Identification of Nonlinear Hybrid Systems in a Bayesian Framework

Ahmad Madary[a,b,*], Hamid Reza Momeni[a], Alessandro Abate[c,*], Kim G. Larsen[b], Adrien Le Coënt[b]

[a] *Department of Electrical and Computer Engineering, Tarbiat Modares University, Jalal AleAhmad, Nasr Bridge, Tehran*
[b] *Department of Computer Science, Aalborg University, Selma Lagerlöfs Vej 300, DK-9220 Aalborg East*
[c] *Department of Computer Science, Oxford University, Wolfson Building, Parks Road, Oxford OX1 3QD United Kingdom*

Abstract

This paper presents a Bayesian framework for the identification of nonlinear hybrid systems in the form of Switched Nonlinear AutoRegressive models with an eXogenous part (SNARX). The identification is done via three levels of inference, using Bayes' rule. At the first level, a hyper-parameter is assigned to each of the model parameters, which are then estimated by maximizing their posterior probabilities. The introduced hyper-parameters control the complexity of the model and leverage the Occam's Razor principle by selecting a model with sufficient complexity and proper accuracy. This is done in the second level of inference, where the optimum values of hyper parameters are calculated. At the third level of inference, a quality of measure is derived in order to contrast different results obtained from various identification procedures, comparing and selecting different model structures and their respective parameters. The proposed framework is compared with existing relevant methods and is tested on different numerical models, which has shown promising performance.

Keywords: Nonlinear hybrid systems, Switched nonlinear ARX models, Bayesian inference, System identification, Occam's Razor principle

*Corresponding Authors
Email addresses: amadary@cs.aau.dk (Ahmad Madary), aabate@cs.ox.ac.uk (Alessandro Abate)

1. Introduction

A Hybrid System (HS) is a dynamical system that consists of components with continuous and discrete behaviors [1]. In other words, a HS comprises more than one dynamical sub-system and the output at a specific time is determined by the governing sub-system at that time: in our setup, this is controlled by a discrete signal, also known as a switching signal. Hybrid systems have attracted considerable attention in the past few years since many current embedded systems are in essence hybrid. furthermore, HSs can be used to model complex nonlinear systems by means of a collection of simpler linear models [1].

In our framework, a HS in the form of a Switched Auto-Regressive Exogenous (SARX) system can be defined as

$$y_i = f_{\lambda_i}(\mathbf{x}_i) + e_i, \tag{1}$$

where $\mathbf{x}_i = [y_{i-1} \quad ... \quad y_{i-n_a} \quad u_{i-1-n_k} \quad ... \quad u_{i-n_b-n_k}]$ is the continuous state composed of n_b and n_a samples of lagged input u and output y respectively, n_k is the number of delayed samples, and e_i is the measurement noise. The exogenous, time-dependent variable $\lambda_i \in \{1, ..., n\}$ denotes the discrete mode and it determines which of the n sub-systems λ_i are active at that specific time (which means the corresponding dynamics are charecterised by the terms f_{λ_i}). If the functions f_{λ_i} are nonlinear, then the resulting system is a Switched Nonlinear ARX system (SNARX).

Considering the training data set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, the nonlinear sub-systems $\{f_j\}_{j=1}^{n}$ can be expressed as a summation of kernel functions in the following form [2]:

$$f_j(\mathbf{x}; \boldsymbol{\alpha}_j, b_j) = \sum_{i=1}^{N} \alpha_{ij} k_j(\mathbf{x}_i, \mathbf{x}) + b_j, \tag{2}$$

where the weights $\boldsymbol{\alpha}_j = [\alpha_{1j} \quad ... \quad \alpha_{Nj}]^T$ and the bias term b_j are the parameters of j^{th} sub-system and $k_j(\cdot)$ is a kernel function that satisfies Mercer's

condition [3] and represents the model structure $\mathcal{H}_j$. It should be emphasised that $\boldsymbol{\alpha}_j$ and b_j are the parameters for each sub-system f_j, while each model structure $\mathcal{H}_j$ has one or more hyper-parameters (e.g., the width of the Gaussian kernel).

