Recursive Identification Method for Piecewise ARX Models: A Sparse Estimation Approach

Per Mattsson, Dave Zachariah, and Petre Stoica

***Abstract*—This paper deals with the identification of nonlinear systems using piecewise linear models. By means of a sparse overparameterization, this challenging problem is turned into a convex optimization problem. The proposed method uses a likelihood-based methodology which adaptively penalizes model complexity and directly leads to a recursive implementation. In this sparse estimation approach, the tuning of user parameters is avoided, and the computational complexity is kept linear in the number of data samples. Numerical examples with both simulated and experimental data are presented and the results are compared with previously published methods.**

***Index Terms*—System identification, nonlinear dynamical system, piecwise linear approximations.**

I. Introduction

IN this paper we consider the problem of identifying a nonlinear model for a scalar dynamical system using a finite record of its output $y(t)$ and input $u(t)$. A broad range of nonlinear dynamical systems can be modeled as

$$y(t) = f(\phi(t)) + e(t), \tag{1}$$

where $\phi(t)$ is a function of past inputs and outputs, $e(t)$ is a white noise process, and $f(\cdot)$ is the unknown function of interest. A common choice for the regression vector $\phi(t)$ is

$$\begin{aligned}\phi(t) = [-y(t-1) \quad \cdots \quad &-y(t-n_a) \\ u(t-1) \quad \cdots \quad &u(t-n_b) \quad 1]^\top.\end{aligned} \tag{2}$$

The model (1) together with the regressor (2) is called a *n*onlinear *a*utog*r*essive e*x*ogenous (NARX) model. The special case when $\phi(t)$ depends only on past outputs $\{y(t-k)\}$ is called nonlinear AR, and when $\phi(t)$ depends only on past inputs $\{u(t-k)\}$ we get a nonlinear finite impulse response (NFIR) model [1]. In this paper we focus on the NARX model structure, but the proposed method can be applied to any regressors $\phi(t)$ that can be computed using data collected until time $t-1$.

The identification problem is then to find the unknown function $f(\cdot)$. In nonlinear black-box modelling, the function $f(\cdot)$ is usually seen as a one-step-ahead predictor for the output $y(t)$ given data collected until time $t-1$ [2]. In this framework we want to find a good predictor model of the underlying system.

If $f(\cdot)$ was allowed to be any function, the identification problem would be very challenging. For this reason, $f(\cdot)$ is usually restricted to some class of functions, such as block-oriented models in which linear dynamics are mixed with static nonlinearities [3], [4]. Such approaches commonly parameterize the nonlinearities using basis functions. An alternative is to linearize $f(\cdot)$ at each given point $\phi(t)$, using a set of training points [5].

A related approach is to approximate $f(\cdot)$ with a piecewise affine function. These functions are known for their universal approximation properties [6], [7], and are therefore popular in system identification. Using a piecewise affine function in (1) and (2) gives the flexible class of *p*iece*w*ise ARX models (PWARX). In such models, the space in which $\phi(t)$ resides is partitioned into separate regions and a local linearization of $f(\cdot)$ is used for each region. These models are also useful for systems that change their modes, e.g., due to saturations, and they have been shown to be useful for both prediction and control of nonlinear systems [8], [9]. Note also that applying a piecewise affine function to the regressor $\phi(t)$ can be viewed as segmented linear regression [10].

Even if the identification of (1) becomes simpler when we restrict $f(\cdot)$ to be a piecewise affine function, this is still a complex task. In fact, it was shown in [11] that identifying $f(\cdot)$ by minimizing a loss function of the error $y(t) - f(\phi(t))$ is an NP-hard problem in general. The main problem here is that the regions and the model parameters in each region have to be estimated simultaneously. Extensive surveys of recent methods and results can be found in [12] and [13].

In [14], it was shown how the PWARX identification problem, under relatively mild conditions, can be reformulated as a mixed-integer linear or quadratic program. However, even though heuristic algorithms for solving such a program exist, the associated problem is still NP-hard, and can be prohibitively time consuming for larger data sets. In [15] and algebraic approach was developed for the noise-free case, but the resulting method is rather sensitive to noise compared to other approaches. To deal with noisy data a bounded-error approach was proposed in [16], which decides the number of regions by a user-specified bound on the prediction error. This results in an NP-hard optimization problem, which is solved using a greedy algorithm that finds a suboptimal solution.

If the regions are given, the problem is reduced to finding the linear submodels for each region. Different heuristics for an initial clustering of the data into regions have been suggested in the literature, such as the k-means like method in [17], and an expectation maximization method in [18]. In [19]

Manuscript received November 09, 2015; revised April 01, 2016 and June 20, 2016; accepted July 02, 2016. Date of publication July 27, 2016; date of current version August 11, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tongtong Li. This work was supported in part by the Swedish Research Council under Contract 621-2014-5874.

The authors are with the Department of Information Technology, Uppsala University, Uppsala 75124, Sweden (e-mail: per.mattsson@it.uu.se; dave.zachariah@it.uu.se; ps@it.uu.se).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2016.2595487

a Bayesian approach was used that alternates between updating the submodels and assigning new samples to each cluster in a greedy manner. A similar approach was taken in [20], where a recursive method was developed in which each new sample is assigned to a region and then the corresponding parameters are locally optimized. Common to these methods is that they find suboptimal solutions, and that good initializations and choice of user parameters are typically needed in order to get a good estimate.

The approach recently proposed in [21] estimates a linear submodel for each observation and penalizes the number of unique submodels. Such a penalization leads to a regularized nonconvex optimization problem, but a convex relaxation using a weighted sum of norms was proposed to tackle it. The unique solution to the relaxed problem is, however, highly dependent on carefully selecting the regularization parameters. Furthermore, as the number of model parameters increases with the number of data points, the computational requirements become prohibitive for large data records.

The method proposed in this paper is based on selecting a set of linearization points of the nonlinear system (1) and then identifying a corresponding set of locally linear model parameters. The method has the following features:

- It is based on a convex optimization problem.
- The problem is statistically motivated and tuning-parameter free.
- The solution can be computed recursively.

Jointly these features address important limitations of the aforementioned existing methods. The set of linearization points, which form the regions of the local linear models, are selected using a data-adaptive clustering technique. This is similar to the approach in [17] and [18]. However, while those methods are constructed for cases with few clusters, the proposed method uses a likelihood-based approach, in which regions with similar dynamics are automatically penalized and pruned out—thus allowing the user to initially overparameterize the model. At the same time, this approach eliminates the need for carefully tuned user parameters as in e.g. [21]. The proposed method automatically identifies a predictive PWARX model of the nonlinear system after selecting the model order and the number of linearization points. Furthermore, the resulting convex problem is solved with a complexity that grows linearly with the number of data points and the solution method is therefore well-equipped to tackle large datasets.

The paper is organized as follows. The model and problem formulation are presented in Section II, followed by a discussion about selecting the linearization points in Section III. The proposed identification method is presented in Section IV together with a discussion about different regularization techniques. A summary of the proposed method is presented in Section V. Finally, in Section VI, the proposed method is tested on both simulated and real data sets.

Remark: In the interest of reproducible research we have made the MATLAB code for the proposed method available at `http://www.it.uu.se/katalog/davza513`.

Notation: $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_F$ denote the ℓ_1, ℓ_2 and Frobenius norms, respectively. $x \odot y$ is the elementwise (Hadamard) product between vectors x and y. Finally, X^- denotes the generalized inverse of matrix X.

Abbreviations: Autoregressive with exogenous input (ARX), nonlinear ARX (NARX), piecewise ARX (PWARX), least-squares (LS), least absolute shrinkage and selection operator (LASSO), maximum aposteriori (MAP).

II. The PWARX Model

Let us first consider an affine ARX model, i.e.,

$$y(t) = -\sum_{i=1}^{n_a} a_i y(t-i) + \sum_{j=1}^{n_b} b_j u(t-j) + c + e(t), \quad (3)$$

where $\{a_i\}$ and $\{b_j\}$ are the model coefficients, c is a constant and $e(t)$ is a zero-mean white process with unknown variance σ^2. The affine equation (3) can also be written in a linear regression form [22], i.e.,

$$y(t) = \phi^\top(t)\vartheta + e(t), \quad (4)$$

where

$$\vartheta = \begin{bmatrix} a_1 & \cdots & a_{n_a} & b_1 & \cdots & b_{n_b} & c \end{bmatrix}^\top \quad (5)$$

is a vector of $d = n_a + n_b + 1$ parameters and $\phi(t)$ is given in (2). For large enough n_a and n_b, ARX models can be used to approximate any linear system [23].

Affine system models as in (4) are also useful as local approximations of nonlinear systems, but they cannot capture nonlinear dynamics. However, if the parameter vector ϑ is allowed to depend on the region in the regressor space $\mathbb{R}^d$ to which $\phi(t)$ belongs, then models with good approximation properties can be constructed [6], [7], [20]. Models of this type are called PWARX.

