k=1}^N \ell(\mathbf{y}(t), \mathbf{u}(t); \hat{\boldsymbol{\beta}}_k) z_{kt} \\ \text{s. t.} & \sum_{k=1}^N z_{kt} = 1, z_{kt} \geq 0, \forall k, t \\ & \sum_{k=1}^N \|\mathbf{z}_k\|_\infty \leq s. \end{aligned} \quad (8)$$

Remark 1. Even with the relaxation $z_{it} \in [0, 1]$, the optimization program (6) remains nonconvex, due to both

$\{\beta_i\}$ and $\{z_{it}\}$ being unknown, hence gets trapped in local minima. Moreover, only certain loss functions lead to a smooth optimization. On the other hand, our proposed formulation in (8) is convex and can be applied to any arbitrary loss function, since, in our case, the value of the loss function is known prior to solving the optimization.

Once we solve the above optimization, we can determine the selected parameters by finding the nonzero vectors z_k . Furthermore, we can find the discrete state σ_t by assigning the observation at time t to its associated selected parameter. In other words, if $\{j_1, \dots, j_s\}$ denote the set of indices of the selected parameters $\{\hat{\beta}_k\}_{k=1}^N$, then we find the discrete state from

$$\sigma_t = \operatorname{argmax}_i z_{jit}. \quad (9)$$

Once we cluster the regressors or the state–input vectors according to their memberships to different submodels, we use a (multi-category) SVM classifier in order to find the partitioning of the domain into different subregions corresponding to different submodels.

3. PIECEWISE–AFFINE CONTROL

We seek solutions to two classes of optimization problems for complex motor control systems: choosing inputs that reproduce reference trajectories as in Crawford and Sastry (1995) and Remy and Thelen (2009); and finding inputs that extremize a performance criterion as in Srinivasan and Ruina (2005) and Aguilar et al. (2012). The underlying physical phenomena evolve in continuous–time, with the transitions between discrete modes triggered (for instance) when appendages establish or break contact with the terrain. Hence, it is natural to compute control inputs that extremize performance in a continuous–time model. However, since we aim to estimate models from noisy sampled data, the PWA approximation for the phenomena will generally be discrete–time, in either the state–space form of Section 2.1 or the input/output (I/O) form of Section 2.2. Therefore we begin by transforming a discrete–time PWA model to a general continuous–time form suitable for control, and subsequently derive a first–order algorithm to compute locally optimal inputs for the model using nonlinear programming.

3.1 Continuous–Time Hybrid Dynamical Systems

The discrete–time state–space PWA model of Section 2.1 is generally written over a partition $\{\Psi_i\}_{i=1}^s$ of state–input space $\mathbb{R}^n \times \mathbb{R}^p = \mathbb{R}^{n+p}$. For the control scheme considered in this section, we restrict to partitions determined entirely by the state, and hence write $\Psi_i = D_i \times \mathbb{R}^p$ for each $i \in S = \{1, \dots, s\}$. We then construct a hybrid dynamical system over the finite disjoint union $D = \coprod_{i \in S} D_i$ where D_i is a connected smooth manifold with boundary for each $i \in S$. We endow D with the natural (piecewise–defined) topology and smooth structure and refer to such spaces as *smooth hybrid manifolds* (see Burden et al. (2012) for more details).

Definition 1. A *hybrid dynamical system* is specified by a tuple $H = (D, F, G, R)$ where:

- $D = \coprod_{i \in S} D_i$ is a smooth hybrid manifold;
- $F : D \rightarrow TD$ is a smooth hybrid vector field;

$G \subset \partial D$ is an open subset of ∂D ;

$R : G \rightarrow D$ is a smooth hybrid map.

For a PWA model, $F|_{D_i}$ is affine for each $i \in S$. Note that, in principle, we allow an arbitrary choice of coordinates in each state domain D_i ; this is advantageous since the principal axes returned from a state–space identification algorithm may differ among discrete modes, necessitating discontinuous state update during mode transition.

Roughly speaking, an *execution* of a hybrid dynamical system is determined from an initial condition in D by following the continuous–time dynamics determined by the vector field F until the trajectory reaches the guard G , at which point the reset map R is applied to obtain a new initial condition. If F is tangent to G at $x \in G$, there is a possible ambiguity in determining a trajectory from x since one may either follow the flow of F on D or apply the reset map to obtain a new initial condition $y = R(x)$.