The problem of identification of nonlinear HS is to fit the best parameters of the nonlinear sub-systems $\{f_j\}_{j=1}^n$ (weights $\boldsymbol{\alpha}_j$ and bias term b_j) and the time-dependent switching signal $\lambda_i \in \{1, ..., n\}$ (which of course selects the corresponding index j) to the training data set $\mathcal{D}$. This problem consists of two sub-problems that should be solved jointly: *the identification of the switching signal* and *the estimation of each sub-system*. If the switching signal is known a-priori, then the problem of identification of hybrid system reduces to a conventional identification of each sub-systems [4]; whereas if the dynamics of sub-systems are known, it becomes a classification problem [5].

Related literature. Various methods have been developed for identification of *linear* HSs. The major categories of methods are: clustering techniques [6, 7], Bayesian approaches [8, 9], mixed integer programming techniques [10, 11], bounded error approaches [12, 13], algebraic approaches [5, 14], and methods based on Support Vector Regression (SVR) [15]. Other methods, such as sum-of-norm optimization [16] and kernel methods using the hybrid stable spline algorithm [17] have been also developed to identify linear HSs. Moreover, the identifiability conditions that are specific to linear switched systems are discussed in [18, 19, 20]. For further information about these methods and other literature regarding the identification of linear hybrid systems, please see [21, 22, 23].

In the field of identification of *nonlinear* hybrid systems, much less research has been done. The authors in [24] extend the SVR method to the hybrid domain. In [2, 25] the authors use reduced-size kernels to decrease the dimension of the problem, so that it can be used for large data sets. In [26], sparse optimization techniques are used for identification. These techniques are extended into SVR and kernel expansion form in [27]. Authors in [28] propose a randomized

approach in order to identify nonlinear HSs, which reduces to a combinatorial optimisation problem. In [29] a Gaussian approach and stochastic simulations (Markov chain Monte Carlo) are used to identify a switched system consisting of one linear and one nonlinear sub-system.

Motivations. In SVR-based methods, the output is a point-wise prediction. Furthermore, the coefficient that determines the trade-off between the complexity of the model and data fitness should be determined by the user, which is a non-trivial task that is usually done by cross validation and search methods (e.g. random search or grid search). Moreover, the best structures for the family of models (i.e, best kernel functions, e.g. polynomial or Gaussian) and their respective parameters (such as the degree of polynomial and the width of a Gaussian) should be selected by the user. In order to choose the best kernels, the identification results for different kernels should be compared with each other. Moreover, the identification of nonlinear hybrid systems with SVR-based method will result in a non-convex optimization problem, which possesses many near optimal solutions [15]. Selecting the best output among the different results requires a comparison process which should encompass all the important factors affecting the quality of the identification. These factors are: fitness of data to the model, data assignment, and model complexity. Without a comprehensive quality measure, this comparison is done by selecting the best fitness (minimum error). However, using the fitness criterion alone is not sufficient as more complex models will just fit the data better. Besides, the quality of the identified switching signal and the amount of assigned data to each sub-system should be considered in selecting the set of the models. This requires a comprehensive quality measure that takes all the vital factors affecting the quality of the identification into account.

Contributions. In this paper, a three-level Bayesian framework[30] is introduced for identification of nonlinear hybrid systems. The model parameters are calculated in the first level, while the hyper-parameters controlling the complexity of the model and the estimated variance of the noise are calculated in the second

level, so that they provide a model with the best trade-off between complexity and data-fitness. In the third level of inference, a comprehensive quality measure is derived to assess the quality of the identification results, compare different kernel structures with various hyper-parameters, and also to compare the resulting estimated systems and to choose the best kernels and hyper-parameters. The derived quality measure takes data fitness, complexity of the model, and number of correct data assignments into consideration, and selects the simplest model with the best fitness and the most correct data assignment. Unlike SVR-based and randomized methods, the prediction provided by this method incorporates the uncertainty in the model parameters and also provides a probability distribution that can be used for sampling.

Contrasted with [24], where the noise can be estimated through a constrained optimization, in the presented method it is estimated by solving a set of equations using simple gradient-based methods. Furthermore, the best parameters for controlling the model complexity and data fitness are calculated in such a way that they satisfy the Occam's Razor factor, while in [24, 25, 2, 27], the trade-off parameter between complexity and data fitness should be determined using search methods, which is time consuming, and provides no guarantee that the chosen parameters produce the simplest model with the best data fit. In [24, 25, 2, 27], the comparison between different types of kernels or different hyper-parameters for kernels is possible only through data-fitness criteria, whereas our proposed method is able to comprehensively compare different kernels and parameters by considering their complexity, uncertainty, data fitness, and mode estimation.