In order to formally define PWARX models, partition the regressor space $\mathbb{R}^d$ into n_r regions $\mathcal{R}_1, \dots, \mathcal{R}_{n_r}$, and let (n_a, n_b) be the maximum model orders for all regions. Then the PWARX model can be expressed as

$$y(t) = \phi^\top(t)\vartheta_i + e(t), \quad \text{if } \phi(t) \in \mathcal{R}_i, \quad (6)$$

where the parameter vector ϑ_i describes the dynamics in region $\mathcal{R}_i$. This is a nonlinear model that is piecewise affine in the regressor space.

Even though (6) is a nonlinear model, it can be formulated as a linear regression if the regions $\mathcal{R}_i$ are given. This is done by stacking the parameter vectors on top of each other, i.e.,

$$y(t) = \varphi^\top(t)\theta + e(t), \quad (7)$$

where

$$\theta = \begin{bmatrix} \vdots \\ \vartheta_i \\ \vdots \end{bmatrix} \in \mathbb{R}^{n_r d}, \qquad \varphi(t) = \begin{bmatrix} \vdots \\ f_i\left(\phi(t)\right) \\ \vdots \end{bmatrix} \in \mathbb{R}^{n_r d}, \quad (8)$$

and f_i is an indicator function

$$f_i\left(\phi(t)\right) = \begin{cases} \phi(t) & \text{if } \phi(t) \in \mathcal{R}_i \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

Given N samples of $y(t)$ and $u(t)$, the problem is to find the parameter vector θ in (7). However, in general the regions $\mathcal{R}_i$ are not given beforehand. This is certainly the case when the PWARX model is used to approximate a general nonlinear system. To tackle this practical case, one approach is to overparameterize the model in (7), by choosing n_r to be large and thus yielding a fine partitioning of the regressor space. This approach is pursued here.

In an overparameterized model, the standard least-squares (LS) method is inadequate without regularization. Different regularization approaches are discussed in Section IV, where a recursive user parameter-free method is proposed.

Remark: In this paper we focus on the PWARX model structure. However, the method proposed in Section V is not restricted to regressors of the form (2). In fact, any regressor $\phi(t)$ that can be computed from $y(1), \dots, y(t-1)$ and $u(1), \dots, u(t-1)$ can be used.

III. Selection of the Linearization Points

In [20], it was shown how a general nonlinear ARX (NARX) model (1) can be approximated by a PWARX model. The strategy is to linearize $f(\phi)$ around a set of points $\mu_1, \dots, \mu_{n_r} \in \mathbb{R}^d$, and then let the linearized submodel around μ_i be used in the region

$$\mathcal{R}_i = \left\{ \phi \in \mathbb{R}^d \mid \|\phi - \mu_i\|_2 \le \|\phi - \mu_j\|_2, \forall j \right\}, \tag{10}$$

which is a convex polyhedron. The linearization point μ_i is the centroid of the region $\mathcal{R}_i \subset \mathbb{R}^d$. Hence, in order to determine the regions $\mathcal{R}_1, \dots, \mathcal{R}_{n_r}$ it is sufficient to choose the linearization points of the model.

In general, however, there is no prior information about how many linearization points that are needed in order to get a good approximation of the NARX model. As mentioned in Section II, the approach taken in this contribution is to choose a 'large' n_r. A reasonable choice is to let the number of parameters $n_r d$ be of the same order as N. If $n_r d$ exceeded N the identification problem would become underdetermined. Note also that a practical upper limit on n_r is set by the computational requirements of the identification method.

The next problem is to decide around which points the model should be linearized, i.e., where to place μ_i. This can be done in several ways. One approach is to let each observed regression vector $\phi(t)$ be a linearization point, thus creating one region for each observation (hence $n_r = N$), cf. [21]. In this case the number of parameters $n_r d$ to estimate will grow linearly with N, which renders the identification problems intractable for large datasets.

For a fixed n_r, an alternative approach is to arrange the linearization points μ_i in a uniform lattice covering the interesting part of the regressor space, thus giving n_r rectangular linearization regions. However, such a partitioning does not take into account that different parts of the regressor space will contain more data than others, and hence they will be more informative.

A common approach to cluster data is to use k-means clustering [24]. In this approach, the observed regression vectors $\phi(t)$ are clustered into n_r sets $\{S_1, \dots, S_{n_r}\}$, that are obtained by solving the following problem:

$$\min_{S_1, \dots, S_{n_r}} \sum_{i=1}^{n_r} \sum_{\phi(t) \in S_i} \|\phi(t) - \mu_i\|_2^2,$$

where the linearization points $\{\mu_i\}$ are the means of each set of discrete points S_i, i.e.,

$$\mu_i = \frac{1}{|S_i|} \sum_{\phi(t) \in S_i} \phi(t).$$

This is an NP-hard problem, but efficient heuristic algorithms exist which scale well with the dataset size [25]–[27]. The regions are then determined by the resulting linearization points together with (10). Using k-means clustering leads to a data-adaptive partioning of the regressor space.

Alternatives to the k-means approach include variants of hierarchical clustering and the k-harmonic means method [28], [29]. Hierarchical clustering provides clusters with varying granularity but is more computationally complex than the k-means approach.

Remark 1: If the observed data were generated by a PWARX model, with the number of regions being known, the regions found by e.g. k-means usually would not be the same as the true regions. For this reason it is desirable to use a fine partitioning, i.e. choose n_r to be significantly larger than the true number of regions.

Remark 2: Using rectangular linearization regions $\mathcal{R}_i$ simplifies the implementation of the identification methods discussed below. When k-means is used, this can be achieved by performing the clustering in each dimension of $\mathbb{R}^d$ separately. This method has been used in the numerical examples below.

IV. Identification Method

By stacking N samples of $y(t)$ into a vector y and the corresponding regressor row vectors $\varphi^\top(t)$ into an $N \times n_r d$ matrix Φ, we can write (7) as

$$y = \Phi\theta + e,$$

where e is the noise with $\mathrm{E}_e[e] = 0$ and $\mathrm{E}_e[ee^\top] = \sigma^2 I_N$, and θ contains the $n_r d$ parameters of interest. A wide class of tractable parameter estimators is obtained by the minimization of a sum of two scalar cost functions

$$\hat{\theta} = \arg\min_{\theta} \; V_r(\theta) + V_c(\theta), \tag{11}$$

where $V_r(\theta)$ is a cost of the residuals, $y - \Phi\theta$, and $V_c(\theta)$ is a cost of the model complexity. The complexity can be quantified in various ways, taking into account the model order as well as the number of distinct regions.

A. Deterministic Approaches

We first consider θ to be an unknown deterministic parameter vector. Then the well-studied LS method is obtained by setting

$$V_r(\theta) = \|y - \Phi\theta\|_2^2, \quad V_c(\theta) = 0,$$

in (11). Under favourable conditions with a sufficient number of samples in each region $\mathcal{R}_j$, and well-exciting signals, the regressor matrix Φ has full rank, yielding a unique solution [22]. For the model under consideration, however, $n_r d$ can be large making it difficult or impossible to guarantee that there are $N \geq n_r d$ samples covering all regions. In such cases the LS method is inadequate, producing either a nonunique solution or estimation errors with high variance due to overfitting.

By introducing a cost of complexity, the LS method can be biased towards some prior knowledge of the parameter vector so as to alleviate the problem of overfitting. The standard regularized least-squares (REG-LS) sets

$$V_r(\theta) = \|y - \Phi\theta\|_2^2, \quad V_c(\theta) = \|\theta\|_\Lambda^2,$$

where $\Lambda \succ 0$ is a weighting matrix. Alternatively, for the LASSO method

$$V_r(\theta) = \|y - \Phi\theta\|_2^2, \quad V_c(\theta) = \lambda\|\theta\|_1,$$

where $\lambda > 0$ is a weight. Both REG-LS and LASSO are convex problems and they can be solved by means of a recursive implementation [22], [30]–[32]. Their cost functions $V_c(\theta)$ penalize coefficients in $\hat{\theta}$ that deviate from 0. This may be suitable for penalizing the model order of ARX but is not appropriate for the PWARX model under consideration since the parameter vector (8) is not sparse.

A more appropriate cost of complexity for PWARX models was suggested in [21]. The idea is to exploit the fact that in a finely partitioned regressor space, neighbouring regions $\mathcal{R}_i$ are likely to exhibit similar dynamics. In [21] the regions are chosen in such a way that there is only one observed regressor $\phi(t)$ for each region $\mathcal{R}_i$. Hence, it is reasonable to assume that for many pairs of regions, the corresponding parameter vectors should be nearly the same, i.e., $\|\vartheta_i - \vartheta_j\|_2$ is close to zero. The method proposed in [21] makes use of a sum-of-norms regularization of the least-squares method (SNR-LS), that penalizes the weighted norms of all pairwise differences $\|\vartheta_i - \vartheta_j\|_2$, and it corresponds to:

$$V_r(\theta) = \|y - \Phi\theta\|_2^2, \tag{12}$$

$$V_c(\theta) = \lambda \sum_{i=1}^{n_r}\sum_{j=1}^{n_r} K(i,j)\|\vartheta_i - \vartheta_j\|_2, \tag{13}$$

where $K(\cdot,\cdot)$ is some user-defined kernel and $\lambda > 0$ is a weight.

