Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Compression Based Identification of PWA Systems ★

Ichiro Maruta * **Henrik Ohlsson** †

* *Kyoto University, Uji, Kyoto 611-0011 Japan (e-mail:* maruta@i.kyoto-u.ac.jp*).*
† *University of California, Berkeley, CA, USA, and Linköping University, Linköping, Sweden (e-mail:* ohlsson@isy.liu.se*).*

Abstract: Piecewise affine (PWA) models serve as an important class of models for nonlinear systems. The identification of PWA models is known to be a difficult task and often implies solving a non-convex combinatorial optimization problems. In this paper, we revisit a recently proposed PWA identification method. We do this to give a novel derivation of the identification method and to show that under certain conditions, the method is optimal in the sense that it finds the PWA function that passes through the measurements and has the least number of hinges. We also show how the alternating direction method of multipliers (ADMM) can be used to solve the underlying convex optimization problem.

Keywords: Nonlinear system identification; Piecewise affine models; Alternating direction method of multipliers; Nonparametric methods

1. INTRODUCTION

Piecewise affine functions are known for their universal approximation properties (Lin and Unbehauen, 1992; Breiman, 1993) and are therefore popular in control. The identification of PWA is however often difficult. In fact, most PWA methods can be seen as clever approaches for solving a non-convex optimization problem. The underlying non-convex optimization problem varies slightly depending on what assumptions that are made but essentially expresses the desire to

- label the training data by identifying what submodel each of the entry in the training data belongs to and
- identify the different submodels using the labeled training data.

The problem is that to label the data, the submodels are needed and to identify the submodels, the labels are needed. The two tasks should therefore ideally be done simultaneously. More formally, given the training data $\{(y_k, \boldsymbol{x}_k)\}_{k=1}^N$ this can be written as

$$\underset{\{M_i\}_{i=1}^m,\{\theta_i\}_{i=1}^m}{\text{minimize}} \sum_{k=1}^{N}\sum_{i=1}^{m} \delta_{k\in M_i} J(y_k - \theta_i^T \boldsymbol{x}_k) \tag{1}$$

where M_i is an index set, containing the indices of the training data that belong to submodel i, J a cost function measuring fit and $\delta_{k\in M_i}$ an indicator function that is zero whenever $k \notin M_i$ and one otherwise. θ_i is the model parameter of submodel i. Note that we both optimize over the model parameters $\{\theta_i\}_{i=1}^m$ and the index sets $\{M_i\}_{i=1}^m$ (essentially the labels). This problem is non-convex.

Existing PWA identification methods can in general be said being either a *greedy* or a *relaxation* approach. Greedy approaches to PWA identification alternate between finding the labels and the submodels. That is, a local optimum to (1) is sought by alternating between minimizing with respect to $\{M_i\}_{i=1}^m$ and $\{\theta_i\}_{i=1}^m$. These types of methods are known for being computationally efficient but are dependent on a good initialization for finding a good estimate.

More recently, a number of methods have been presented that approach the PWA identification problem by the use of relaxations and convex optimization. This type of method seek a convex relaxation of (1). The performance of these methods does not depend on a clever initialization. This valuable property is often traded for with a higher computational complexity compare to the greedy methods.

In this contribution, we study a method of the last type. Rather different than many other relaxation methods, the problem of simultaneously computing the label and the submodels is avoided by never computing the submodels. Instead, of using $\theta_i^T \boldsymbol{x}^*$ as an estimate for the output at $\boldsymbol{x}^*$ from submodel i, the method uses that we can linearly interpolate/extrapolate data from submodel i to give an estimate for the output at $\boldsymbol{x}^*$ from submodel i. Since the submodel parameters $\{\theta_i\}_{i=1}^m$ are never computed, the algorithm does not produce an estimate for $\{\theta_i\}_{i=1}^m$. Instead, given a set of training data $\{(y_k, \boldsymbol{x}_k)\}_{k=1}^{d+1}$ from some unknown PWA function f, the considered method computes a *data-based representation* of a PWA map Π which models f. That is, given a set of grid points $\{\hat{\boldsymbol{x}}_k\}_{k=1}^{\hat{N}}$, the corresponding outputs $\{\hat{y}_k\}_{k=1}^{\hat{N}}$ are estimated. The PWA method is therefore *discriminative* rather than *generative*, since it only implicitly computes a PWA model. One important advantage of the approach is that it implicitly provides the rule for labeling new data, while many other methods only provide the labels for the training set. The method seeks the set of outputs that:

(1) Maximizes the fit to the training data and
(2) generate an estimate for the sought outputs at the given grid points from a PWA model with the minimum number of hinges.

★ The authors gratefully acknowledge support by the Swedish Research Council in the Linnaeus center CADICS, the European Research Council under the advanced grant LEARN, contract 267381, by a postdoctoral grant from the Sweden-America Foundation, donated by ASEA's Fellowship Fund, and by a postdoctoral grant from the Swedish Research Council. This work was also supported by JSPS KAKENHI Grant Number 25820179.

The method we revisit in this contribution was previously presented in Maruta and Sugie (2011, 2012). In this contribution, theoretical background of the method is investigated, and a fact which is useful in determining the regularization parameter, which is the most important design parameter in the method, is derived. Also, it is shown that the alternating direction method of multipliers (ADMM) algorithm is effective numerical solver for the method.

2. BACKGROUND

PWA systems serve as popular models of nonlinear systems due to their universal approximation properties (Lin and Unbehauen, 1992; Breiman, 1993). In addition, it can also be shown that PWA systems are equivalent to certain types of hybrid systems, see *e.g.,* Bemporad et al. (2000); Heemels et al. (2001). This makes PWA systems a very important class of systems with an increasing interest. Five methods that have attained special attention in the literature are the clustering-based approach (Ferrari-Trecate et al., 2003), the bounded error approach (Bemporad et al., 2005), the mixed integer quadratic programming approach (Bemporad et al., 2001; Roll et al., 2004), the Bayesian approach (Juloski et al., 2005) and the algebraic approach (Vidal et al., 2003). For an overview of contributions see Paoletti et al. (2007); Garulli et al. (2012). The identification of PWA models is a complex task in which, simultaneously, both the labels and the linear submodels have to be found. The underlying problem is often non-convex and many methods can be seen as greedy approaches. These are then highly dependent on a good initialization for delivering a satisfying model. See *e.g.,* Roll (2003) for an overview.

The approaches discussed in Ohlsson (2010); Maruta and Sugie (2011); Bako (2011) approximates the underlying optimization problem with a convex relaxed problem. It is therefore insensitive to initialization, since it is convex, while being solvable for problems of practical sizes. Among them, the approaches studied in Ohlsson (2010); Bako (2011) only provide labels for the training data while the method discussed in this contribution (Maruta and Sugie, 2011) implicitly provides the rule for labeling new data.

The approach presented in Bemporad et al. (2001); Roll et al. (2004) discuss solving the non-convex problem (1) directly, without relaxing it. In the setting studied in Bemporad et al. (2001); Roll et al. (2004) the non-convex problem can be shown to be a mixed integer quadratic program. Such programs are known to be hard to solve (NP-hard in the worst case (Roll et al., 2004)) and the approach is therefore practically applicable only to very small problems.