Outline of the paper. The rest of the paper is organized as follows: in Section 2 the Bayesian set-up is introduced. The first, second, and third level of inference are introduce in Sections 3,4 and 5, respectively. Comparison with existing relevant methods, case studies and numerical simulations are presented in Section 6, while the results are discussed in Section 7.

2. Bayesian Set-Up

The identification problem for SNARX systems in a Bayesian framework consists in estimating several sets of parameters and hyper-parameters by maximizing their respective posterior probabilities. These posterior probabilities are calculated according to Bayes' rule in three levels of inference as shown in Figure 1. As it can be seen in Figure 1, the evidence of each level is the likelihood of the next level. We now introduce the parameters and the hyper-parameters that are required for this framework:

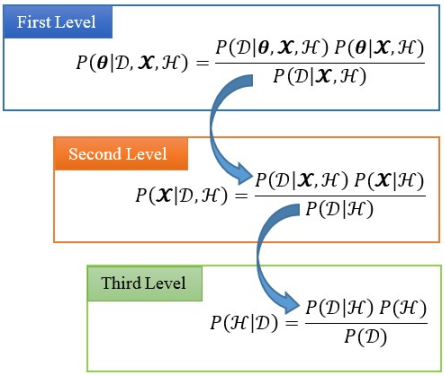


Figure 1: The three levels of inference in the Bayesian framework

- The total vector of the model parameters: $\boldsymbol{\theta} = [\boldsymbol{\alpha}, \boldsymbol{b}]^T$: where $\boldsymbol{\alpha}$ is the vector of the models' weights and $\boldsymbol{b}$ is the vector of bias terms: $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1 ... \boldsymbol{\alpha}_n]^T$, $\boldsymbol{b} = [b_1 ... b_n]^T$ (n is the number of sub-systems);
- The total vector of the model hyper-parameters: $\boldsymbol{\mathcal{X}} = [\boldsymbol{\mu}, \beta]$: This vector contains the variances for prior distribution of the weights and estimated noise variance;
- The family of kernels: $\mathcal{H} = \{\mathcal{H}_j | j = 1, ..., n\}$: is the family of the models with different structures and/or different values for parameters (e.g. the width of the Gaussian kernel or the degree or the polynomial kernel).

3. First level of inference: Model parameters

At the first level of inference, the vector of the model parameters $\boldsymbol{\theta} = [\boldsymbol{\alpha}, \boldsymbol{b}]^T$, which consists of weights and bias terms for each sub-system, is calculated through maximizing their posterior probabilities. The conditional posterior probability of the model parameters given the training data set consisting of N points $\mathcal{D} = \{x_i, y_i\}_{i=1}^N$, the vector of hyper-parameters $\boldsymbol{\chi}$ and the family of the kernels $\mathcal{H}$ is calculated according to the Bayes' rule

$$P\left(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\mathcal{X}}, \mathcal{H}\right) = \frac{P\left(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\mathcal{X}}, \mathcal{H}\right) P\left(\boldsymbol{\theta}|\boldsymbol{\mathcal{X}}, \mathcal{H}\right)}{P\left(\mathcal{D}|\boldsymbol{\mathcal{X}}, \mathcal{H}\right)}, \tag{3}$$

where $P\left(\boldsymbol{\theta}|\boldsymbol{\mathcal{X}}, \mathcal{H}\right)$ is the prior probability distribution of the model parameters.

The term $P\left(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\mathcal{X}}, \mathcal{H}\right)$ is the "*likelihood*" of the data points. The denominator of the equation (3) is called the hyper-parameter evidence and usually ignored in the calculation process of the model parameters since as it will be shown, it is not a function of model parameters [30].