Assumption 1. F is nowhere tangent to G .

We formalize the definition of an execution using the notion of a *hybrid time trajectory* from Lygeros et al. (2003); we emphasize the natural (disjoint–union) topology to simplify the subsequent definitions of *execution* and *costate*.

Definition 2. A *hybrid time trajectory* is a disjoint union of intervals $\tau = \coprod_{i=1}^N \tau_i$ such that:

- (1) $N \in \mathbb{N} \cup \{\infty\}$;
- (2) $\tau_i \subset \mathbb{R}$ is an interval for all $i < N$, and if $N < \infty$ then $\tau_N \subset \mathbb{R}$ is also an interval;
- (3) $\tau_i \cap \tau_{i+1} = \{t_i\}$ for all $i < N$ (i.e. $\tau_i \cap \tau_{i+1}$ is nonempty and consists of a single element).

Definition 3. An *execution* of a hybrid dynamical system (D, F, G, R) is a smooth hybrid map $x : \tau \rightarrow D$ defined over a hybrid time trajectory $\tau = \coprod_{i=1}^N \tau_i$ such that:

- (1) $\forall t \in \tau$, in coordinates: $\frac{d}{dt}x(t) = F(x(t))$;
- (2) $\forall i < N$, with $\{t_i\} = \tau_i \cap \tau_{i+1}$, $x^- = x|_{\tau_i}(t_i)$, $x^+ = x|_{\tau_{i+1}}(t_i)$: $x^- \in G$, $R(x^-) = x^+$.

In the next section, we compute the variation of the final state of an execution with respect to the initial condition.

3.2 First–Order Variation of Final State Cost

We assume that a performance criteria $J : D \rightarrow \mathbb{R}$ has been given that we wish to extremize at a specified final time¹. Without loss of generality, we assume that J is a *cost* that we aim to minimize. At a local minimum, the derivative $D_\xi J$ of J with respect to the initial condition $\xi \in D$ will be zero (see Sussmann (1999)). Instead we directly compute the derivative of J by propagating a *costate* backward in time along x .

Definition 4. A *costate* associated with an execution $x : \tau \rightarrow D$ of a hybrid dynamical system (D, F, G, R) is a smooth hybrid map $\lambda : \tau \rightarrow T^*D$ defined over the hybrid time trajectory $\tau = \coprod_{i=1}^N \tau_i$ such that:

- (1) $\forall t \in \tau$, in coordinates: $\frac{d}{dt}\lambda(t) = \lambda(t)D_x F(x(t))$;

¹ If a running cost $L : D \rightarrow \mathbb{R}$ is provided as well, we augment the hybrid system with an additional state variable $z \in \mathbb{R}$ with dynamics $\dot{z} = L(x)$ and consider the final cost $\tilde{J}((x, z)) = J(x) + z$.

- (2) $\forall i < N$, in coordinates where² F is orthogonal to G , with $\{t_i\} = \tau_i \cap \tau_{i+1}$, $x^- = x|_{\tau_i}(t_i)$, $x^+ = x|_{\tau_{i+1}}(t_i)$, $\lambda^- = \lambda|_{\tau_i}(t_i)$, and $\lambda^+ = \lambda|_{\tau_{i+1}}(t_i)$:

$$\lambda^- = \lambda^+ D_x R(x^-) + F(x^-) \frac{\lambda^+ F(x^+)}{\|F(x^-)\|^2}.$$

If $s = \sup \tau < \infty$, then by initializing a costate with the final value $\lambda(s) = dJ(x(s)) \in T_{x(s)}^* D$, the initial value $\lambda(0) \in T_{x(0)}^* D$ is the derivative $D_{x(0)} J$ of the final cost with respect to the initial state $x(0)$. This enables us in the next section to develop a scalable algorithm that iteratively approximates local minima of the final cost.