The discussed approach is also related to sparse subspace clustering (SSC (Elhamifar and Vidal, 2013)).

3. NOTATION AND ASSUMPTIONS

For conciseness, we denote the set $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$ by $\{\boldsymbol{x}_k\}_{k=1}^{N}$. We will in general use $\boldsymbol{x}$ for inputs and y for outputs and assume that the inputs are in $\mathbb{R}^d$ and that the outputs are real scalars. Given a set of input-output pairs $\{(y_k, \boldsymbol{x}_k)\}_{k=1}^{d+1}$, we will be interested in the d-dimensional hyperplane spanned by the set of pairs. In particular, we will denote the linear interpolation of the output at a given input $\boldsymbol{x} \in \mathbb{R}^d$ using the notation:

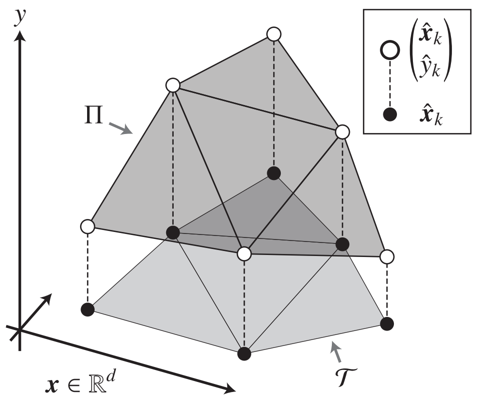
y
Π
$\boldsymbol{x} \in \mathbb{R}^d$
$\mathcal{T}$
$\begin{pmatrix}\hat{\boldsymbol{x}}_k \\ \hat{y}_k\end{pmatrix}$
$\hat{\boldsymbol{x}}_k$

Fig. 1. Illustration of a data-based representation of a PWA map model. The PWA map model Π is represented by a set of data points $\{(\hat{\boldsymbol{x}}_k, \hat{y}_k)\}$ connected according to a triangulation $\mathcal{T}$.

$$\mathrm{Lerp}\left(\boldsymbol{x}, \{(\boldsymbol{x}_k, y_k)\}_{k=1}^{d+1}\right) \triangleq \begin{bmatrix}\boldsymbol{x} \\ 1\end{bmatrix}^T \cdot \begin{bmatrix}\boldsymbol{x}_1^T & 1 \\ \boldsymbol{x}_2^T & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_{d+1}^T & 1\end{bmatrix}^{-1} \cdot \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{d+1}\end{bmatrix}. \qquad (2)$$

We will define $\mathrm{Co}(P)$ as the convex hull of a set of points P. For a vector $\boldsymbol{v}$, we denote ℓ_p norm of $\boldsymbol{v}$ by $\|\boldsymbol{v}\|_p$. Especially, $\|\boldsymbol{v}\|_0$ is defined as the number of non-zero elements in $\boldsymbol{v}$.

4. PWA MODEL IDENTIFICATION

In this paper, we study an PWA model identification method which is based on data-based representation of PWA maps and ℓ_1 relaxation. Before describing the main problem, the data-based scheme used for representing the PWA map model Π and complexity measure for the model are explained.

4.1 Data-based Representation of PWA Map

In the data-based representation scheme, a PWA map Π is represented by a set of data points $\{(\hat{\boldsymbol{x}}_k, \hat{y}_k)\}_{k=1}^{\hat{N}}$ connected according to a triangulation $\mathcal{T}$. $\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots \hat{\boldsymbol{x}}_{\hat{N}} \in \mathbb{R}^d$ and $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{\hat{N}} \in \mathbb{R}$ are inputs and outputs of the data points which represents the PWA map, respectively and $\mathcal{T}$ is a triangulation which subdivides $\mathrm{Co}\left(\{\hat{\boldsymbol{x}}_k\}_{k=1}^{\hat{N}}\right)$ into d-dimensional simplexes (triangles if $d = 2$). An example of data-based PWA map is shown in Fig. 1, and concrete definition of the data-based PWA map is as follows.

Definition 1. (Data-based PWA map). For given data set $\{(\hat{\boldsymbol{x}}_k, \hat{y}_k)\}_{k=1}^{\hat{N}}$ and triangulation $\mathcal{T}$, define the PWA map

$$\Pi : \mathrm{Co}\left(\{\hat{\boldsymbol{x}}_k\}_{k=1}^{\hat{N}}\right) \subset \mathbb{R}^d \mapsto \mathbb{R} \qquad (3)$$

as the map whose value $\Pi(\boldsymbol{x})$ is calculated by the following procedure [P1]–[P2]:

[P1] Choose the simplex (triangle) which includes $\boldsymbol{x}$ from $\mathcal{T}$, and let the vertexes of the simplex be $\hat{\boldsymbol{x}}_{v_1}, \hat{\boldsymbol{x}}_{v_2}, \ldots, \hat{\boldsymbol{x}}_{v_{(d+1)}}$.

[P2] Determine $\Pi(\boldsymbol{x})$ by linearly interpolating the data chosen in [P1] as

$$\mathrm{Lerp}\left(\boldsymbol{x}, \left\{\begin{pmatrix}\hat{\boldsymbol{x}}_{v_k} \\ \hat{y}_{v_k}\end{pmatrix}\right\}_{k=1}^{d+1}\right). \qquad (4)$$

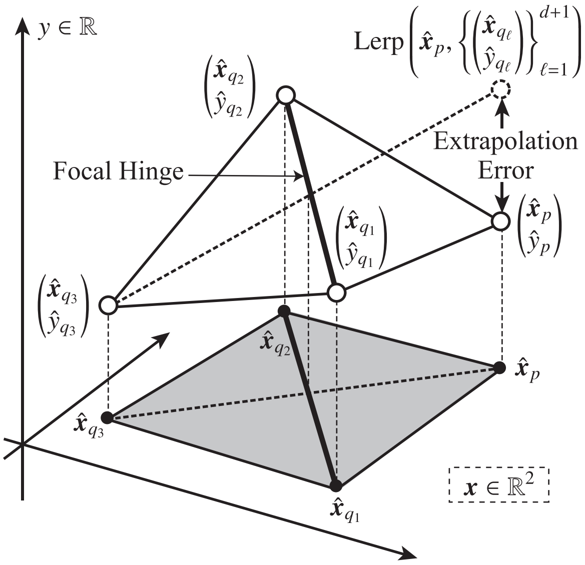
$y \in \mathbb{R}$
$\mathrm{Lerp}\left(\hat{\boldsymbol{x}}_p, \left\{\begin{pmatrix}\hat{\boldsymbol{x}}_{q_\ell}\\ \hat{y}_{q_\ell}\end{pmatrix}\right\}_{\ell=1}^{d+1}\right)$
$\begin{pmatrix}\hat{\boldsymbol{x}}_{q_2}\\ \hat{y}_{q_2}\end{pmatrix}$
Focal Hinge
Extrapolation Error
$\begin{pmatrix}\hat{\boldsymbol{x}}_{q_1}\\ \hat{y}_{q_1}\end{pmatrix}$
$\begin{pmatrix}\hat{\boldsymbol{x}}_{p}\\ \hat{y}_{p}\end{pmatrix}$
$\begin{pmatrix}\hat{\boldsymbol{x}}_{q_3}\\ \hat{y}_{q_3}\end{pmatrix}$
$\hat{\boldsymbol{x}}_{q_2}$
$\hat{\boldsymbol{x}}_{p}$
$\hat{\boldsymbol{x}}_{q_3}$
$\boldsymbol{x} \in \mathbb{R}^2$
$\hat{\boldsymbol{x}}_{q_1}$

Fig. 2. Illustration of the extrapolation error calculated for distinguishing whether the focal hinge is broken or not ($d = 2$).