3.1. Prior probability of the model parameters

In order to calculate the prior distribution of the model parameters, it is assumed that the parameters of each sub-system are independent from other sub-systems. Furthermore, the weights $\boldsymbol{\alpha}_j$ and bias term b_j are independent for each sub-system [30, 31, 32, 33]. Therefore, the conditional probability of the prior distribution over model parameters can be written as:

$$P\left(\boldsymbol{\alpha}, \boldsymbol{b}|\boldsymbol{\mathcal{X}}, \mathcal{H}\right) = \prod_{j=1}^{n} P\left(\boldsymbol{\alpha}_j|\boldsymbol{\mathcal{X}}, \mathcal{H}\right) P\left(b_j|\boldsymbol{\mathcal{X}}, \mathcal{H}\right). \tag{4}$$

It should be mentioned that while it is not possible to assume that the output of each sub-system at different data points are independent, due to the governing dynamical equations), the assumption of independent model parameters (weights and bias terms) can be sensibly made.

In the next step, it is assumed that the prior distribution of the weights $\boldsymbol{\alpha}_j$ of j^{th} sub-system has a normal distribution with zero mean and covariance

matrix equal to $\mu_j^{-1} I_N$:

$$P\left(\boldsymbol{\alpha}, \boldsymbol{b} | \boldsymbol{\mathcal{X}}, \mathcal{H}\right) = \prod_{j=1}^{n} P\left(\boldsymbol{\alpha}_j | \boldsymbol{\mathcal{X}}, \mathcal{H}\right) P\left(b_j | \boldsymbol{\mathcal{X}}, \mathcal{H}\right). \tag{5}$$

$$\begin{aligned} P\left(\boldsymbol{\alpha}_j | \boldsymbol{\mathcal{X}}, \mathcal{H}\right) &= \frac{1}{Z_{\boldsymbol{\alpha}_j}} e^{-\frac{\mu_j}{2} \boldsymbol{\alpha}^T \boldsymbol{\alpha}}; \\ Z_{\boldsymbol{\alpha}_j} &= \left(\frac{2\pi}{\mu_j}\right)^{\frac{N}{2}}. \end{aligned} \tag{6}$$

It is possible to use other types of prior, for example a Laplace distribution, but as it will be shown, the evidence cannot be computed in exact form and the obtained evidence is an approximation [30]. In (6), μ_j represents how sure we are about the weights a priori. This term will be discussed further in Section 4.

The second term in (5) is the prior probability distribution on the bias terms which is usually considered to be uninformative, due to the lack of prior information [30]. Under the assumption of non-informative prior distribution for the bias term b_j and of normal prior distribution (6) for weights $\boldsymbol{\alpha}_j$, the prior distribution of the model parameters can be written as follows:

$$P\left(\boldsymbol{\alpha}, \boldsymbol{b} | \boldsymbol{\mathcal{X}}, \mathcal{H}\right) = \frac{1}{\prod_{j=1}^{n} Z_{\boldsymbol{\alpha}_j}} e^{\left(\sum_{j=1}^{n} -\frac{\mu_j}{2} \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j\right)}. \tag{7}$$

3.2. Likelihood of the first level

The conditional distribution of $P\left(\mathcal{D} | \boldsymbol{\alpha}, \boldsymbol{b}, \boldsymbol{\mathcal{X}}, \mathcal{H}\right)$ is the likelihood term that can be seen as a model of the system noise that disturbs the measured training data. In order to write the complete likelihood, the data points should first be assigned to their respective sub-systems. For this purpose, the maximum likelihood principle is used [15]. The maximum likelihood mode estimation for hybrid systems tries to assign each data point $(\boldsymbol{x}_i, y_i)$ to a sub-system that most likely generates the data point, i.e. the one that maximizes the likelihood of the data with respect to the estimated sub-system $\hat{f}_j$. The maximum likelihood mode estimation can be expressed as:

$$\begin{aligned} \hat{\lambda}_i &= \underset{j=1,\ldots,n}{\arg\max} \; P(y_i | \boldsymbol{x}_i, \hat{f}_j), \\ P(y_i | \boldsymbol{x}_i, \hat{f}_j) &= \frac{e^{-\ell\left(y_i - \hat{f}_j(\boldsymbol{x}_i)\right)}}{Z_\delta}, \end{aligned} \tag{8}$$

where $\ell(.)$ is a proper loss function and Z_δ is a normalizing constant, while $\hat{f}_j$ is the estimated model of the j^{th} sub-system. Here we choose the likelihood function as a Gaussian distribution with the variance equal to $1/\beta$, which is our prior belief on the noise variance of the system. The term $y_i - \hat{f}_j(\boldsymbol{x}_i)$ in (8) is the prediction error and $P(y_i|\boldsymbol{x}_i, \hat{f}_j)$ is the probability density function of the prediction errors [34]. A typical assumption is that the prediction errors are independent (more information regarding this assumption can be found in Chapter 5 of [34]). Using this assumption, the complete likelihood of the data can be written as:

$$\begin{aligned} P\left(\mathcal{D}|\boldsymbol{\alpha}, \boldsymbol{b}, \boldsymbol{\mathcal{X}}, \mathcal{H}\right) &= \prod_{i=1}^{N} \underset{j=1,\ldots,n}{\arg\max}\ P\left(y_i - \hat{f}_j(\boldsymbol{x}_i)\right) \\ &= \prod_{i=1}^{N} \underset{j=1,\ldots,n}{\arg\max}\ \frac{e^{-\frac{\beta}{2}\left(y_i - \hat{f}_j(\boldsymbol{x}_i)\right)^2}}{Z_\delta} = \prod_{i=1}^{N} \frac{1}{Z_\delta} e^{\underset{j}{\arg\max} -\frac{\beta}{2}\left(y_i - \hat{f}_j(\boldsymbol{x}_i)\right)^2}, \\ Z_\delta &= (\frac{2\pi}{\beta}). \end{aligned} \tag{9}$$

3.3. Posterior probability distribution of model parameters

The posterior probability of the model parameters is calculated by combining the prior distribution of parameters (7) and the complete likelihood of the data (9) as:

$$\begin{aligned} P\left(\boldsymbol{\alpha}, \boldsymbol{b}|\mathcal{D}, \boldsymbol{\mathcal{X}}, \mathcal{H}\right) &= \frac{\prod_{i=1}^{N} Z_\delta^{-1} \prod_{j=1}^{n} Z_{\boldsymbol{\alpha}_j}^{-1} e^{-\mathcal{J}_1(\boldsymbol{\alpha}, \boldsymbol{b})}}{P(\mathcal{D}|\boldsymbol{\mathcal{X}}, \mathcal{H})}, \\ \mathcal{J}_1(\boldsymbol{\alpha}, \boldsymbol{b}) &= \sum_{j=1}^{n} \frac{\mu_j}{2} \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j + \frac{\beta}{2} \sum_{i=1}^{N} \underset{j=1,\ldots,n}{\arg\min}\ \left(y_i - \hat{f}_j(\boldsymbol{x}_i)\right)^2. \end{aligned} \tag{10}$$

In this expression, the normalizing term $P(\mathcal{D}|\boldsymbol{\mathcal{X}}, \mathcal{H})$ is the evidence of the hyper-parameters which will be used as the likelihood in the next level of inference. In order to obtain the parameters of the model, this posterior probability distribution should be maximized, which results in maximum a posteriori estimation of the parameters, denoted by $\boldsymbol{\alpha}^{MAP}$ and $\boldsymbol{b}^{MAP}$. Maximizing this term is equivalent to minimizing the negative logarithm of the posterior distribution, which

is expressed as

$$\boldsymbol{\alpha}^{MAP}, \boldsymbol{b}^{MAP} : \min_{\boldsymbol{\alpha},\boldsymbol{b}} \ \mathcal{J}_1 = \min_{\boldsymbol{\alpha},\boldsymbol{b}} \ \sum_{j=1}^{n} \frac{\mu_j}{2} \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j + \frac{\beta}{2} \sum_{i=1}^{N} \underset{j=1,\ldots,n}{\arg\min} \ \left(y_i - \hat{f}_j(\boldsymbol{\alpha}_j, \boldsymbol{b}_j, \boldsymbol{x}_i) \right)^2 . \tag{11}$$

Remark on the size of the data set. Since each data-point is associated with a weight for every sub-system, this optimization problem will become computationally expensive as the size of the data set increases; This problem is in fact common with every kernel based method, unless some appropriate measures such as dimension reduction techniques are being used to reduce the number of variables [25]. This issue will be addressed in our future research.