Unlike for REG-LS and LASSO, there is no recursive implementation of SNR-LS available in the literature which renders it computationally intractable for large N. Furthermore, the above three regularized identification methods share a central drawback in that they require the user to tune a set of parameters, which can be somewhat impractical.

B. Stochastic Approaches

We now consider θ to be a random variable with a distribution $p_\theta(\theta)$ such that $\mathrm{E}_\theta[\theta] = \mu_\theta$ and $\mathrm{Cov}_\theta[\theta] = P_\theta \succ 0$. We also consider a distributional form of the independent white process $e(t) \sim p_e(e(t)|y_{t-1})$, where $y_{t-1} = \{y(k)\}_{k=1}^{t-1}$. Here we assume Gaussian distributions, due to the resulting tractability and robustness of the estimator [33], [34].

The maximum a posterior (MAP) estimator is obtained by maximizing $p(\theta|y)$ [35]. We note that

$$p(y(t)|y_{t-1},\theta) = p_e(y(t) - \varphi^\top(t)\theta|y_{t-1}).$$

By a recursive application of the chain rule we therefore obtain the likelihood function

$$\begin{aligned} p(y|\theta) &= \prod_{t=1}^{N} p(y(t)|y_{t-1},\theta) \\ &= \prod_{t=1}^{N} p_e(y(t) - \varphi^\top(t)\theta|y_{t-1}), \end{aligned} \tag{14}$$

where

$$p_e(y(t) - \varphi^\top(t)\theta|y_{t-1}) \propto \exp\left(-\frac{(y(t) - \varphi^\top(t)\theta)^2}{2\sigma^2}\right).$$

The logarithm of the sought posterior distribution equals

$$\begin{aligned} \ln p(\theta|y) &= \ln p(y|\theta) + \ln p_\theta(\theta) + K_1 \\ &= -\sum_{k=1}^{N} \frac{(y(t) - \varphi^\top(t)\theta)^2}{2\sigma^2} \\ &\quad - \frac{1}{2}(\theta - \mu_\theta)^\top P_\theta^{-1}(\theta - \mu_\theta) + K_2 \\ &= -\frac{1}{2}\left(\sigma^{-2}\|y - \Phi\theta\|_2^2 + \|\theta - \mu_\theta\|_{P_\theta^{-1}}^2\right) + K_3, \end{aligned} \tag{15}$$

where the K : s are constant w.r.t θ. Therefore the maximization of (15) is equivalent to (11) with

$$V_r(\theta) = \sigma^{-2}\|y - \Phi\theta\|_2^2, \quad V_c(\theta) = \|\theta - \mu_\theta\|_{P_\theta^{-1}}^2. \tag{16}$$

The parameters σ^2, μ_θ and P_θ are however unknown. These would therefore be tuning parameters in any practical scenario. We now propose a tractable and statistically motivated method to automatically estimate the unknown parameters.

C. Proposed Method

The MAP estimator can be expressed in an alternative way, by means of a reparameterization of θ. Let the parameters corresponding to a specific linearization region $\mathcal{R}_\star$ be denoted as $\vartheta_\star$. Consider this region to be the reference and let the parameters of the remaining regions be formed by a set of differences $\{\delta_j\}_{j=1}^{q}$ from $\vartheta_\star$. Here we consider $q = n_r - 1$, although other alternatives are possible. It is reasonable to assume a priori that the reference $\vartheta_\star$ and the q differences $\delta = [\delta_1^\top, \ldots, \delta_q^\top]^\top$ are uncorrelated.

Any given ordering of the regions is possible. In general, θ, δ and $\vartheta_\star$ are related by a given linear transformation D, such that

$$\theta = D\begin{bmatrix} \vartheta_\star \\ \delta \end{bmatrix},$$

and correspondingly

$$\mu_\theta = D\begin{bmatrix} \mu_\star \\ \mu_\delta \end{bmatrix}, \quad P_\theta = D\begin{bmatrix} P_\star & 0 \\ 0 & P_\delta \end{bmatrix} D^\top.$$

As an example, consider $q = n_r - 1$ incremental differences and let $\vartheta_\star = \vartheta_1$ so that we can write the parameters for each region as a cumulative sum

$$\begin{aligned} \vartheta_2 &= \vartheta_\star + \delta_1 \\ \vartheta_3 &= \vartheta_\star + \delta_1 + \delta_2 \\ &\vdots \\ \vartheta_{n_r} &= \vartheta_\star + \delta_1 + \delta_2 + \cdots + \delta_{n_r-1}. \end{aligned}$$

For this example the incremental difference matrix becomes

$$D = \begin{bmatrix} I & 0 & 0 & \cdots & 0 \\ I & I & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & & 0 \\ I & I & I & \cdots & I \end{bmatrix}.$$

Given the general linear relation between θ, δ and $\vartheta_\star$, the MAP estimates $\hat{\vartheta}_\star$ and $\hat{\delta}$ are obtained as:

$$\begin{aligned} \min_{\vartheta_\star,\,\delta} \; & \sigma^{-2} \left\| y - \Phi D \begin{bmatrix} \vartheta_\star \\ \delta \end{bmatrix} \right\|_2^2 \\ & + \|\vartheta_\star - \mu_\star\|^2_{P_\star^{-1}} + \|\delta - \mu_\delta\|^2_{P_\delta^{-1}}, \end{aligned} \tag{17}$$

and they yields an estimate θ

$$\hat{\theta} = D \begin{bmatrix} \hat{\vartheta}_\star \\ \hat{\delta} \end{bmatrix}.$$

When D is an invertible transformation, as in the example above, $\hat{\theta}$ is equivalent to the MAP estimator given by (11) and (16).

We consider the following model choices for the statistical moments:

(i) No prior knowledge of the reference $\vartheta_\star$ is assumed. This ignorance can be modeled using a noninformative prior $p_\vartheta(\vartheta_\star) = \text{const.}$, viz. by setting $P_\star = \lambda I_d$ with $\lambda \to \infty$, so that $P_\star^{-1} \to 0$ and $\mu_\star$ is eliminated from (17).

(ii) Given the assumption that most regions share similar dynamics, prior knowledge on the cumulative differences $p_\delta(\delta)$ takes the form of a prior mean $\mu_\delta = 0$ with unknown covariance P_δ.

(iii) For reasons of parsimony, we let P_δ be diagonal, i.e. assuming no correlations between the differences prior to observing the data. Note that P_δ is an auxiliary variable, irrespective of any ‘true’ covariance of δ which is naturally unknown.

Let $\Phi D = [F \; G]$ where the blocks F and G contain the data and correspond to $\vartheta_\star$ and δ, respectively. Given the lack of prior knowledge on $\vartheta_\star$ (see (i) above), the minimizers in (17) can be written as

$$\begin{aligned} \hat{\vartheta}_\star &= (F^\top C^{-1} F)^{-} F^\top C^{-1} y \\ \hat{\delta} &= P_\delta G^\top C^{-1} (y - F\hat{\vartheta}_\star), \end{aligned} \tag{18}$$

where

$$C \triangleq G P_\delta G^\top + \sigma^2 I_N.$$

Note that $\hat{\delta}$ is always unique and $\hat{\vartheta}_\star$ is unique when F has full rank [36].

Following [37], [38], the unknown covariance parameters σ^2 and P_δ can be estimated jointly by maximizing the marginal likelihood function

$$p(y|\vartheta_\star; \sigma^2, P_\delta) = \int p(y|\vartheta_\star, \delta; \sigma^2) p_\delta(\delta; P_\delta) d\delta, \tag{19}$$

where $p(y|\vartheta_\star, \delta; \sigma^2)$ follows from (14). Then we obtain the following distributions

$$p(y|\vartheta_\star, \delta; \sigma^2) \propto \exp\left(-\frac{1}{2\sigma^2} \|y - F\vartheta_\star - G\delta\|_2^2 \right), \tag{20}$$

$$p_\delta(\delta; P_\delta) \propto \exp\left(-\frac{1}{2} \|\delta\|^2_{P_\delta^{-1}} \right). \tag{21}$$

After inserting the expressions into (19) we obtain, as we show in Appendix A

$$p(y|\vartheta_\star; \sigma^2, P_\delta) = \frac{1}{\sqrt{(2\pi)^N |C|}} \exp\left(-\frac{1}{2} \|y - F\vartheta_\star\|^2_{C^{-1}} \right).$$

Maximizing the marginal likelihood is equivalent to solving

$$\min_{\vartheta_\star, \sigma^2, P_\delta} \; (y - F\vartheta_\star)^\top C^{-1} (y - F\vartheta_\star) + \ln|C| \tag{22}$$

and results in estimates of P_δ and σ^2 which can be used in (17). While this method avoids a heuristic tuning of user parameters, it has multiple local optima issues and cannot readily be implemented recursively. We therefore consider linearizing the concave function $\ln|C|$.