By Definition 1, Π reproduces the data set, that is

$$\Pi\left(\hat{\boldsymbol{x}}_k\right) = \hat{y}_k \quad \left(k = 1, 2, \ldots, \hat{N}\right), \tag{5}$$

and is a piece-wise affine map with d-simplex facets.

4.2 Measure of Model Complexity

PWA map models can have arbitrary complicated structures and can reproduce any data set. However, excessively complicated models constructed from noisy data sets tend to reproduce the noise in the training data and are not useful in practical situations. To control the complexity of the PWA model, we introduce a complexity measure.

In essence, we measure the complexity of the map by counting the number of broken hinges in the map. To distinguish whether a hinge in the map is broken or not in a mathematical way, we focus on the pair of faces which shares a hinge as their edge, and let the indexes corresponding to their vertexes be $(q_1, q_2, \ldots, q_{d+1})$ and $(p, q_1, \ldots, q_d)$ (see Fig. 2). We then calculate the error between the extrapolation from the data set on one face $(q_1, q_2, \ldots, q_{d+1})$ and the point exclusively belonging to the other face p, that is,

$$\hat{y}_p - \mathrm{Lerp}\left(\hat{\boldsymbol{x}}_p, \left\{\begin{pmatrix}\hat{\boldsymbol{x}}_{q_\ell}\\ \hat{y}_{q_\ell}\end{pmatrix}\right\}_{\ell=1}^{d+1}\right), \tag{6}$$

and call this error the extrapolation error. Fig. 2 illustrates this extrapolation error for $d = 2$. As seen in the figure, the hinge is broken if the error is non-zero. The number of broken hinges is hence equal to the number of non-zero extrapolation errors. We can hence write the complexity measure as

$$\sum_{\left(p,\{q_\ell\}_{\ell=1}^{d+1}\right)\in S} \left\| \hat{y}_p - \mathrm{Lerp}\left(\hat{\boldsymbol{x}}_p, \left\{\begin{pmatrix}\hat{\boldsymbol{x}}_{q_\ell}\\ \hat{y}_{q_\ell}\end{pmatrix}\right\}_{\ell=1}^{d+1}\right) \right\|_0, \tag{7}$$

where S is the index set of all neighboring simplex-point pairs in $\mathcal{T}$, and the summation is done over all elements in S. Since this summation counts a hinge twice, (7) equals twice the number of broken hinges in Π.

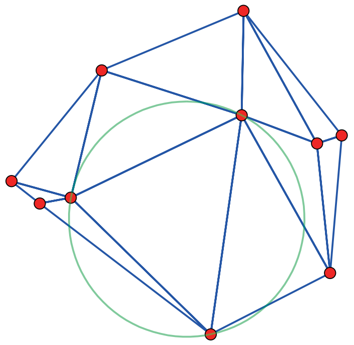

Fig. 3. An example of Delaunay triangulation for P in the plane. No point in P is inside the circumcircle of any Delaunay triangle (simplex).

4.3 Problem Setting

Here, we consider a system described by

$$y = f(\boldsymbol{x}) + \eta, \tag{8}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is the known input of the system; $y \in \mathbb{R}$ is the measurable output of the system; and $f : \mathbb{R}^d \mapsto \mathbb{R}$ is an unknown map. Although f is unknown, it is assumed that f is a PWA map with a finite number of modes, and $\eta \in \mathbb{R}$, which is defined as the difference between $f(\boldsymbol{x})$ and y, can be regarded as a stochastic noise.

The problem here is to construct a PWA map $\Pi : \mathbb{R}^d \mapsto \mathbb{R}$ which models f given a training data set consisting of N pairs of I/O data points

$$\left\{ \begin{pmatrix}\boldsymbol{x}_1\\ y_1\end{pmatrix}, \begin{pmatrix}\boldsymbol{x}_2\\ y_2\end{pmatrix}, \ldots, \begin{pmatrix}\boldsymbol{x}_N\\ y_N\end{pmatrix} \right\}, \tag{9}$$

from (8). In constructing Π, a set of grid points at which the output of Π is of interest and a triangulation $\mathcal{T}$ is assumed to be given. For example, $\{\hat{\boldsymbol{x}}_k\}_k^{\hat{N}}$ and $\mathcal{T}$ could be chosen by randomly sample from the required model domain and a Delaunay triangulation could be used.

Remark 2. To obtain $\mathcal{T}$, we could use Delaunay triangulation. Delaunay triangulation is a triangulation, where the interior of the circumcircle of any triangle in Delaunay triangulation contains no points of the set (see Fig. 3). Delaunay triangulation also can be extended to higher dimensions and has a lot of applications. For more information about Delaunay triangulation, see *e.g.,* de Berg et al. (2008).

4.4 Identification Procedure

Given a noisy data set $\{(\boldsymbol{x}_k, y_k)\}_{k=1}^{N}$, the grid points $\{\hat{\boldsymbol{x}}_k\}_{k=1}^{\hat{N}}$ and the triangulation $\mathcal{T}$, we seek to construct Π by minimizing the complexity and by maximizing the fit to the given training data set. Consequently, we consider to determine $\{\hat{y}_k\}_{k=1}^{\hat{N}}$ through the following optimization problem:

$$\begin{aligned}\underset{\hat{y}_1,\hat{y}_2,\ldots,\hat{y}_{\hat{N}}}{\text{minimize}}\ & \frac{1}{2}\sum_{k=1}^{N}\left(y_k - \Pi\left(\boldsymbol{x}_k\right)\right)^2 \\ & + \lambda \sum_{\substack{p\\ \left(\{q_\ell\}_{\ell=1}^{d+1}\right)\in S}} \left\| \hat{y}_p - \mathrm{Lerp}\left(\hat{\boldsymbol{x}}_p, \left\{\begin{pmatrix}\hat{\boldsymbol{x}}_{q_\ell}\\ \hat{y}_{q_\ell}\end{pmatrix}\right\}_{\ell=1}^{d+1}\right) \right\|_0\end{aligned} \tag{10}$$

where S is the set consists of all pairs of faces and neighboring vertexes in $\mathcal{T}$, and $\lambda > 0$ is a regularization parameter for balancing between model fit and the model complexity. Since $\Pi\left(\boldsymbol{x}_k\right)$ and Lerp$\left(\hat{\boldsymbol{x}}_p, \left\{\begin{pmatrix}\hat{\boldsymbol{x}}_{q_\ell}\\ \hat{y}_{q_\ell}\end{pmatrix}\right\}_{\ell=1}^{d+1}\right)$ in (10) are linear in $\hat{y}_1, \dots, \hat{y}_{\hat{N}}$, (10) can be written in the following form:

$$\underset{\hat{\boldsymbol{y}}}{\text{minimize}}\ \frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{A}\hat{\boldsymbol{y}}\right\|_2^2+\lambda\left\|\boldsymbol{F}\hat{\boldsymbol{y}}\right\|_0, \tag{11}$$

where $\boldsymbol{y} \triangleq \left[y_1, \dots, y_N\right]^T$ is the vector of given output data; $\hat{\boldsymbol{y}} \triangleq \left[\hat{y}_1, \dots, \hat{y}_{\hat{N}}\right]^T$ is the vector of output data for representing the PWA model Π; $\boldsymbol{A} \in \mathbb{R}^{N\times\hat{N}}$ is the matrix depends on $\{\boldsymbol{x}_k\}_{k=1}^{N}, \{\hat{\boldsymbol{x}}_k\}_{k=1}^{\hat{N}}$ and triangulation $\mathcal{T}$; $\boldsymbol{F} \in \mathbb{R}^{M\times\hat{N}}$ is the matrix depends on $\{\hat{\boldsymbol{x}}_k\}_{k=1}^{\hat{N}}$ and triangulation $\mathcal{T}$; and M is the number of hinges in Π.