3.3 Optimization and Optimal Control

For brevity and clarity, in the discussion above we restricted our attention to optimization of the initial state $\xi \in D$ of the hybrid system $H = (D, F, G, R)$. In practice, only a subset of initial states are available for optimization, and there may be a *parameter* θ that must be selected from within a set Θ as well. So long as the continuous and discrete dynamics depend smoothly on the parameter θ that resides in a finite-dimensional smooth boundaryless manifold Θ , then we may augment the state of H with the parameter θ to obtain a new hybrid dynamical system $H_\Theta = (D \times \Theta, F_\Theta, G \times \Theta, R_\Theta)$ where

$$F_\Theta(x, \theta) = (F(x; \theta), 0), \quad R_\Theta(x, \theta) = (R(x; \theta), \theta).$$

Note in particular that the parameters may specify a combination of application-appropriate (smooth) basis functions that determine an *open-loop input* for the system. Optimization of a finitely-parameterized smooth control input for H is equivalent to optimization of the initial condition for H_Θ .

Given the derivative $D_\xi J$ of the final cost $J : D \rightarrow \mathbb{R}$ with respect to the initial condition ξ from the previous section, we apply gradient descent with an appropriate stepsize selection rule (see Bertsekas (1999) for more details) to iteratively approximate an initial condition that locally minimizes J , terminating when $\|D_\xi J\|$ drops below a prespecified threshold. In practice, we use the L-BFGS-B algorithm of Zhu et al. (1997) included in the SciPy optimization suite developed by Jones et al. (2001–). The sourcecode used to generate the results in this paper will be made available online upon publication of this paper.

4. EXPERIMENTS

Motivated by complex motor control systems, in this section we identify dynamics and synthesize optimal control inputs for a jumping robot using experimental data. Specifically, we consider the model shown in Figure 1 originally developed to approximate a physical jumping robot built by Aguilar et al. (2012). We believe this low-dimensional example provides a good benchmark for PWA identification and control since the underlying physical phenomena is well-approximated with a PWA model. The system evolves through an *aerial* mode D_a and a *ground* mode D_g :

² Such coordinates exist since F is nowhere tangent to G by Assumption 1.

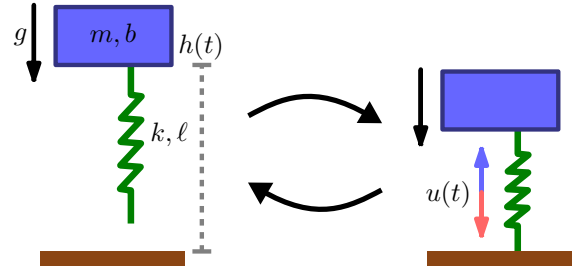


Fig. 1. Illustration of the jumping robot from Section 4. *Left*: mass moves vertically in a gravitational field. *Right*: actuator and leg spring exert forces on the mass when the leg spring is in contact with the ground.

$$D_a = \left\{ (h, \dot{h}) \in T\mathbb{R} : h \geq \ell \right\},$$

$$D_g = \left\{ (h, \dot{h}) \in T\mathbb{R} : h \leq \ell \right\},$$

where ℓ is the nominal leg spring length. The continuous-time dynamics are given by:

$$F|_{D_a} : m\ddot{h} = +u - mg,$$

$$F|_{D_g} : m\ddot{h} = +u - mg - b\dot{h} - k(h - \ell),$$

where m is the hopper mass, g is the acceleration due to gravity, b is the damping coefficient, and k is the leg spring stiffness. Discrete transitions are triggered when the leg is at its rest length:

$$G_a = \left\{ (\ell, \dot{h}) \in \partial D_a : \dot{h} < 0 \right\}, \quad R|_{G_a}(\ell, \dot{h}) = (\ell, c\dot{h}),$$

$$G_g = \left\{ (\ell, \dot{h}) \in \partial D_g : \dot{h} > 0 \right\}, \quad R|_{G_g}(\ell, \dot{h}) = (\ell, \dot{h}),$$

where $c \in [0, 1]$ is the coefficient of restitution of impact. Together, this determines a hybrid system (D, F, G, R) :

$$D = D_a \amalg D_g, \quad F : D \rightarrow TD, \quad G = G_a \amalg G_g, \quad R : G \rightarrow D.$$

We also use the same parameters as in Aguilar et al. (2012) except that (a) we allow the ground height to be a non-zero parameter (1mm in our experiments); (b) we set the coefficient of restitution of the robot to $c = 1$, corresponding to continuous state-update maps; (c) we set the units for the output height and velocity to be mm and mm/sec, respectively, to improve numerical conditioning of the identification problem. The natural frequency of the model is 11.2Hz, hence it nominally performs a jumping cycle in approximately 90msec. For each trial we record the input/output data and the ground-truth discrete mode (*aerial* and *ground*) at 200Hz sampling rate.