For the sake of parsimony, choose $\tilde{P}_\delta = 0$ and an arbitrary variance $\tilde{\sigma}^2 = c$ as the linearization point. Then we obtain the first-order approximation

$$\ln|C| \simeq \frac{1}{c} \operatorname{tr}\{C\} + K, \tag{23}$$

where c and K are constants, as shown in Appendix B. The right hand side of (23) is convex in P_δ and σ^2, cf. [39], [40]. Note that this linearization is valid for all $\tilde{P}_\delta$ and $\tilde{\sigma}^2$ such that $C = cI$. After linearization, the following convex problem is obtained

$$\min_{\vartheta_\star, \sigma^2, P_\delta} \; (y - F\vartheta_\star)^\top C^{-1} (y - F\vartheta_\star) + \frac{1}{c} \operatorname{tr}\{C\}. \tag{24}$$

Lemma 1: The minimizer $\hat{\vartheta}_\star$ of (24) is invariant to the choice of $c > 0$.

Proof: See (18) and [40], [41]. ■

Theorem 1: The estimators in (18), using the covariance parameters obtained from (24), can be computed as the solution to the following augmented minimization problem

$$\begin{aligned} \min_{\vartheta_\star, \delta, \sigma^2, P_\delta} \; & \sigma^{-2} \|y - F\vartheta_\star - G\delta\|_2^2 + \|\delta\|^2_{P_\delta^{-1}} \\ & + \operatorname{tr}\{G P_\delta G^\top + \sigma^2 I_N\}. \end{aligned} \tag{25}$$

Proof: The minimizers $\hat{\vartheta}_\star$ and $\hat{\delta}$ in (25) equal the expressions in (18). After concentrating out δ in (25) one obtains (24) with $c = 1$. It follows from Lemma 1 that the minimizer $\hat{\vartheta}_\star$ of (25) is equal to that in (24) for any $c > 0$. ■

The minimizing covariance parameters in (25) are given in closed form as

$$\hat{\sigma}^2 = \|y - F\vartheta_\star - G\delta\|_2/\sqrt{N}$$
$$[\hat{P}_\delta]_{ii} = |[\delta]_i|/\|[G]_i\|_2, \; i = 1, \ldots, qd, \tag{26}$$

where $[\delta]_i$ and $[G]_i$ denote the ith element of δ and the ith column of G. Define the $(q+1)d$-dimensional parameter vector

$$\tilde{\theta} \triangleq \begin{bmatrix} \vartheta_\star \\ \delta \end{bmatrix} \text{ and } H \triangleq \Phi D.$$

Then inserting (26) into (17) yields a concentrated cost function that can be written in the compact form of (11), with

$$V_r(\tilde{\theta}) = \|y - H\tilde{\theta}\|_2, \quad V_c(\tilde{\theta}) = \|w \odot \tilde{\theta}\|_1, \tag{27}$$

and

$$w = \frac{1}{\sqrt{N}} \left[0, \ldots, 0, \|h_{d+1}\|_2, \, \|h_{d+2}\|_2, \, \ldots, \|h_{(q+1)d}\|_2\right]^\top.$$

Here h_i denotes the ith column of H. The estimator of θ using the likelihood-based approach is then given by $\hat{\theta} = D\hat{\tilde{\theta}}$. Note that the method based on (27) uses the norm of the residual vector as a cost, $V_r(\tilde{\theta})$, which is the square-root of the cost used in LS, REG-LS, LASSO and SNR-LS. Further, the cost of model complexity $V_c(\tilde{\theta})$ penalizes the parameter differences between the regions using an adaptively weighted ℓ_1-norm. Consequently, the method yields sparse estimates of δ with few elements significantly different from zero. That is, only few regions will have rather dissimilar dynamics.

V. SUMMARY OF THE PROPOSED METHOD

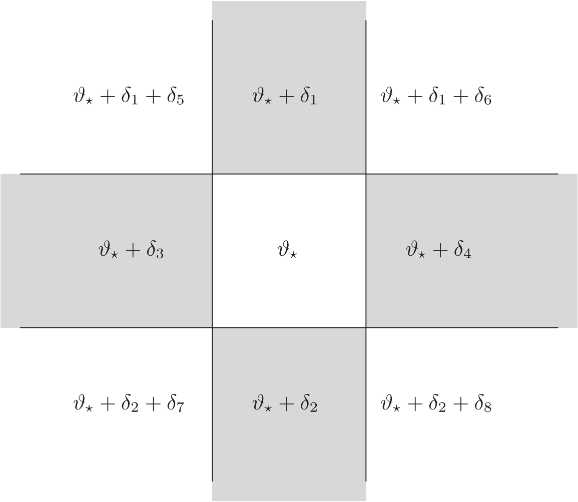
In this section, the resulting *r*ecursive *i*dentification method using *l*ocally *l*inearized models (RILL) is summarized (after forming the regressor vectors $\{\phi(t)\}_{t=1}^N$ from the data $\{u(t), y(t)\}_{t=1}^N$):

1) Set the model orders n_a and n_b and choose n_r.
2) Choose the n_r linearization points using $\{\phi(t)\}_{t=1}^N$.
3) Construct incremental difference matrix D.
4) Solve problem (11) with (27) recursively.
5) Finally compute $\hat{\theta} = D\hat{\tilde{\theta}}$.

The optimization problem in Step 4 requires no tuning parameters, and as shown in Section V-B, it can be solved recursively with the same computational complexity as REG-LS and LASSO, which is linear in N.

In Step 1, the model integer parameters have to be decided. One possible method for this is to perform cross-validation on the data set $\{u(t), y(t)\}_{t=1}^N$. That is, use the first N' samples to identify the model for a given triplet. Then, predict the output of the remaining $N - N'$ samples, $\hat{y}(t) = \varphi^\top(t)\hat{\theta}$, via (7) and choose the triplet which yields the minimum sum of squared output errors.

In Step 2–3, the linearization points and the difference matrix D have to be chosen. This can be done in many ways, but in Section V-A we discuss a standard way of doing this that reduces the number of choices that has to be made by the user.



$\vartheta_\star + \delta_1 + \delta_5$	$\vartheta_\star + \delta_1$	$\vartheta_\star + \delta_1 + \delta_6$
$\vartheta_\star + \delta_3$	$\vartheta_\star$	$\vartheta_\star + \delta_4$
$\vartheta_\star + \delta_2 + \delta_7$	$\vartheta_\star + \delta_2$	$\vartheta_\star + \delta_2 + \delta_8$

Fig. 1. Stylized example of the linearization regions and parameterization used, where $n_r = 9$ (3×3 grid). The middle region is chosen as the reference $\mathcal{R}_\star$ with $\vartheta_\star$, and the differences δ_j as illustrated.

A. Incremental Differences

In this paper we choose the regions as in (10), and thus Step 2 in the proposed method amounts to choosing a set of linearization points μ_i. In the case there is no prior knowledge to use in the selection of linearization points, we propose to place them in a lattice using k-means clustering in each dimension separately. This yields rectangular linearization regions, and as we will see this simplifies the selection of the difference matrix D. Also, using k-means ensures that the data is well spread across the regions.

In Step 3 the difference matrix D has to be determined. The parameterization we propose to use is illustrated in Fig. 1 for the case when the linearization regions are divided only with respect to the current input and output, i.e. $u(t)$ and $y(t)$. That is, let the middle region correspond to $\vartheta_\star$, and let the incremental differences δ_i extend either vertically or horizontally. This generalizes into higher dimensions if needed. Thus, the only choice the user has to make is the size of the grid and in which direction the differences should extend, where the latter choice is binary.