Although the solution of the problem (11) would be the one with least number of broken hinges with a certain fit, the problem is known to be NP-hard. So, we relax (11) to the ℓ_1-regularized least squares problem:

$$\underset{\hat{\boldsymbol{y}}}{\text{minimize}}\ \frac{1}{2}\left\|\boldsymbol{y}-\boldsymbol{A}\hat{\boldsymbol{y}}\right\|_2^2+\lambda\left\|\boldsymbol{F}\hat{\boldsymbol{y}}\right\|_1, \tag{12}$$

This relaxed problem can be solved efficiently and still provides us good models as shown in the following sections.

Remark 3. $\boldsymbol{A}$ is sparse matrix with $d+1$ non-zero elements in each row since every element in $\boldsymbol{y}$ is interpolated from $d+1$ elements in $\hat{\boldsymbol{y}}$ by Π.

Remark 4. $\boldsymbol{F}$ is sparse matrix with $d+2$ non-zero elements in each row since every extrapolation error is calculated from $d+2$ elements in $\hat{\boldsymbol{y}}$.

5. THEORETICAL ANALYSIS

5.1 Regularization Path and Critical Parameter Value

The estimated parameter $\hat{\boldsymbol{y}}$ as a function of the regularization parameter λ is called the regularization path for the problem. Roughly, large values of λ result in PWA map models with simpler structure but worst fit. To find a suitable model, we have to evaluate $\hat{\boldsymbol{y}}$ for various λ. A basic result from convex analysis tell us that there is a value $\lambda^{\max}$ such that $\hat{\boldsymbol{y}}(\lambda)$ corresponds a flat linear map (no broken hinges) if and only if $\lambda \geq \lambda^{\max}$. It is very helpful to know the critical parameter value $\lambda^{\max}$ in practice since it gives a very good starting point in finding a suitable value of λ.

Let $\hat{\boldsymbol{y}}(\lambda^{\max})$ be the estimated parameter correspond to flat linear map. Then $\hat{\boldsymbol{y}}(\lambda^{\max})$ is calculated as

$$\hat{\boldsymbol{y}}(\lambda^{\max}) = \hat{\boldsymbol{X}}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{y}, \tag{13}$$

where

$$\boldsymbol{X} \triangleq \begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_N\\ 1 & 1 & \cdots & 1\end{bmatrix}^T, \quad \hat{\boldsymbol{X}} \triangleq \begin{bmatrix}\hat{\boldsymbol{x}}_1 & \hat{\boldsymbol{x}}_2 & \cdots & \hat{\boldsymbol{x}}_{\hat{N}}\\ 1 & 1 & \cdots & 1\end{bmatrix}^T.$$

Then $\hat{\boldsymbol{y}}(\lambda^{\max})$ is optimal solution of (12) if and only if 0 is in the subdifferential of the objective function at $\hat{\boldsymbol{y}} = \hat{\boldsymbol{y}}(\lambda^{\max})$,

$$\left\{\boldsymbol{A}^T\left(-\boldsymbol{y}+\boldsymbol{A}\hat{\boldsymbol{y}}\left(\lambda^{\max}\right)\right)+\lambda\boldsymbol{F}^T\boldsymbol{z}\ \middle|\ \|z\|_\infty \leq 1\right\}, \tag{14}$$

and we can calculate $\lambda^{\max}$ from the solution of the linear program

$$\begin{align} \underset{\mu, z}{\text{minimize}} \quad & -\mu \tag{15a}\\ \text{subject to} \quad & \mu\boldsymbol{A}^T\left(-\boldsymbol{y}+\boldsymbol{A}\hat{\boldsymbol{y}}\left(\lambda^{\max}\right)\right)+\boldsymbol{F}^T\boldsymbol{z} \tag{15b}\\ & \|z\|_\infty \leq 1 \tag{15c}\end{align}$$

as $\lambda^{\max} = \frac{1}{\mu}$.

5.2 Guaranteed Recovery of the Solution with the Least Number of Hinges

It is interesting to analyze under what conditions the relaxation of the zero norm by the ℓ_1 norm is tight. To gain some insight, consider the noise free case. We are then led to the problem of finding a PWA function that satisfies the measurements and that minimizes the number of hinges,

$$\begin{aligned} \underset{\hat{\boldsymbol{y}}}{\text{minimize}} \quad & \|\boldsymbol{F}\hat{\boldsymbol{y}}\|_0,\\ \text{subject to} \quad & \boldsymbol{y} = \boldsymbol{A}\hat{\boldsymbol{y}}.\end{aligned} \tag{16}$$

Interestingly, (11) and (16) can be shown equivalent if $\lambda \leq \lambda^{\min}$ in (11), for some certain $\lambda^{\min} \geq 0$. It can also be shown that there exists a $\tilde{\lambda}^{\min} \geq 0$ such that whenever $\lambda \leq \tilde{\lambda}^{\min}$ the relaxed versions of (11) and (16) are the same.

A hot topic in compressive sensing (CS, Candès et al. (2006); Donoho (2006)) has been to derive conditions for when the relaxation is tight. However, all work we are aware of discuss $\boldsymbol{F} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix.

The following definition will become useful.

Definition 5. (RIP). For a given integer k, $\boldsymbol{A} \in \mathbb{R}^{N\times\hat{N}}$, and $\boldsymbol{F} \in \mathbb{R}^{M\times\hat{N}}$, we say that the pair $(\boldsymbol{A}, \boldsymbol{F})$ is (ϵ, k)-RIP, with

$$\begin{aligned} \epsilon = \text{argmax}_{\hat{\boldsymbol{y}}\neq 0}&\left|\frac{\|\boldsymbol{A}\hat{\boldsymbol{y}}\|_2^2}{\|\hat{\boldsymbol{y}}\|_2^2}-1\right|\\ \text{subject to} \quad & \|\boldsymbol{F}\hat{\boldsymbol{y}}\|_0 \leq k.\end{aligned} \tag{17}$$

Theorem 6. (Uniqueness). Let $(\boldsymbol{A}, \boldsymbol{F})$ be $(\epsilon, 2k)$-RIP with $\epsilon < 1$. If $\hat{\boldsymbol{y}}$ satisfies $\|\boldsymbol{F}\hat{\boldsymbol{y}}\|_0 \leq k$, and $\boldsymbol{y} = \boldsymbol{A}\hat{\boldsymbol{y}}$ then $\hat{\boldsymbol{y}}$ is unique.