After calculating the optimal values for the sub-system parameters through (11), the estimated sub-systems $\hat{f}_j$ is calculated using (2). At this stage, since the estimated sub-systems are known, the discrete mode of each data point can be calculated by utilizing the maximum likelihood principle: the probability of each data point belonging to all the sub-system is calculated. The data point belongs to the sub-system with highest probability. Substituting the optimal values of the sub-system parameters obtained earlier in the maximum likelihood estimation (8) results in:

$$\hat{\lambda}_i = \underset{j=1,\ldots,n}{\arg\max} \ P(y_i | \boldsymbol{x}_i, \hat{f}_j(.; \boldsymbol{\alpha}^{MAP}, b^{MAP})), \qquad i = 1, ..., N. \tag{12}$$

This is a continuous-discrete optimization problem. In order to avoid the optimization problem on both continuous and discrete variables, [35] proposes to replace the min function on discrete variables with the *Product of Errors (PE)* estimator as a smooth approximation for the min function. Although the *PE estimation* can be used to approximate the min function, it is not the best smooth approximation. In this paper, we propose to use *Min LogSumExp (MinLSE) function* instead of the min. The logarithm of Summation of Exponential or *LSE* is a smooth approximation for maximum function [36]. The MinLSE function is defined based on this approximation, as follows.

Definition 1. *The MinLSE function for a set of* $\{x_j\}_{j=1}^n$ *is defined as*

$$MinLSE(x_1, ..., x_n) = -\kappa^{-1} \log \left(\sum_{j=1}^{n} \exp(-\kappa x_j) \right), \tag{13}$$

where $\kappa > 0$ *is a scale factor to further improve the accuracy of the approximation.*

The accuracy of a PE estimator depends on both the values and the numbers of its arguments. However, the maximum difference of MinLSE from the true minimum depends only on the number of the function arguments. The lower and upper bounds of the MinLSE are expressed in the following Theorem.

Lemma 1. *The MinLSE approximation of the* min *function for a set of* n *variables* $\{x_j\}_{j=1}^n$ *has the following lower and upper bounds:*

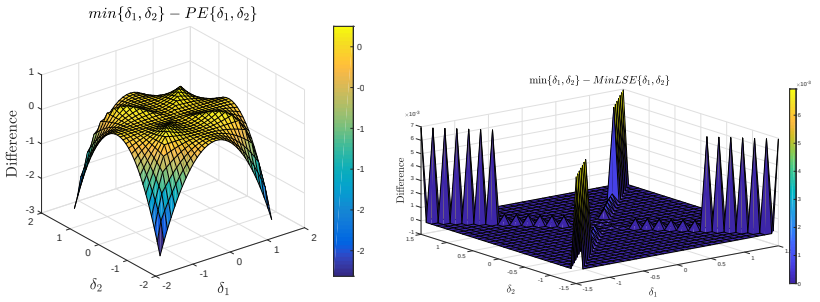
$$min\{x_1, ..., x_n\} - \kappa^{-1} log(n) \leq MinLSE(x_1, ..., x_n) < \min\{x_1, ..., x_n\}. \tag{14}$$

Proof. We can write $\min\limits_{j=1,...,n} \{x_j\} = -\kappa^{-1} \log \left(\exp \left(\max\limits_{j=1,...,n} \{-\kappa x_j\} \right) \right)$. Suppose that the $\max\limits_{j=1,...,n} \{-\kappa x_j\} = -\kappa x^*$. The logarithm on right hand side has the following upper bound:

$$\begin{aligned} \log \left(\exp \left(\max_{j=1,...,n} \{-\kappa x_j\} \right) \right) &< \log \left(\sum_{j=1}^{n} \exp(-\kappa x_j) \right) \\ &\leq \log \left(n \times \exp(-\kappa x^*) \right) = \log n - \kappa x^*. \end{aligned} \tag{15}$$

By multiplying (15) with $-\kappa^{-1}$, the lower and upper bounds in (14) will be obtained. □

It should be evident that with a proper κ, this lower bound can be made sufficiently small. The accuracy of PE estimation and MinLSE estimation versus min function is shown in Figure 2 for a two-argument case. It can be seen that the MinLSE function is very accurate compared to PE estimation and its performance slightly deteriorates only when the two arguments are very close to each other. Yet, its difference with the actual minimum is negligible.



(a) PE vs. min

(b) MinLSE vs. min

Figure 2: Accuracy of PE and MinLSE compared to actual min function.