B. Recursive Formulation

A similar optimization problem to that considered here was solved numerically in [41] but for a different regression problem. This type of solving technique is employed in Step 4 above. The convex optimization problem given by (11) and (27) can be solved recursively via a cyclic minimization approach, which minimizes the cost function with respect to one variable $\tilde{\theta}_i$ at a time, while holding the remaining ones constant [42]. That is, we solve following the convex problem cyclically for $i = 1, \ldots, (q+1)d$:

$$\min_{\tilde{\theta}_i} \; \|\bar{y}_i - h_i\tilde{\theta}_i\|_2 + w_i|\tilde{\theta}_i|, \tag{28}$$

where $\bar{y}_i = y - \sum_{j\neq i} h_j\tilde{\theta}_j$. Let $\check{\tilde{\theta}}_i$ denote the current estimate. Define the quantities

$$\Gamma^N \triangleq H^\top H$$

$$\rho^N \triangleq H^\top y$$

$$\kappa^N \triangleq y^\top y.$$

Then the minimizer of (28) for $i = 1, \dots, d$ equals

$$\hat{\tilde{\theta}}_i = \frac{\zeta_i + \Gamma_{ii}^N \check{\tilde{\theta}}_i}{\Gamma_{ii}^N}, \tag{29}$$

where we define the cyclically computed variable

$$\zeta \triangleq \rho^N - \Gamma^N \check{\tilde{\theta}}.$$

For $i = d+1, \dots, (q+1)d$, the minimizer of (28) takes the form

$$\hat{\tilde{\theta}}_i = \begin{cases} \hat{r}_i e^{j\hat{\omega}_i}, & \text{if } \sqrt{N-1}\gamma_i > \sqrt{\alpha_i\beta_i - \gamma_i^2} \\ 0, & \text{else,} \end{cases} \tag{30}$$

via a reparameterization of $\tilde{\theta}_i$ in polar form, where

$$\alpha_i = \eta + \Gamma_{ii}^N (\check{\tilde{\theta}}_i)^2 + 2\check{\tilde{\theta}}_i \zeta_i$$

$$\beta_i = \Gamma_{ii}^N$$

$$\gamma_i = |\zeta_i + \Gamma_{ii}^N \check{\tilde{\theta}}_i|$$

$$\hat{r}_i = \frac{\gamma_i}{\beta_i} - \frac{1}{\beta_i}\left(\frac{\alpha_i\beta_i - \gamma_i^2}{N-1}\right)^{1/2}$$

$$\hat{\omega}_i = \arg(\zeta_i + \Gamma_{ii}^N \check{\tilde{\theta}}_i) \tag{31}$$

and

$$\eta \triangleq \kappa^N + \check{\tilde{\theta}}^\top \Gamma^N \check{\tilde{\theta}} - 2\check{\tilde{\theta}}^\top \rho^N.$$

Note that the following variables can be computed recursively

$$\Gamma^t \triangleq \Gamma^{t-1} + D^\top \varphi(t)\varphi^\top(t) D$$

$$\rho^t \triangleq \rho^{t-1} + D^\top \varphi(t) y(t)$$

$$\kappa^t \triangleq \kappa^{t-1} + y(t)^2,$$

where $t = 1, \dots, N$.

This enables the online computation of the estimate of $\tilde{\theta}$ for each new sample $y(t)$ and $\varphi(t)$, as summarized in Algorithm 1 where we have dropped the superindices for notational convenience and we replace N with t in (29)–(31). See [41] for further details. Algorithm 1 computes an estimate of $\tilde{\theta}$ for each t by cyclically minimizing (28). When new data $(y(t+1), \varphi(t+1))$ arrives, the quantities Γ, ρ, κ are recursively updated and the cyclic minimization can be performed for step $t+1$.

The total computational complexity is $\mathcal{O}(NLq^2d^2)$, where L is the number of iterations performed per sample. As $L \to \infty$, the estimate converges to the minimizer of the convex problem in (11) and (27) for each t [43]. In practice, however, even a small L works well since we cycle all parameters L times for each new data sample, so in total each parameter gets updated NL times. In numerical experiments it has been seen that $L = 1$ produces good results when N is sufficiently large. For small N, we found that increasing L to about 5 results in good estimates.

Algorithm 1: Recursive solution to (11) with (27).

```
Input: $y(t)$, $\varphi(t)$ and $\check{\tilde{\theta}}$
$\Gamma := \Gamma + D^\top \varphi(t)\varphi^\top(t) D$
$\rho := \rho + D^\top \varphi(t) y(t)$
$\kappa := \kappa + y(t)^2$
$\eta = \kappa + \check{\tilde{\theta}}^\top \Gamma \check{\tilde{\theta}} - 2\check{\tilde{\theta}}^\top \rho$
$\zeta = \rho - \Gamma \check{\tilde{\theta}}$
repeat
    $i = 1, \dots, (q+1)d$
    Compute scalars in (31)
    Compute $\hat{\tilde{\theta}}_i$ using (29) ($i \leq d$) otherwise (30)
    $\eta := \eta + \Gamma_{ii}(\check{\tilde{\theta}}_i - \hat{\tilde{\theta}}_i)^2 + 2(\check{\tilde{\theta}}_i - \hat{\tilde{\theta}}_i)\zeta_i$
    $\zeta := \zeta + [\Gamma]_i(\check{\tilde{\theta}}_i - \hat{\tilde{\theta}}_i)$
    $\check{\tilde{\theta}}_i := \hat{\tilde{\theta}}_i$
until number of iterations equals $L$
Output: $\hat{\tilde{\theta}}$
```

VI. Numerical Evaluation

RILL has been evaluated on several numerical examples. In Sections VI-C and VI-D, two simulated PWARX models are identified. In Sections VI-E and VI-F, real data from a pick-and-place machine and a water tank are considered. These latter examples illustrate the utility of the locally linearized submodels for identification of real nonlinear systems.

A. Performance Metric

In the first two examples of this section the data is generated by a PWARX model, and thus there are true regions $\mathcal{R}_1^0, \dots, \mathcal{R}_{n_r^0}^0$ and a true parameter vector θ_0. However, since the regions used in the identification part, $\mathcal{R}_1, \dots, \mathcal{R}_{n_r}$, are not the same as the true regions, and $n_r > n_r^0$, it is not possible to compare the parameter vector θ_0 and the identified parameter vector $\hat{\theta}$ directly. Instead we evaluate the identification methods with respect to the model output

$$\hat{y}(t;\theta) = \hat{\phi}^\top(t;\theta)\vartheta_i, \quad \text{if } \hat{\phi}(t) \in \mathcal{R}_i,\ i = 1, \dots, n_r$$

where

$$\hat{\phi}(t;\theta) = [-\hat{y}(t-1;\theta) \quad \cdots \quad -\hat{y}(t-n_a;\theta) \quad u(t-1) \quad \cdots \quad u(t-n_b) \quad 1]^\top$$

for $t > \max(n_a, n_b)$ and a given input signal $u(t)$. The performance is then evaluated by the sum of mean square errors on a validation dataset of T samples,

$$\text{MSE} = \sum_{t=1}^{T} \text{E}\left[(y(t) - \hat{y}(t;\hat{\theta}))^2\right].$$

For the sake of comparison we normalize the error as

$$\text{NMSE} = \frac{\text{MSE}}{\text{MSE}_0},$$

where MSE_0 is the mean square error corresponding to the true parameter vector and true regions $\mathcal{R}_i^0$, cf. [44],

$$\mathrm{MSE}_0 = \sum_{t=1}^{T} \mathrm{E}\left[(y(t) - \hat{y}(t;\theta_0))^2\right].$$

The expectations are evaluated numerically using 1000 Monte Carlo simulations.

For the real datasets considered below, NMSE is not defined because θ_0 does not exist nor can we evaluate the mean-square error. For these sets we use a metric that compares the output errors with those obtained using the empirical mean as a predictor, viz.

$$\mathrm{FIT} = 100\left(1 - \frac{\|y - \hat{y}\|_2}{\|y - \bar{y}\mathbf{1}\|_2}\right),$$

where $\hat{y}$ contains the model output, $\bar{y}$ is the empirical mean of y and $\mathbf{1}$ is a vector of ones.

B. Setup of Identification Methods

In this section we will describe how the numerical experiments were conducted. Three methods have been used: Rill, Snr-LS [21] and the affine Arx model in (3).

For Rill we follow the steps in Section V. In particular, we have chosen the linearization points and incremental differences as described in Section V-A. For all examples we have used a 9×9 grid of linearization points. Therefore only the model orders n_a, n_b and the vertical/horizontal orientation of the differences have to be chosen.

The Snr-Ls method in [21] uses a sum of-norm-regularization as in (12)–(13). For the kernel we used the one suggested in [21], i.e.,

$$K_\ell(i,j) = \begin{cases} 1, & \text{if } \phi(i) \text{ is one of the } \ell \text{ closest neighbors of} \\ & \quad \phi(j) \text{ among all observations,} \\ 0, & \text{otherwise.} \end{cases} \tag{32}$$

In this method the user has to specify the regularization parameter λ and ℓ. In each example we have manually tuned λ with respect to NMSE. The optimization problem has then been solved using a Cvx-based implementation [45], [46] provided by the authors of [21]. As the number of parameters to estimate increases with the number of observed data points N, we observed a rapid rise in the runtime of this algorithm. For $N > 650$, the Monte Carlo simulations required to evaluate the NMSE became intractable and for $N \geq 1000$ the memory requirement became infeasible. Therefore this method was only tested on smaller data sets. For Rill we observed a runtime that is linear in N as expected by analysis in Section V-B.

The last step in the Snr-Ls approach is to divide the regressor space into regions. The authors of [21] suggest using e.g. a support vector machine (Svm), but note that such an approach is not suitable for more complicated regions. We found that indeed Svm approach does not always yield a desired number of regions. Therefore we opted for using the more general nearest neighbour classifier [24].