Proof. Assume that $\hat{\boldsymbol{y}}$ is not unique and that there exists a $\tilde{\boldsymbol{y}}$ such that $\boldsymbol{y} = \boldsymbol{A}\tilde{\boldsymbol{y}}$, $\|\boldsymbol{F}\tilde{\boldsymbol{y}}\|_0 \leq k$, and $\tilde{\boldsymbol{y}} \neq \hat{\boldsymbol{y}}$. Since both $\boldsymbol{y} = \boldsymbol{A}\tilde{\boldsymbol{y}}$ and $\boldsymbol{y} = \boldsymbol{A}\hat{\boldsymbol{y}}$, we have that $\boldsymbol{A}(\tilde{\boldsymbol{y}}-\hat{\boldsymbol{y}}) = 0$. It also holds that $\|\boldsymbol{F}(\tilde{\boldsymbol{y}}-\hat{\boldsymbol{y}})\|_0 \leq 2k$. We hence have that

$$\left|\frac{\|\boldsymbol{A}(\tilde{\boldsymbol{y}}-\hat{\boldsymbol{y}})\|_2^2}{\|\hat{\boldsymbol{y}}\|_2^2}-1\right| = 1 \tag{18}$$

for a vector $\tilde{\boldsymbol{y}}-\hat{\boldsymbol{y}} \neq 0$ and $\|\boldsymbol{F}(\tilde{\boldsymbol{y}}-\hat{\boldsymbol{y}})\|_0 \leq 2k$. This is a contradiction since it was assumed that $(\boldsymbol{A}, \boldsymbol{F})$ is $(\epsilon, 2k)$-RIP with $\epsilon < 1$. We hence have that $\tilde{\boldsymbol{y}} = \hat{\boldsymbol{y}}$ and that the solution is unique.

Corollary 7. (Recovery). Let $\boldsymbol{y}^*$ be the solution of

$$\begin{aligned} \underset{\hat{\boldsymbol{y}}}{\text{minimize}} & \|\boldsymbol{F}\hat{\boldsymbol{y}}\|_1,\\ \text{subject to} \quad & \boldsymbol{y} = \boldsymbol{A}\hat{\boldsymbol{y}}.\end{aligned} \tag{19}$$

The solution of (16) is identical to that of (19) if $\|\boldsymbol{F}\boldsymbol{y}^*\|_0 \leq k$ and $(\boldsymbol{A}, \boldsymbol{F})$ is $(\epsilon, 2k)$-RIP with $\epsilon < 1$.

Proof. This result follows from the previous theorem since under the conditions of the corollary, any solution to $\boldsymbol{y} = \boldsymbol{A}\hat{\boldsymbol{y}}$ with $\|\hat{\boldsymbol{y}}\|_0 \leq k$ is the unique solution to (16).

6. NUMERICAL SOLVER

The problem (12) can be converted to a convex quadratic programming problem and a number of optimization methods for convex optimization problems and quadratic programming problems are available for solving the problem. Among them, we focus on the alternating direction method of multipliers (ADMM) algorithm which possess remarkable advantages for the problem as described in this section.

Algorithm 1 ADMM algorithm for (12)

```
Require: $\boldsymbol{y}, \lambda, \boldsymbol{A}, \boldsymbol{F}$
Ensure: $\hat{\boldsymbol{y}}$
1: Initialize $\boldsymbol{z}^1 \in \mathbb{R}^M, \boldsymbol{u}^1 \in \mathbb{R}^M$
2: $k := 1$
3: repeat
4:     $\hat{\boldsymbol{y}}^{k+1} := \left(\boldsymbol{A}^T\boldsymbol{A} + \rho\boldsymbol{F}^T\boldsymbol{F}\right)^{-1}\left(\boldsymbol{A}^T\boldsymbol{y} + \rho\boldsymbol{F}^T\left(\boldsymbol{z}^k - \boldsymbol{u}^k\right)\right)$
5:     $\boldsymbol{z}^{k+1} := S_{\lambda/\rho}\left(\rho\boldsymbol{F}\hat{\boldsymbol{y}}^{k+1} + \boldsymbol{u}^k\right)$
6:     $\boldsymbol{u}^{k+1} := \boldsymbol{u}^k + \boldsymbol{F}\hat{\boldsymbol{y}}^{k+1} - \boldsymbol{z}^{k+1}$
7:     $k := k + 1$
8: until termination criteria satisfied
9: $\hat{\boldsymbol{y}} := \hat{\boldsymbol{y}}^k$
```

The essential part of ADMM algorithm for solving (12) is shown in Algorithm 1, where $\rho > 0$ is a user-defined constant and $S_{\lambda/\rho}$ is the soft thresholding operator with level λ/ρ. For the details of these components and termination criteria, see *e.g.,* Boyd et al. (2011).

6.1 Scalability

The first advantage of ADMM algorithm in solving (12) is its capability to solve large-scale problems. Indeed, the complexity is only linear in $\hat{N}$ for each ADMM iteration as discussed in the following.

Here, we assume that grid points $\{\hat{\boldsymbol{x}}_k\}$ and $\mathcal{T}$ are obtained via Delaunay triangulation of randomly distributed points. Under this assumption, the expected number of hinges M is proportional to the number of grid points $\hat{N}$ (Dwyer, 1991). The complexity of the Delaunay triangulation algorithm is $O\left(\hat{N}^{\lceil d/2 \rceil + 1}\right)$ and polynomial in $\hat{N}$ (Cignoni et al., 1998).

In Algorithm 1, we have to solve a linear system for updating $\hat{\boldsymbol{y}}^k$ (line 4) and this is the most significant step when considering the required amount of computation. Although solving the linear equation for general $\boldsymbol{A}^T\boldsymbol{A} + \rho\boldsymbol{F}^T\boldsymbol{F}$ costs $O\left(\hat{N}^3\right)$ flops for initial factorization and $O\left(\hat{N}^2\right)$ flops for each iteration, the equation can be solved in much more efficient way when $\boldsymbol{A}^T\boldsymbol{A} + \rho\boldsymbol{F}^T\boldsymbol{F}$ is sparse (Boyd et al., 2011, §4.2.2). As for the problem (12), $\boldsymbol{A}^T\boldsymbol{A} + \rho\boldsymbol{F}^T\boldsymbol{F}$ is a sparse matrix with $O\left(d^2\hat{N}\right)$ non-zero elements because $\boldsymbol{A}$ and $\boldsymbol{F}$ have sparsity patterns stated in Remarks 3 and 4. This sparsity allows efficient calculation with a permuted Cholesky factorization. Although the permutation in this process rely on heuristic algorithm and it is difficult to ensure the performance, the algorithm is well established (George and Liu, 1989) and the total amount of calculation for solving the linear system is empirically proportional to $\hat{N}$ as shown through a numerical example in Section 7. Note that the other calculations in the algorithm also costs flops proportional to $\hat{N}$ since the number of non-zero elements in $\boldsymbol{A}$ and $\boldsymbol{F}$ are proportional to $\hat{N}$.