4.1 Identification

Given the input-output observations from multiple experimental trials, we use our proposed identification algorithm in order to learn a PWA discrete-time state-space model for the jumping robot. We set $n = 2$ and $s = 2$ to learn a second-order state-space model for the robot with two discrete modes. In order to find candidate submodels, we take each trial, divide it into overlapping segments of length K and estimate a discrete-time state-space model for each segment using the subspace identification algorithm. We set $K = 20$ in our experiments, for which we obtain reasonable results.

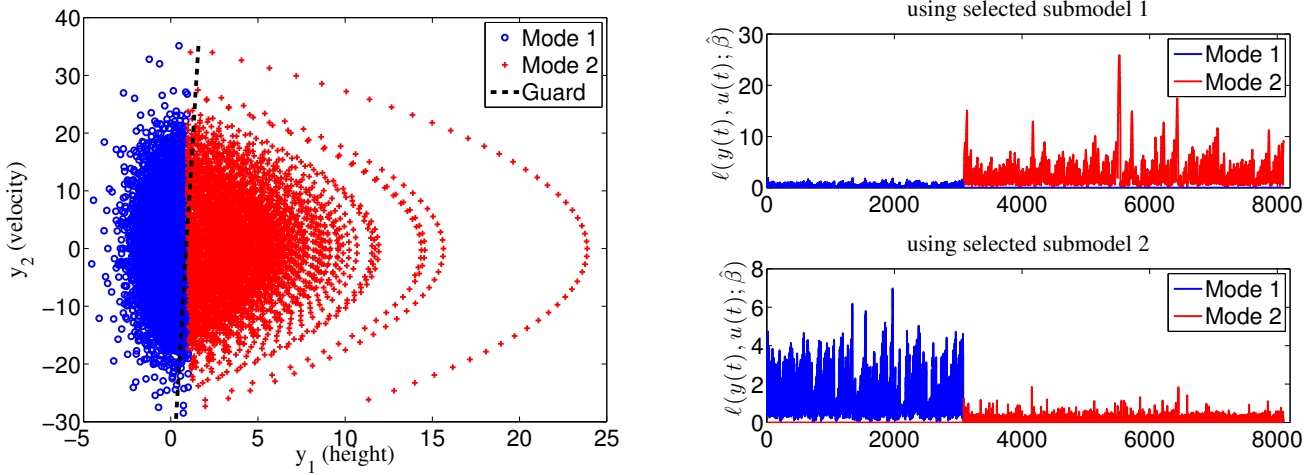


Fig. 2. *Left*: height–velocity graph using the ground-truth discrete modes and the separating hyperplane learned by our proposed algorithm. *Right*: representation error of the data using the two submodels selected by our algorithm from the set of estimated models. The observation indices on the horizontal axis are reordered such that the first 3079 indices correspond to mode 1, while the next 5021 indices correspond to mode 2.