TABLE I
NMSE in Example in Section VI-C

N	250	500	1000
Arx	4.96	4.91	4.87
Snr-Ls	1.76	1.57	–
Rill	1.61	1.33	1.25

The affine Arx models have been estimated by the standard Ls method [22].

Finally, an important part of the experimental setup is the choice of input signal. In identification of linear models the importance of persistent excitation is well understood, see e.g. [22]. A commonly used input signal for identification of linear models is a pseudorandom binary sequence (Prbs), which is a signal that shifts between two levels in a certain fashion. One reason for using a Prbs is that it has similar correlation properties to white noise [22].

For Pwarx identification it is also important that the signals significantly vary in amplitude, since otherwise most regions will have no data. In this way Pwarx identification has much in common with identification of Hammerstein and Wiener models [47], [48]. The problem with a Prbs sequence is that it is poorly distributed in amplitude. A remedy to this problem is to multiply the signal in each interval of constant level with a random uniformly distributed factor, cf. [49]. This type of input signal has previously been used in e.g. Wiener model identification, and is also used in three of the numerical examples in this section.

C. Hammerstein System

Consider the system

$$y(t) = -0.5y(t-1) - 0.1y(t-2) + v(t-1) + e(t), \tag{33}$$

where $v(t)$ is a saturated version of $u(t)$,

$$v(t) = \begin{cases} 1 & \text{if } u(t) \geq 1 \\ u(t) & \text{if } -1 \leq u(t) \leq 1 \\ -1 & \text{if } u(t) < -1. \end{cases} \tag{34}$$

Here $(n_a^0, n_b^0, n_r^0) = (2, 1, 3)$. This type of system, with a static nonlinear block followed by a linear dynamic block, is commonly referred to as a Hammerstein system. The input $u(t)$ was a zero-mean white Gaussian process with variance 4, and the process noise $e(t)$ was white Gaussian with variance 0.04. This same setup was used in [18] and [21]. The system was identified using Rill, Snr-Ls, and the affine Arx model. The model orders n_a, n_b where chosen equal to the true model orders for all methods.

For Rill, we used $n_r = 81$ regions for which the differences extend vertically, see Section VI-B.

For the Snr-Ls method we let $\ell = 8$ in (32), as in the corresponding example found in [21]. For the regularization weight, we chose $\lambda = 0.08$ which produced a lower NMSE than the value used in [21].

The NMSE was computed for N equal to 250, 500 and 1000. The results are shown in Table I. As noted in Section VI-B, we

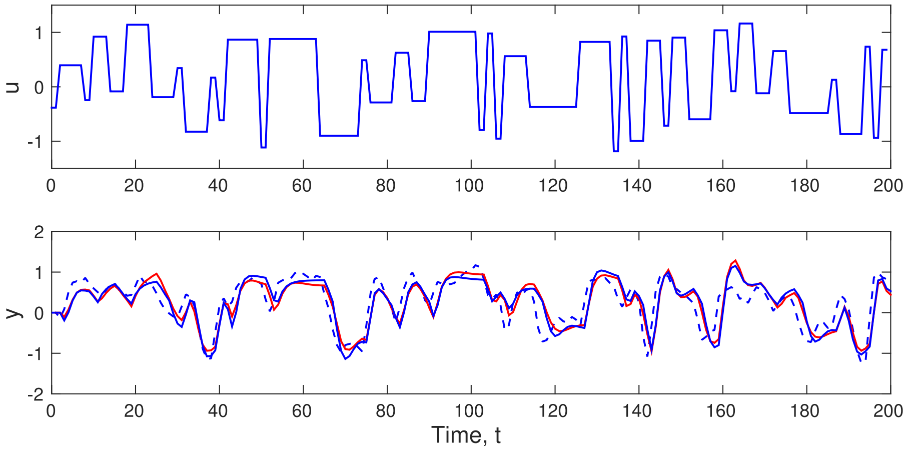


Fig. 2. A input-output realization of (35) with noise (blue dashed), without noise (blue solid); also the output of the model identified by RILL using $N = 500$ samples (red).

were unable to evaluate SNR-LS for $N \geq 650$. The ARX model is, as expected, outperformed by both SNR-LS and RILL. For RILL no particular tuning was used except for choosing the direction of the differences. Nevertheless, it performs better than SNR-LS. Moreover, if it is known that the system has a Hammerstein structure then this prior knowledge can be exploited by RILL by only partitioning the regressor space along the u-dimension. Using $n_r = 81$ as above, the NMSE of RILL is then reduced to 1.14 already for $N = 500$.

D. Piecewise Affine ARX System

For the Hammerstein system in example in Section VI-C the poles in each region of the regressor space are the same. By contrast, consider the following PWARX system,

$$y(t) = \begin{cases} y(t-1) - 0.5y(t-2) & \text{if } y(t-1) \leq 0.3 \\ \quad +0.5v(t-1) + e(t), & \\ 1.2y(t-1) - 0.35y(t-2) & \text{if } y(t-1) > 0.3 \\ \quad +0.15v(t-1) + e(t) & \end{cases} \tag{35}$$

where $v(t)$ is again a saturated version of $u(t)$,

$$v(t) = \begin{cases} 0.8 & \text{if } u(t) \geq 0.8 \\ u(t) & \text{if } -0.8 \leq u(t) \leq 0.8 \\ -0.8 & \text{if } u(t) < -0.8. \end{cases}$$

Here $(n_a^0, n_b^0, n_r^0) = (2, 1, 6)$. Note that both linear subsystems in (35) have a static gain equal to one, but the poles are real for $y(t) \geq 0.3$ and complex when $y(t)$ goes below 0.3. In the simulations, $e(t)$ was chosen as white Gaussian noise with variance 0.01. The input signal $u(t)$ was chosen as a modified PRBS, as described in Section VI-B.

The model orders n_a, n_b where chosen equal to the true model orders for all three identification methods. For RILL, we used $n_r = 81$ linearization points, for which the differences extend horizontally, see Section VI-B. For the SNR-LS method we let $\ell = 8$ in (32), and tuned the regularization weight to $\lambda = 0.05$.

TABLE II
NMSE IN EXAMPLE IN SECTION VI-D

N	250	500	1000
ARX	2.55	2.17	2.00
SNR-LS	1.25	1.18	–
RILL	1.18	1.10	1.05

The results for N equal to 250, 500 and 1000 are shown in Table II. As noted in Section VI-B, we were unable to evaluate SNR-LS for $N \geq 650$. As in example in Section VI-C, the affine ARX model is outperformed by both SNR-LS and the RILL. Similarly, RILL performs better than SNR-LS. Note that in both examples we obtain similar performance as SNR-LS with $N = 500$ using only $N = 250$ data samples.

Fig. 2 shows a realization of (35) together with the model output using the parameters identified with RILL using $N = 500$ samples. For the sake of clarity we also show the same output realization when there is no process noise, it can be seen that the identified model follows the noise-free output quite well.

E. Pick-and-Place Machine

In this example, a pick-and-place machine is studied. This machine is used to place electronic components on a circuit board, and is described in detail in [50]. The machine can be in several different modes, with two major modes being the free mode and the impact mode. In the free mode, the machine is carrying an electronic component, but is not in contact with the circuit board. When the electronic component gets in contact with the circuit board the system switches to impact mode. Besides these modes, the system could also exhibit saturation, etc. These characteristics of the machine have made it a popular choice for studying identification methods for PWARX systems. The data used here are from a real physical process, and were also used in e.g. [16], [21], [51]. The data set consists of a 15 s recording of the voltage input $u(t)$ and the vertical position of the mounting head $y(t)$. The data were sampled at 50 Hz, and the first 8s were used for identifying the model and the last 7 s for validation.

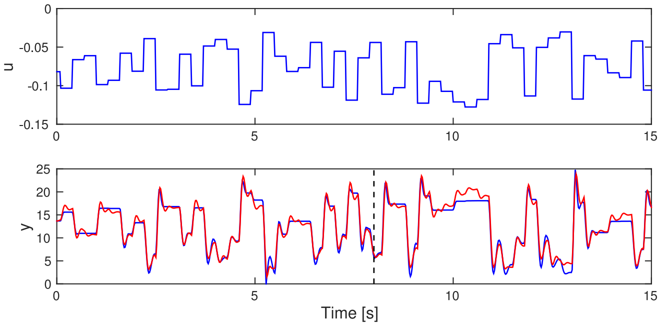


Fig. 3. The input/output data (blue) for example in Section VI-E plotted together with the output of the model (red) identified by RILL. The system was identified using the first 8 seconds of data.

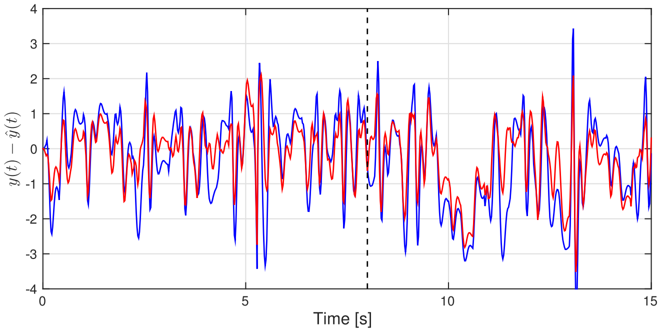


Fig. 4. Output error for example in Section VI-E, both for the ARX model identified using LS (blue) and the PWARX model identified by RILL (red).