6.2 Warm Start

The second advantage of ADMM algorithm is its capability to perform warm start. In the initialization step (line 1), $\boldsymbol{z}^1$ and $\boldsymbol{u}^1$ are initialized by zero in a standard manner. When we run the algorithm multiple times for slightly different problems, we can initialize these values by $\boldsymbol{z}^k$ and $\boldsymbol{u}^k$ in the previous run to obtain good results in far fewer iterations. This is called warm start. For obtaining the regularization path, we have to solve (12) a number of times for different λ, and this warm start initialization significantly reduce the computation time.

7. EXAMPLES

We illustrate the method and the above discussions through examples. In the following examples, MATLAB is used for implementing the ADMM algorithm, and CVX (Grant and Boyd, 2013, 2008) is used for solving (15).

Example 1: Numerical Example

Consider the system

$$y = \max\left(1 - \|\boldsymbol{x}\|_1, 0\right) + \eta \tag{20}$$

where $d = 2$ and $\eta \sim \mathcal{N}\left(0, 0.1^2\right)$. For this system, a PWA map model Π, whose domain is $\{\boldsymbol{x} \mid \|\boldsymbol{x}\|_\infty < 1.1\}$, is constructed. The input to the system $\{\boldsymbol{x}\}_k^N$ is uniformly distributed random number. For the identification process, $2\,000\,(= N)$ samples are generated from the system (20) and additional $2\,000\left(= N^{\text{vld}}\right)$ samples $\left\{\left(\boldsymbol{x}_k^{\text{vld}}, y_k^{\text{vld}}\right)\right\}_k^{N^{\text{vld}}}$ are generated for validation purpose.

First, the grid points $\{\hat{\boldsymbol{x}}_k\}_{k=1}^{\hat{N}}$ and the triangulation $\mathcal{T}$ for representing Π, which covers the model domain, is set by clipping Delaunay triangulation of uniformly distributed points. The number of data points used for representing Π is $8830\left(= \hat{N}\right)$. Then, $\{\hat{y}_k\}_{k=1}^{\hat{N}}$ is estimated with the method described in Section 4.4. The result for $2\lambda^{\max} \geq \lambda \geq 10^{-4}\lambda^{\max}$, where $\lambda^{\max}$ is calculated in advance by solving (15), is shown in Fig. 4 with the error between the obtained map Π and the validation data. It is confirmed that a good model which reproduce the pyramid-shaped structure of the system is obtained by choosing λ which minimizes the validation error. We clearly see that small λ results in model affected by noise and large λ eliminates important model structures.

As stated in Section 5.1, $\lambda \geq \lambda^{\max}$ produces the flat linear model and calculation of $\lambda^{\max}$ provides a good starting point for searching best λ.

To clarify the capability of the method for large-scale problems, calculation time required for one $\hat{\boldsymbol{y}}^k$ update in Algorithm 1 is examined for various $\hat{N}$ (other settings are the same as above). The circles in Fig. 5 shows the required time for the factorization stage (executed at the first iteration) and required time for each iteration separately. The lines show the fitted power functions. As seen in the plot, required computation time is almost linear to $\hat{N}$ as stated in Section 6.1, and the scalability of the method is confirmed.

Example 2: Application to Mechanical System

Let us consider an example with real data to confirm the effectiveness in practical situations. The rotary actuator FHA-17C-100-E250 manufactured by Harmonic Drive Systems Inc. (shown in Fig. 6) is the target system in this example. This actuator is equipped with a harmonic drive speed reducer, which is commonly used in industrial robots and have strong nonlinear friction (Taghirad and Bélanger, 1998). The input to the system is torque command voltage u [V], and the output is the angular speed ω [rad/s] which is obtained by taking

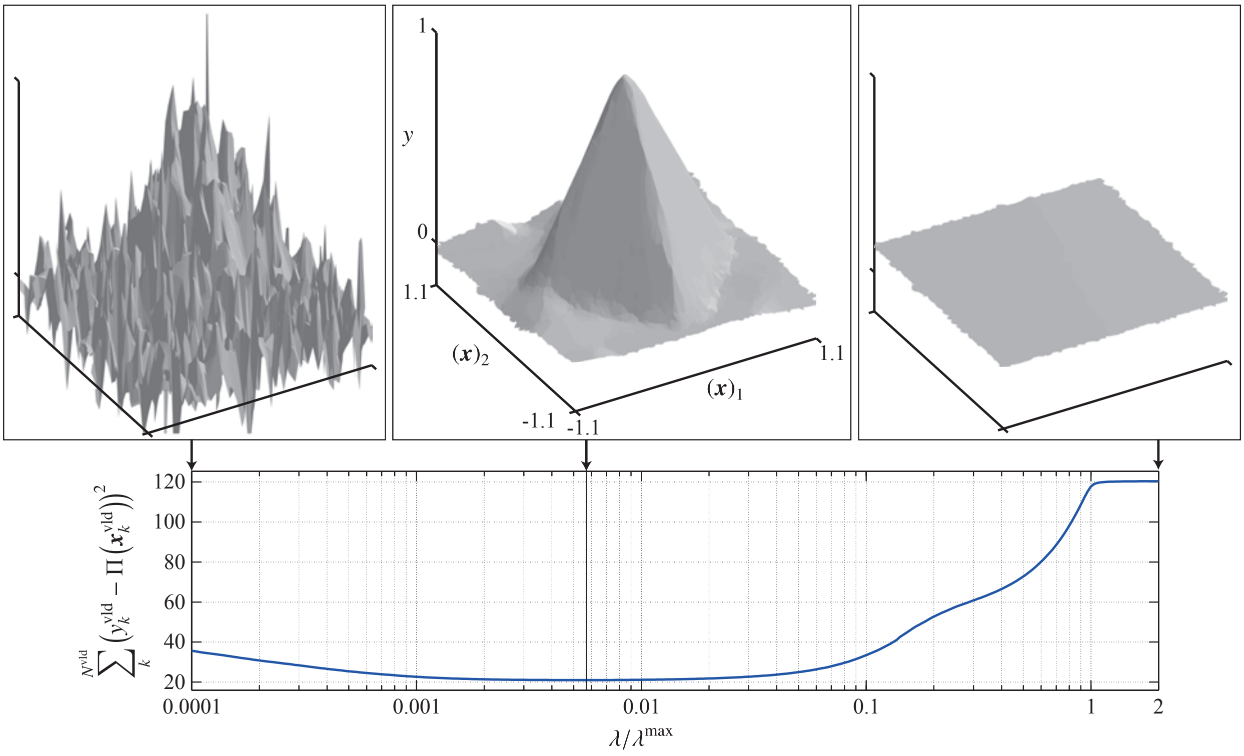

Fig. 4. Regularization path and error with validation data for Example 1. Obtained PWA map Π for $\lambda = 10^{-4}\lambda^{\max}, 5.7 \times 10^{-3}\lambda^{\max}, 2\lambda^{\max}$ are shown.