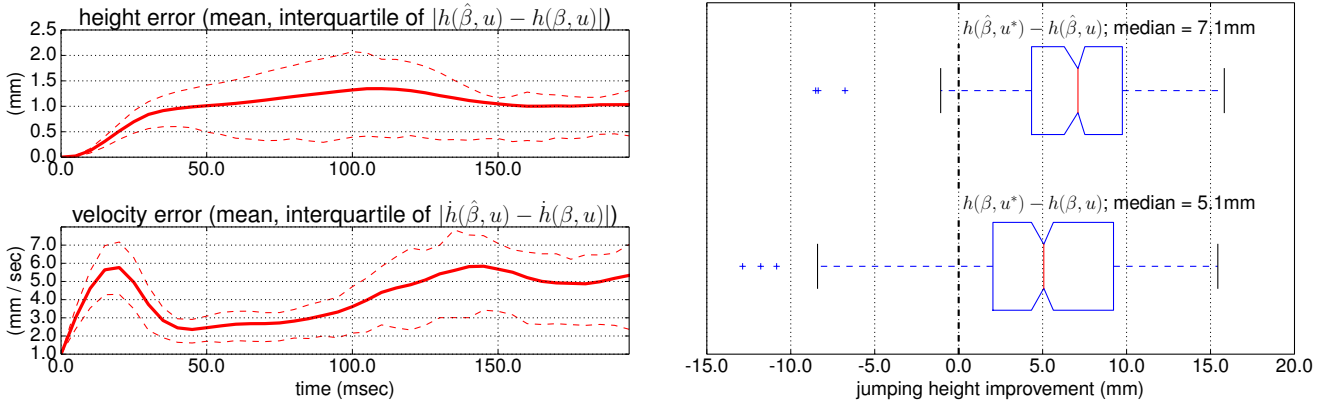


Fig. 3. Control of jumping robot using estimated model. *Left*: prediction error. *Right*: jumping height improvement.

Figure 2 shows the results of our proposed identification algorithm on the dataset. The plot on the left shows the height–velocity graph for the data, where the ground-truth data in mode 1 (ground) and mode 2 (aerial) are shown by blue circles and red pluses, respectively. The separating hyperplane learned by our algorithm is shown by the black dashed line in the graph. Notice that the learned hyperplane almost perfectly separates the data in the two discrete modes. This can also be seen by the fact that the true guard corresponds to the vertical line $y_1 = 1$. Using the ground-truth discrete-state sequence, we also compute the misclassification error, which is 5.56%, corroborating the effectiveness of our proposed method. The plot on the right of Figure 2 shows the representation error of all the observations $\{\mathbf{y}(t), \mathbf{u}(t)\}$ over all trials, using the two estimated submodels selected by our algorithm from the set of estimated models. The indices of observations on the horizontal axis are reordered such that the first 3079 indices correspond to the observations from mode 1, while the next 5021 indices correspond to the observations from mode 2. Notice that the first selected submodel represents the data from mode 1 by a small error while it obtains a large error for the data in mode 2. Similarly, the second selected submodel represents the data from mode 1 by a

large error and obtains a small error for the data in mode 2. Therefore, the selected submodels are reliable models for the two operation modes of the jumping robot.

4.2 Control

Given the state–space model for the jumping robot identified in the previous section, we seek to synthesize a control input that achieves maximum jumping height in the actual system model. Since the estimated model differs from the ground truth, we do not expect the synthesized input to improve jumping height in every trial, therefore we synthesize controls for an ensemble of initial conditions and evaluate the distribution of jumping height outcomes.

We divided the hopper dataset into segments starting at maximum compression in the ground mode; there were 213 such segments in the dataset. For each segment, we simulated both the actual β and estimated $\hat{\beta}$ models for 200msec using the nominal input u used during the identification procedure; the left panel of Figure 3 shows the mean and interquartile range of prediction error for height (*top*) and velocity (*bottom*). We then computed an optimal control input u^* to achieve maximum jumping

height 100msec after maximum compression using the estimated model, and subsequently simulated the actual model with this input; the right panel of Figure 3 shows a boxplot of the improvement achieved for the estimated model (*top*) and actual model (*bottom*) over their nominal simulations. Note that jumping height improves in the actual model in more than 75% of the trials, with a median improvement of 5.1mm.

5. CONCLUSION

In this paper, we proposed a novel framework for identification and control of complex motor control systems modeled by the PieceWise Affine (PWA) subclass of hybrid systems. We first proposed a novel PWA identification algorithm based on convex optimization that, given a set of model estimates, selects a few submodels that best represent the observations together. Using the identified model, we considered the control of PWA systems where we cast the two problems of trajectory tracking and performance extremization in a scalable optimization framework. Our results on the mechanical model of a jumping robot demonstrate the effectiveness of our framework for the study of complex motor control systems. In future work, we aim to apply our identification and control framework to data obtained from real physical systems.

ACKNOWLEDGEMENTS

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute for Government purposes notwithstanding any copyright notation herein.