The order of the PWARX model was set to $n_a = 2$ and $n_b = 2$ as in [21] and for RILL we used $n_r = 81$ linearization points, for which the differences extend horizontally, cf. Section VI-B. The input/output data are shown, together with the output of the identified model, in Fig. 3. The fit to the validation data was 79.4% for RILL, which is slightly better than the one of 78.6% reported in [21]. These results can be compared to the fit achieved with an affine ARX model of the same order, which was 73.2%. See Fig. 4 for a comparison of the output errors.

F. Tank Process

In this example a cascade tank process is studied. It consists of two tanks mounted on top of each other, with free outlets. The top tank is fed with water by a pump. The input signal is given by the voltage applied to the pump, and the output consists of the water level in the lower tank. The setup is described in more detail in [49].

The input signal for the system was a modified PRBS as described in Section VI-B. The data set consists of 2500 samples collected every five seconds. The identification was performed using $n_a = 4$ and $n_b = 2$ and $n_r = 81$ linearization points, for which the differences extend horizontally, cf. Section VI-B. The input/output data are shown, together with the output of the identified model, in Fig. 5. The first 1250 samples where used for identification, and the last 1250 samples for validation. The fit to the validation data was 86.9% for RILL, which can be compared to the fit achieved with an affine ARX model which was 77.4%. See Fig. 6 for a comparison of the output errors.

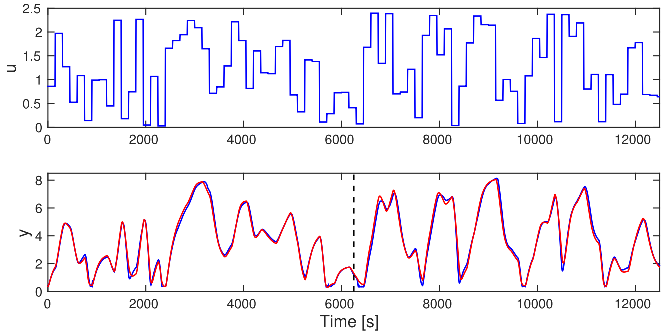


Fig. 5. The input/output data (blue) for example in Section VI-F plotted together with the output of the model identified by the RILL (red). The system was identified using the first 6250 seconds of data.

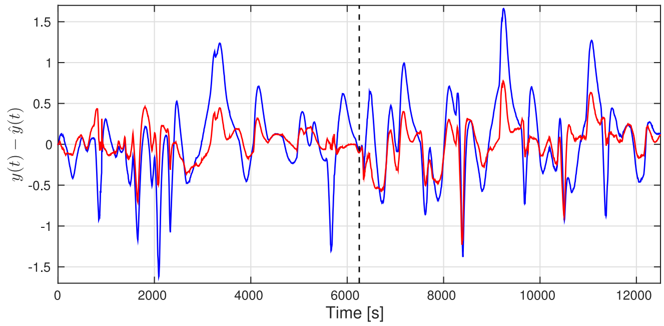


Fig. 6. Output error for example in Section VI-F, both for the ARX model identified using LS (blue) and the PWARX model identified by RILL (red).

VII. CONCLUSION

In this work we considered identification of nonlinear systems using piecewise linear models. We developed a recursive method which solves a statistically motivated convex optimization problem, avoids the tuning of user parameters, and has a computational complexity that is linear in the number of data samples. The proposed method uses a likelihood-based methodology which adaptively penalizes the complexity of a overparameterized sparse model. Both simulated and experimental data were used to evaluate the proposed method and the results showed that the method is a good candidate for application to a wide range of nonlinear systems, including but not confined to piecewise linear systems.

APPENDIX A
DERIVATION OF DISTRIBUTION

The sought-after distribution $p(y;\sigma^2, P_\delta)$ is given by (19)–(21). Note that $p(y|\vartheta_\star, \delta; \sigma^2)$ is not Gaussian since F and G depends on y. We have

$$\begin{aligned}
&\ln(p(y|\vartheta_\star, \delta; \sigma^2)p_\delta(\delta; P_\delta)) \\
&\quad = \alpha - \frac{1}{2}\left(\frac{1}{\sigma^2}\|y - F\vartheta_\star - G\delta\|_2^2 + \|\delta\|_{P_\delta^{-1}}\right) \\
&\quad = \alpha - \frac{1}{2}\left(\|\delta - \mu\|_{\Sigma^{-1}}^2 + \|y - F\vartheta_\star\|_{C^{-1}}^2\right)
\end{aligned}$$

where the matrix inversion lemma is utilized to show the second equality. Here

$$\alpha = -\frac{1}{2}\ln((2\pi)^{N+q}(\sigma^2)^N|P_\delta|)$$
$$C = GP_\delta G^\top + \sigma^2 I$$
$$\Sigma = \left(P_\delta^{-1} + \sigma^{-2}G^\top G\right)^{-1}$$
$$\mu = \sigma^{-2}\Sigma G^\top(y - F\vartheta_\star).$$

Since

$$\int \exp\left(-\frac{1}{2}\|\delta - \mu\|^2_{\Sigma^{-1}}\right) d\delta = \sqrt{(2\pi)^q|\Sigma|}$$

it follows that

$$p(y|\vartheta_\star;\sigma^2,P_\delta) = \frac{1}{\sqrt{(2\pi)^N(\sigma^2)^N|\Sigma|^{-1}|P_\delta|}} \times \exp\left(-\frac{1}{2}\|y - F\vartheta_\star\|^2_{C^{-1}}\right).$$

Since

$$\begin{aligned}|\Sigma|^{-1}|P_\delta| &= |\Sigma^{-1}P_\delta| = |I + \sigma^{-2}P_\delta G^\top G| \\ &= (\sigma^2)^{-N}|\sigma^2 I + GP_\delta G^\top| = (\sigma^2)^{-N}|C|\end{aligned}$$

it follows that

$$p(y|\vartheta_\star;\sigma^2,P_\delta) = \frac{1}{\sqrt{(2\pi)^N|C|}} \times \exp\left(-\frac{1}{2}\|y - F\vartheta_\star\|^2_{C^{-1}}\right).$$

Appendix B
Linearization of the Log-Determinant

Let $P_\delta = \mathrm{diag}(p)$ and denote the linearization point as $p = \tilde{p}$ and $\sigma^2 = \tilde{\sigma}^2$ for which $C = \tilde{C}$. Then the first-order Taylor expansion of the log-determinant can be written as

$$\begin{aligned}\ln|C| \simeq\ &\ln|\tilde{C}| + \partial_p \ln|C|_{|C=\tilde{C}}(p - \tilde{p}) \\ &+ \partial_{\sigma^2}\ln|C|_{|C=\tilde{C}}(\sigma^2 - \tilde{\sigma}^2).\end{aligned} \tag{36}$$

For the derivatives we have

$$\begin{aligned}\frac{\partial \ln|C|}{\partial p_i} &= \mathrm{tr}\left\{C^{-1}\frac{\partial C}{\partial p_i}\right\} \\ &= \mathrm{tr}\left\{C^{-1}g_i g_i^\top\right\},\end{aligned}$$

where g_i is the ith column of G. Similarly,

$$\begin{aligned}\frac{\partial \ln|C|}{\partial \sigma^2} &= \mathrm{tr}\left\{C^{-1}\frac{\partial C}{\partial \sigma^2}\right\} \\ &= \mathrm{tr}\left\{C^{-1}\right\}.\end{aligned}$$

When $\tilde{p}$ and $\tilde{\sigma}^2$ are chosen such that $\tilde{C} = cI_N$, then using the derivative expression above in (36) gives

$$\begin{aligned}\ln|C| &\simeq \sum_i \frac{1}{c}\mathrm{tr}\{g_i g_i^\top p_i\} + \frac{1}{c}\mathrm{tr}\{\sigma^2 I_N\} + K \\ &= \frac{1}{c}\mathrm{tr}\{GP_\delta G^\top + \sigma^2 I_N\} + K \\ &= \frac{1}{c}\mathrm{tr}\{C\} + K\end{aligned}$$

which equals the expression in (23). Here K is a constant.

References

[1] J. Sjöberg *et al.*, "Nonlinear black-box modeling in system identification: A unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.

[2] L. Ljung, *System Identification: Theory for the User*. Upper Saddle River, NJ, USA: Pearson Education, 1998.

[3] F. Giri and E.-W. Bai, *Block-Oriented Nonlinear System Identification*, vol. 1. New York, NY, USA: Springer-Verlag, 2010.

[4] T. Wigren, "Recursive prediction error identification using the nonlinear Wiener model," *Automatica*, vol. 29, no. 4, pp. 1011–1025, 1993.