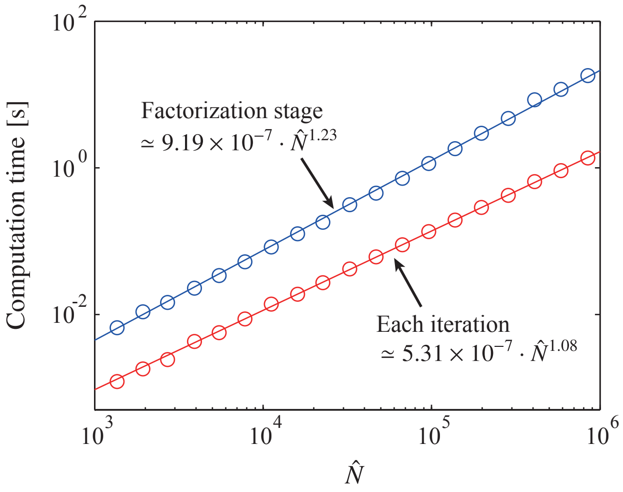

Fig. 5. Calculation time required for $\hat{\boldsymbol{y}}^k$ update in Algorithm 1 and fitted power functions. Calculation time required for factorization stage and each iteration are separately shown.

backward difference of the rotation angle measured by the built-in optical encoder. This system is already modeled by a PWA model with two-dimensional input in Maruta and Sugie (2012). Here, we are going to model the system by a PWA model with three-dimensional input to obtain a more accurate model and to show the advantage of using ADMM in a relatively large problem. The problem here is to construct a non-linear dynamic model

$$\omega(k) = \Pi\left([\omega(k-1), \omega(k-2), u(k)]^T\right), \tag{21}$$

Fig. 6. Rotary actuator (Harmonic Drive Systems Inc. FHA-17C-100-E250)

where $\omega(k)$ and $u(k)$ are the k-th sample of ω and u sampled with 10 ms interval, respectively. And, Π is the PWA map

$$\Pi : \boldsymbol{x} \mapsto y \qquad \left(\begin{bmatrix} \omega(k-1) \\ \omega(k-2) \\ u(k) \end{bmatrix} \mapsto \omega(k)\right) \tag{22}$$

going to be estimated.

The I/O data set for the identification is obtained by changing u with random numbers normally distributed with zero mean and 1 V standard deviation in every 0.1 s. The I/O data for the first 5 s is shown in Fig. 7. Here, 512 000 samples are obtained and 300 001st to 400 000th samples are used as the training data

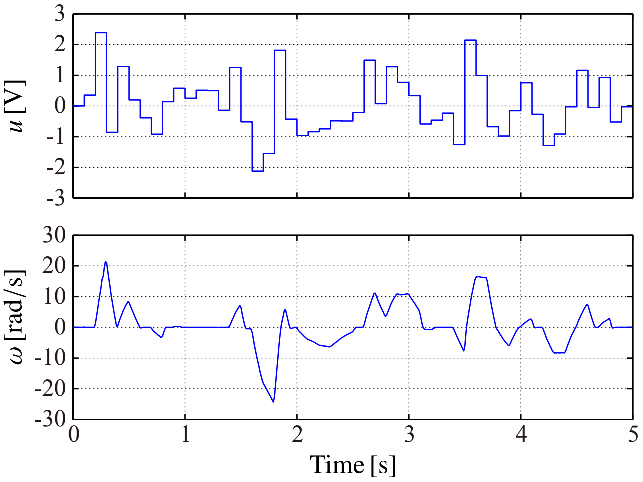

Fig. 7. Training data obtained from rotary actuator for first 5 seconds

$\{\boldsymbol{x}_k, y_k\}_{k=1}^{N}$ ($N = 100\,000$). For the validation purpose, 400 001st to 500 000th samples are used.

For constructing model, the grid points $\{\hat{\boldsymbol{x}}_k\}_{k=1}^{\hat{N}}$ ($\hat{N} = 10\,000$) are set by random points normally distributed with the empirical mean and covariance of $\{\boldsymbol{x}_k\}_{k=1}^{N}$, and triangulation $\mathcal{T}$ is set as Delaunay triangulation of $\left\{\Sigma^{-1/2}\hat{\boldsymbol{x}}_k\right\}_{k=1}^{\hat{N}}$, where Σ is the empirical covariance matrix of $\{\boldsymbol{x}_k\}_{k=1}^{N}$ and used for normalizing the input variables with different physical dimensions.

With the above setting, Π is constructed by the method described in Section 4.4 for $\lambda \in \left[10^{-4}\lambda^{\max}, 2\lambda^{\max}\right]$, where $\lambda^{\max}$ is calculated in advance by solving (15). The plot in Fig. 8 shows the root mean square error (RMSE) between the output signal of the validation set and the output of the model (21) with Π obtained for various λ. The figure shows that $\lambda^{\max}$ obtained from (15) is a good starting point for finding the best λ for this problem. The RMSE value is minimized to 0.585 rad/s when $\lambda = 3.08 \times 10^{-2}\lambda^{\max}$ and the last 10 s of the model output for the validation input is shown in Fig. 9. Since RMS value of the validation output is 8.37 rad/s, about 93% of the output is successfully modeled.

For the comparison purpose, the output of the Hammerstein-Wiener model obtained with `nlhw` function in MATLAB System Identification Toolbox is also shown in Fig. 9. In using `nlhw` function, the input and output nonlinearities are set to 10 units piecewise linear blocks (default setting), and the model orders for the linear dynamic block are set to $\left(n_b, n_f, n_k\right) = (7, 8, 0)$, where n_b is the number of zeros plus 1, n_f is the number of poles, and n_k is the input delay. These parameters are chosen from

$$\left\{\left(n_b, n_f, n_k\right) \middle| 1 \le n_b \le 10,\ 1 \le n_f \le 10,\ 0 \le n_k \le 9\right\} \qquad (23)$$

to minimize the RMSE value for the validation data. The RMSE value for the model is 1.998 rad/s, and about 76% of the output is modeled by the method.

To validate the generalization ability, we compare the response of the models with the real system for different type of input signal. The cyclic input signal used for the validation and responses are shown in Fig. 10. In the figure, the response of the experiment device for 200 cycles are shown as the red thin lines and the response of the model (21) with Π obtained for

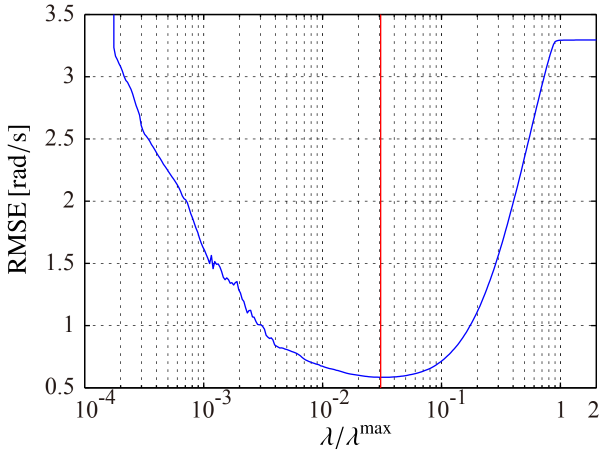

Fig. 8. Regularization path and error with validation data for Example 2.