Using the MinLSE function, the optimization problem (11) is re-written as follows. The posterior distribution of the model parameters can be summarized using the calculated values for $\boldsymbol{\alpha}^{MAP}, \boldsymbol{b}^{MAP}$ and the confidence interval on these maximum a-posteriori parameters. The confidence intervals are calculated from the curvature of the posterior distribution [30]. The posterior can be approximated locally with a Gaussian distribution as:

$$\begin{aligned} \boldsymbol{\alpha}^{MAP}, \boldsymbol{b}^{MAP} : \min_{\boldsymbol{\alpha},\boldsymbol{b}} \ \mathcal{J}_1 = \min_{\boldsymbol{\alpha},\boldsymbol{b}} \ & \sum_{j=1}^{n} \frac{\mu_j}{2} \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j \\ & + \frac{\beta}{2} \sum_{i=1}^{N} \underset{j=1,\ldots,n}{MinLSE} \left(\left(y_i - \hat{f}_j(\boldsymbol{x}_i) \right)^2 \right), \end{aligned} \tag{16}$$

$$P(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\mathcal{X}}, \mathcal{H}) \approxeq P(\boldsymbol{\theta}^{MAP}|\mathcal{D}, \boldsymbol{\mathcal{X}}, \mathcal{H}) \exp\left(-\frac{1}{2} \triangle \boldsymbol{\theta}^T \Sigma \triangle \boldsymbol{\theta} \right), \tag{17}$$

where $\boldsymbol{\theta}^{MAP} = [\boldsymbol{\alpha}^{MAP}, \boldsymbol{b}^{MAP}]^T$ and $\triangle\boldsymbol{\theta} = \boldsymbol{\theta} - \boldsymbol{\theta}^{MAP}$. In (17), Σ is the Hessian matrix, namely $\Sigma = - \bigtriangledown \bigtriangledown \log P(\boldsymbol{\alpha}, \boldsymbol{b}|\mathcal{D}, \boldsymbol{\mathcal{X}}, \mathcal{H})$, and the covariance (confidence interval) of $\mathcal{J}_1$ is equal to Σ^{-1}. The accuracy of this approximation depends on the problem. For the quadratic term that is used in this research, the approximation is exact [30].

After the most probable values of parameters have been obtained, the mode estimation will be done according to (12) and values of λ_i are calculated for each data point. The estimated modes can be encoded in a discrete variable

B_{ij} that is defined as

$$\begin{aligned} & B_{ij} \in \{0,1\}, \ \ \forall i = 1, ..., N \ \ j = 1, ..., n, \\ & s.t \ \ B_{ij} = 1 \ \text{ iff } \ \lambda_i = j \ \text{ and } \ \sum_{j=1}^{n} B_{ij} = 1, \end{aligned} \tag{18}$$

which encodes each data point to a sub-system.

Introducing this discrete variable into (11), the cost function $\mathcal{J}_1$ can be re-written as

$$\mathcal{J}_1 = \sum_{j=1}^{n} \frac{\mu_j}{2} \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j + \frac{\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{n} B_{ij} \left(y_i - \hat{f}_j(\boldsymbol{x}_i) \right)^2 . \tag{19}$$

The first term in this equation is called *regularization*, which expresses the kind of smoothness we expect from resulting model[30]. The second term is the data fitness.

4. Second level of inference: Hyper-parameters

The purpose of the second and third levels of inference is to obtain the optimal values for the model hyper-parameters, i.e. the variances of the weights for each model ($1/\mu_j$) and the a-priori noise variance ($1/\beta$). But, why is it necessary to obtain the optimal values of the hyper parameters? The answer to this question is that even for the case of dynamical (non-hybrid) systems, the model parameters depend heavily on the values of prior variances of the weights and noise, as they can cause severe under-fitting or over-fitting [30, 31, 37] (depending on the values of model parameters and the ratio β/μ_j). For hybrid systems, this is even more important, since the purpose is not only to fit models on the data, but also to estimate the the switching sequence. Improper values for μ_j,β and model parameters may result in the wrong mode estimation.

One can argue that only the ratio β/μ_j is important. This is true if the goal is only to obtain the best-fit parameters. But the advantage of separating these two parameters is that it provides the capability to incorporate the knowledge from other sources (for example the bound on the value of the noise). Also,

in order to construct the confidence intervals or to generate samples from the posterior distribution for use in Monte Carlo methods, this separation becomes important [31].