[5] E.-W. Bai, "Non-parametric nonlinear system identification: An asymptotic minimum mean squared error estimator," *IEEE Trans. Autom. Control*, vol. 55, no. 7, pp. 1615–1626, Jul. 2010.

[6] J.-N. Lin and R. Unbehauen, "Canonical piecewise-linear approximations," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 39, no. 8, pp. 697–699, Aug. 1992.

[7] L. Breiman, "Hinging hyperplanes for regression, classification, and function approximation," *IEEE Trans. Inf. Theory*, vol. 39, no. 3, pp. 999–1013, May 1993.

[8] E. D. Sontag, "Interconnected automata and linear systems: A theoretical framework in discrete-time," in *Hybrid Systems III*. New York, NY, USA: Springer-Verlag, 1996, pp. 436–448.

[9] A. Bemporad, G. Ferrari-Trecate, and M. Morari, "Observability and controllability of piecewise affine and hybrid systems," *IEEE Trans. Autom. Control*, vol. 45, no. 10, pp. 1864–1876, Oct. 2000.

[10] P. Lerman, "Fitting segmented regression models by grid search," *Appl. Statist.*, vol. 29, pp. 77–84, 1980.

[11] F. Lauer, "On the complexity of piecewise affine system identification," *Automatica*, vol. 62, pp. 148–153, 2015.

[12] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, "Identification of hybrid systems a tutorial," *Eur. J. Control*, vol. 13, nos. 2/3, pp. 242–260, 2007.

[13] A. Garulli, S. Paoletti, and A. Vicino, "A survey on switched and piecewise affine system identification," in *Proc. 16th IFAC Symp. Syst. Identif.*, 2012, vol. 16, pp. 344–355.

[14] J. Roll, A. Bemporad, and L. Ljung, "Identification of piecewise affine systems via mixed-integer programming," *Automatica*, vol. 40, no. 1, pp. 37–50, 2004.

[15] R. Vidal, S. Soatto, Y. Ma, and S. Sastry, "An algebraic geometric approach to the identification of a class of linear hybrid systems," in *Proc. 42nd IEEE Conf. Decision Control*, 2003, vol. 1, pp. 167–172.

[16] A. Bemporad, A. Garulli, S. Paoletti, and A. Vicino, "A bounded-error approach to piecewise affine system identification," *IEEE Trans. Autom. Control*, vol. 50, no. 10, pp. 1567–1580, Oct. 2005.

[17] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari, "A clustering technique for the identification of piecewise affine systems," *Automatica*, vol. 39, no. 2, pp. 205–217, 2003.

[18] H. Nakada, K. Takaba, and T. Katayama, "Identification of piecewise affine systems based on statistical clustering technique," *Automatica*, vol. 41, no. 5, pp. 905–913, 2005.

[19] A. Juloski, S. Weiland, and W. Heemels, "A Bayesian approach to identification of hybrid systems," *IEEE Trans. Autom. Control*, vol. 50, no. 10, pp. 1520–1533, Oct. 2005.

[20] L. Bako, K. Boukharouba, E. Duviella, and S. Lecoeuche, "A recursive identification algorithm for switched linear/affine models," *Nonlinear Anal., Hybrid Syst.*, vol. 5, no. 2, pp. 242–253, 2011.

[21] H. Ohlsson and L. Ljung, "Identification of switched linear regression models using sum-of-norms regularization," *Automatica*, vol. 49, no. 4, pp. 1045–1050, 2013.

[22] T. Söderström and P. Stoica, *System Identification*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.

[23] L. Ljung and B. Wahlberg, "Asymptotic properties of the least-squares method for estimating transfer functions and disturbance spectra," *Adv. Appl. Probab.*, vol. 24, pp. 412–440, 1992.

[24] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[25] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[26] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy, "The effectiveness of lloyd-type methods for the k-means problem," in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci.*, 2006, pp. 165–176.

[27] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2007, pp. 1027–1035.

[28] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning* (Springer series in Statistics), vol. 1. Berlin, Germany: Springer-Verlag, 2001.

[29] G. Hamerly and C. Elkan, "Alternatives to the k-means algorithm that find better clusterings," in *Proc. 11th Int. Conf. Inf. Knowl. Manage.*, 2002, pp. 600–607.

[30] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B, Methodol.*, vol. 58, pp. 267–288, 1996.

[31] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[32] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: Where RLS meets the ℓ_1 -norm," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3436–3447, Jul.2010.

[33] P. Stoica and P. Babu, "The Gaussian data assumption leads to the largest Cramér-Rao bound," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 132–133, May 2011.

[34] S. Park, E. Serpedin, and K. Qaraqe, "Gaussian assumption: The least favorable but the most useful," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 183–186, Nov. 2013.

[35] H. Van Trees and K. Bell, *Detection Estimation and Modulation Theory, Part I. Detection Estimation and Linear Modulation Theory*, 2nd ed. Hoboken, NJ, USA: Wiley, 2013 [1968].

[36] C. Rao and M. Rao, *Matrix Algebra and Its Applications to Statistics and Econometrics*. Singapore: World Scientific, 1998.

[37] J. Berger, *Statistical Decision Theory and Bayesian Analysis* (Springer Series in Statistics). New York, NY, USA: Springer-Verlag, 1985.

[38] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50, no. 3, pp. 657–682, 2014.

[39] P. Stoica, P. Babu, and J. Li, "New method of sparse parameter estimation in separable models and its use for spectral analysis of irregularly sampled data," *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 35–47, Jan. 2011.

[40] P. Stoica, D. Zachariah, and J. Li, "Weighted SPICE: A unifying approach for hyperparameter-free sparse estimation," *Digit. Signal Process.*, vol. 33, pp. 1–12, 2014.

[41] D. Zachariah and P. Stoica, "Online hyperparameter-free sparse estimation method," *IEEE Trans. Signal Process.*, vol. 63, no. 13, pp. 3348–3359, Jul. 2015.

[42] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: With applications in machine learning and signal processing," *IEEE Signal Process. Mag.*, vol. 33, no. 1, pp. 57–77, Jan. 2016.

[43] W. I. Zangwill, *Nonlinear Programming: A Unified Approach*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1969.

[44] M. L. Stein, *Interpolation of Spatial Data: Some Theory for Kriging*. New York, NY, USA: Springer Science & Business Media, 1999.

[45] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control* (Lecture Notes in Control and Information Sciences), V. Blondel, S. Boyd, and H. Kimura, Eds. New York, NY, USA: Springer-Verlag, 2008, pp. 95–110. [Online]. Available: http://stanford.edu/ boyd/graph_dcp.html

[46] M. Grant and S. Boyd, CVX: MATLAB software for disciplined convex programming, version 2.1, Mar. 2014. [Online]. Available: http://cvxr.com/cvx

[47] P. Stoica and T. Söderstrom, "Instrumental-variable methods for identification of Hammerstein systems," *Int. J. Control*, vol. 35, no. 3, pp. 459–476, 1982.

[48] T. Wigren, "User choices and model validation in system identification using nonlinear Wiener models," in *Preprints 13th IFAC Symp. Syst. Identif.*, 2003, pp. 863–868.

[49] T. Wigren, "Recursive prediction error identification and scaling of nonlinear state space models using a restricted black box parameterization," *Automatica*, vol. 42, no. 1, pp. 159–168, 2006.

[50] A. L. Juloski, W. Heemels, and G. Ferrari-Trecate, "Data-based hybrid modelling of the component placement process in pick-and-place machines," *Control Eng. Pract.*, vol. 12, no. 10, pp. 1241–1252, 2004.

[51] A. Juloski, S. Paoletti, and J. Roll, "Recent techniques for the identification of piecewise affine and hybrid systems," in *Current Trends in Nonlinear Systems and Control*, L. Menini, L. Zaccarian, and C. Abdallah, Eds. Boston, MA, USA: Birkhäuser, 2006, pp. 79–99.

Per Mattsson received the M.Sc. degree in engineering physics from Uppsala University, Uppsala, Sweden, in 2010, where he is currently working toward the Ph.D. degree. His research interests include system identification, machine learning, and modeling of biomedical systems.

Dave Zachariah received the M.S. degree in electrical engineering and the Tech. Lic. and Ph.D. degrees in signal processing from the Royal Institute of Technology, Stockholm, Sweden, in 2007, 2011, and 2013, respectively. During 2007–2008, he was a Research Engineer at Global IP Solutions, Stockholm. He is currently a Researcher at Uppsala University, Uppsala, Sweden. His research interests include statistical signal processing, machine learning, sensor fusion, and localization.

Petre Stoica is a Researcher and an Educator in the field of signal processing and its applications to radar/sonar, communications and biomedicine. He is a Professor of signal and system modeling at Uppsala University, Sweden, and a Member of the Royal Swedish Academy of Engineering Sciences, the Romanian Academy (honorary), the European Academy of Sciences, and the Royal Society of Sciences in Uppsala.