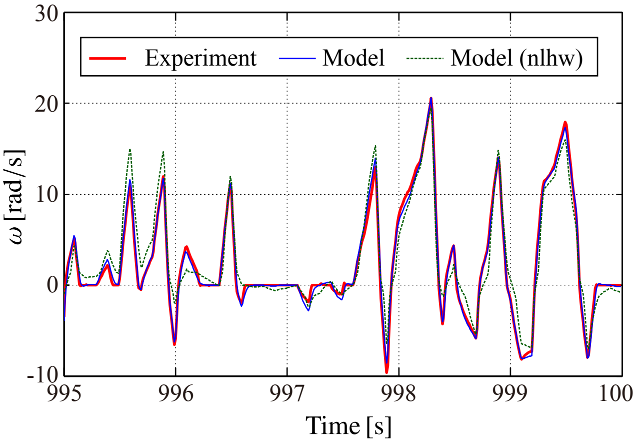

Fig. 9. Response of rotary actuator (red thick line) model with Π for $\lambda = 3.08 \times 10^{-2}\lambda^{\max}$ (blue solid line) and Hammerstein-Wiener model obtained with `nlhw` function in MATLAB System Identification Toolbox (green dashed line) for validation data

$\lambda = 3.08 \times 10^{-2}\lambda^{\max}$ is shown as the blue solid line. In addition, the Hammerstein-Wiener model obtained with `nlhw` function in MATLAB System Identification Toolbox is also shown as green dashed line for comparison. These results shows that an appropriate PWA model is obtained by the method described in Section 4.4 for the practical system, which is difficult to model with the method based on Hammerstein-Wiener model.

As for the calculation time, it is confirmed that the warm-start feature of the ADMM algorithm is effective in a practical situation. While it takes 48.2 s to solve (12) by interior-point algorithm with MATLAB `quadprog` function for $\lambda = 0.1$, the ADMM algorithm requires only 6.8 s to obtain a solution of comparable quality starting from the solution for $\lambda = 0.08$.

8. CONCLUSION

This paper revisits a previously presented method for PWA system identification. We revisit the method to give a novel derivation which shows that under some certain conditions, the method computes the PWA function that passes through the training data and that has the least amount of broken hinges. We also derive an efficient implementation using ADMM and a way

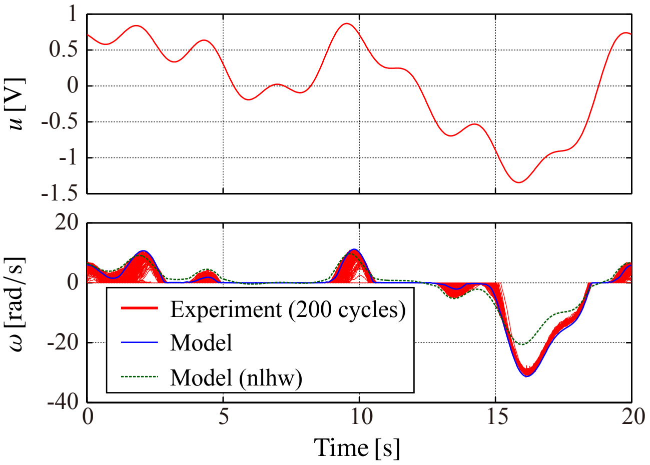

Fig. 10. Cyclic torque command signal for model validation (above) and response of experiment device and models (below)

to obtain a good starting point for the regularization parameter. A number of examples are shown to illustrate these discussions and the effectiveness of the method.

REFERENCES

Bako, L. (2011). Identification of switched linear systems via sparse optimization. *Automatica*, 47(4), 668–677.

Bemporad, A., Ferrari-Trecate, G., and Morari, M. (2000). Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control*, 45(10), 1864–1876.

Bemporad, A., Garulli, A., Paoletti, S., and Vicino, A. (2005). A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50(10), 1567–1580.

Bemporad, A., Roll, J., and Ljung, L. (2001). Identification of hybrid systems via mixed-integer programming. In *Proceedings of the 40th IEEE Conference on Decision and Control*, volume 1, 786–792.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1), 1–122.

Breiman, L. (1993). Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transactions on Information Theory*, 39(3), 999 –1013.

Candès, E.J., Romberg, J., and Tao, T. (2006). Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2), 489–509.

Cignoni, P., Montani, C., and Scopigno, R. (1998). DeWall: A fast divide and conquer Delaunay triangulation algorithm in E^d. *Computer-Aided Design*, 30(5), 333 – 341.

de Berg, M., Cheong, O., van Kreveld, M., and Overmars, M. (2008). *Computational Geometry: Algorithms and Applications*. Springer-Verlag, third edition.

Donoho, D. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4), 1289 –1306.

Dwyer, R.A. (1991). Higher-dimensional voronoi diagrams in linear expected time. *Discrete & Computational Geometry*, 6, 343–367.

Elhamifar, E. and Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11), 2765–2781.

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.

Garulli, A., Paoletti, S., and Vicino, A. (2012). A survey on switched and piecewise affine system identification. In *Proceedings of the 16th IFAC Symposium on System Identification, SYSID 2012*, 344–355.

George, A. and Liu, J. (1989). The evolution of the minimum degree ordering algorithm. *SIAM Review*, 31(1), 1–19.

Grant, M. and Boyd, S. (2008). Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura (eds.), *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, 95–110. Springer-Verlag Limited. `http://stanford.edu/~boyd/graph_dcp.html`.

Grant, M. and Boyd, S. (2013). CVX: Matlab software for disciplined convex programming, version 2.0 beta. `http://cvxr.com/cvx`.

Heemels, W.P.M.H., Schutter, B.D., and Bemporad, A. (2001). Equivalence of hybrid dynamical models. *Automatica*, 37(7), 1085 – 1091.

Juloski, A.L., Weiland, S., and Heemels, W.P.M.H. (2005). A Bayesian approach to identification of hybrid systems. *IEEE Transactions on Automatic Control*, 50(10), 1520–1533.

Lin, J.N. and Unbehauen, R. (1992). Canonical piecewise-linear approximations. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 39(8), 697 –699.

Maruta, I. and Sugie, T. (2011). Identification of PWA models via data compression based on l1 optimization. In *Proceedings of the 50th IEEE Conference on Decision and Control*, 2800–2805.

Maruta, I. and Sugie, T. (2012). Identification of pwa models via optimal data compression. In *Proceedings of the 16th IFAC Symposium on System Identification, SYSID 2012*.

Ohlsson, H. (2010). *Regularization for Sparseness and Smoothness - Applications in System Identification and Signal Processing*. Linköping Studies in Science and Technology. Dissertations. No. 1351, Linköping Univeristy.

Paoletti, S., Juloski, A., Ferrari-Trecate, G., and Vidal, R. (2007). Identification of hybrid systems: a tutorial. *European Journal of Control*, 13(2-3), 242–260.

Roll, J. (2003). *Local and Piecewise Affine Approaches to System Identification*. Linköping studies in science and technology. thesis no 802, Linköping University.

Roll, J., Bemporad, A., and Ljung, L. (2004). Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1), 37 – 50.

Taghirad, H.D. and Bélanger, P.R. (1998). Modeling and parameter identification of harmonic drive systems. *Journal of Dynamic Systems, Measurement, and Control*, 120(4), 439–444.

Vidal, R., Soatto, S., Ma, Y., and Sastry, S. (2003). An algebraic geometric approach to the identification of a class of linear hybrid systems. In *Proceedings of the 42nd IEEE Conference on Decision and Control (CDC)*, 167–172. Hawaii, USA.