REFERENCES

- Aguilar, J., Lesov, A., Wiesenfeld, K., and Goldman, D. (2012). Lift-off dynamics in a simple jumping robot. *Physical Review Letters*, 109(174301).
- Baotic, M., Christophersen, F.J., and Morari, M. (2006). Constrained optimal control of hybrid systems with a linear performance index. *IEEE Transactions on Automatic Control*, 51(12), 1903–1919.
- Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.
- Bernstein, N.A. (1967). *The co-ordination and regulation of movements*. Pergamon Press Ltd.
- Bertsekas, D.P. (1999). *Nonlinear Programming*. Athena Scientific, 2nd edition.
- Branicky, M., Borkar, V., and Mitter, S. (1998). A unified framework for hybrid control: Model and optimal control theory. *IEEE Transactions on Automatic Control*, 43(1), 31–45.
- Burden, S.A., Ohlsson, H., and Sastry, S.S. (2012). Parameter identification near periodic orbits of hybrid dynamical systems. In *Proceedings of the IFAC Symposium on System Identification*, 1197–1202.
- Crawford, L.S. and Sastry, S.S. (1995). Biological motor control approaches for a planar diver. In *Proceedings of the IEEE Conference on Decision and Control*, 3881–3886.
- De Schutter, B. (2000). Optimal control of a class of linear hybrid systems with saturation. *SIAM Journal on Control and Optimization*, 39(3).
- De Schutter, B. and Van den Boom, T. (2001). Model predictive control for max-min-plus-scaling systems. *Proceedings of the American Control Conference*, 319–324.
- der Schaf, A.J.V. and Schumacher, J.M. (1998). Complementarity modeling of hybrid systems. *IEEE Transactions on Automatic Control*, 43(4), 483–490.
- Ferrari-Trecate, G., Muselli, M., Liberati, D., and Morari, M. (2003). A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2), 205–217.
- Ferrari-Trecate, G. and Schinkel, M. (2003). Conditions of optimal classification for piecewise affine regression. *Proceedings of the Workshop on Hybrid systems: Computation and Control*, 188–202.
- Heemels, W. and J. Schumacher, S.W. (2000). Linear complementarity systems. *SIAM Journal on Applied Mathematics*, 60(4).
- Jones, E., Oliphant, T., Peterson, P., et al. (2001–). SciPy: Open source scientific tools for Python. URL <http://www.scipy.org/>.
- Lygeros, J., Johansson, K.H., Simic, S.N., Zhang, J., and Sastry, S.S. (2003). Dynamical properties of hybrid automata. *IEEE Transactions on Automatic Control*, 48(1), 2–17.
- Nakada, H., Takaba, K., and Katayama, T. (2005). Identification of piecewise affine systems based on statistical clustering technique. *Automatica*, 41(5), 905–913.
- Passenberg, B., Caines, P., Leibold, M., Stursberg, O., and Buss, M. (2013). Optimal control for hybrid systems with partitioned state space. *IEEE Transactions on Automatic Control*, 58(8), 2131–2136.
- Ragot, J., G, G.M., and Maquin, D. (2003). Parameter estimation of switching piecewise linear systems. *Proceedings of the IEEE Conference on Decision and Control*, 5783–5788.
- Remy, C.D. and Thelen, D.G. (2009). Optimal estimation of dynamically consistent kinematics and kinetics for forward dynamic simulation of gait. *Journal of Biomechanical Engineering*, 131(3), 031005.
- Riedinger, P., Iung, C., and Kratz, F. (2003). An optimal control approach for hybrid systems. *European Journal of Control*, 9(5), 449–458.
- Roll, J., Bemporad, A., and Ljung, L. (2004). Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1), 37–50.
- Sontag, E.D. (1981). Nonlinear regulation: the piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2), 346–358.
- Srinivasan, M. and Ruina, A. (2005). Computer optimization of a minimal biped model discovers walking and running. *Nature*, 439(7072), 72–75.
- Sussmann, H.J. (1999). A maximum principle for hybrid optimal control problems. In *Proceedings of the IEEE Conference on Decision and Control*, 425–430.
- Vasudevan, R., Gonzalez, H., Bajcsy, R., and Sastry, S.S. (2013). Consistent approximations for the optimal control of constrained switched systems – part 1: A conceptual algorithm. *SIAM Journal on Control and Optimization*.
- Verdult, V. and Verhaegen, M. (2004). Subspace identification of piecewise linear systems. *Proceedings of the IEEE Conference on Decision and Control*, 3838–3843.
- Vidal, R., Soatto, S., and Sastry, S.S. (2003). An algebraic geometric approach to the identification of linear hybrid systems. *Proceedings of the IEEE Conference on Decision and Control*, 167–172.
- Zhu, C., Byrd, R.H., Lu, P., and Nocedal, J. (1997). Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software*, 23(4), 550–560.