The second level of inference is dedicated to maximizing the posterior distribution of the hyper-parameters given the data points and the model using Bayes formula. This posterior probability distribution is expressed as

$$P(\boldsymbol{\mathcal{X}}|\mathcal{D},\mathcal{H}) = \frac{P(\mathcal{D}|\boldsymbol{\mathcal{X}},\mathcal{H})P(\boldsymbol{\mathcal{X}}|\mathcal{H})}{P(\mathcal{D}|\mathcal{H})}, \tag{20}$$

where $P(\boldsymbol{\mathcal{X}}|\mathcal{H})$ is the prior distribution given the model set $\mathcal{H}$. Since before the training little information is known about the optimum values of the hyper-parameters, their prior distribution is assumed to be flat [31] (flat over logarithmic scale, since they are scale parameters). This assumption implies that none of the values for the hyper-parameters have any advantages against others and all of them are equally probable. For more information about priors, one can refer to [30]. Also, $P(\mathcal{D}|\mathcal{H})$ is the evidence of the model, which will be used in the third level of inference.

4.1. Likelihood of the second level of inference

The term $P\left(\mathcal{D}|\boldsymbol{\mathcal{X}},\mathcal{H}\right)$ is the likelihood of the training data given the model hyper-parameters and model family $\mathcal{H}$, which according to (3) is the evidence of the first level of inference. Using the assumption of uniform prior for hyper-parameters, maximizing the posterior distribution is equivalent to maximizing the likelihood of the second level. Let $\boldsymbol{\theta} = [\boldsymbol{\alpha}\ \ \boldsymbol{b}]^T$ represent the parameters of the model. The evidence of the first level is calculated by marginalizing over model parameters using the following integral [30].

$$P(\mathcal{D}|\boldsymbol{\mathcal{X}},\mathcal{H}) = \int P(\mathcal{D}|\boldsymbol{\theta},\boldsymbol{\mathcal{X}},\mathcal{H})P(\boldsymbol{\theta}|\boldsymbol{\mathcal{X}},\mathcal{H})d\boldsymbol{\theta}. \tag{21}$$

It is common that this posterior has a peak around the most probable values for model parameters, so the evidence integral can be approximated with the integrand's peak and its width $\triangle\boldsymbol{\theta}$ [31]. The best fit likelihood is multiplied by

the Occam's factor, which is less than one and penalises model $\mathcal{H}$ for having parameter $\boldsymbol{\theta}$:

$$\underbrace{P(\mathcal{D}|\boldsymbol{\mathcal{X}},\mathcal{H})}_{\text{Evidence}} \approx \underbrace{P(\mathcal{D}|\boldsymbol{\theta}^{MAP},\boldsymbol{\mathcal{X}},\mathcal{H})}_{\text{Best Fit Likelihood}} \quad \underbrace{P(\boldsymbol{\theta}^{MAP}|\boldsymbol{\mathcal{X}},\mathcal{H}) \triangle \boldsymbol{\theta}}_{\text{Occam's Factor}}. \tag{22}$$

The Occam's Razor principle states that "a model should be sufficiently complex to fit the data" or, in other words, a model should not be overly complex. Complex models which possess a lot of parameters that can take values in a broad interval will typically penalized with a large Occam factor, compared to simple models [30]. The Occam factor rewards simpler models. This factor also penalizes models that need to be tuned finely in order to fit the data [30]. In other words, it encourages models that require rough precision on their parameters [30]. The integral (21) can be approximated locally as a Gaussian distribution with covariance matrix Σ, as follows:

$$P(\mathcal{D}|\boldsymbol{\mathcal{X}},\mathcal{H}) = \prod_{j=1}^{n} Z_{\boldsymbol{\alpha}_j}^{-1} \prod_{i=1}^{N} Z_{\delta}^{-1} e^{-\mathcal{J}_1(\boldsymbol{\theta}^{MAP})} (2\pi)^{\frac{n(N+1)}{2}} |\Sigma|^{-\frac{1}{2}}, \tag{23}$$

where $Z_{\boldsymbol{\alpha}_j} = \left(\frac{2\pi}{\mu_j}\right)^{\frac{N}{2}}$, $Z_{\delta} = \left(\frac{2\pi}{\beta}\right)^{\frac{1}{2}}$ and Σ is the Hessian matrix of the first-level cost function.

Remark on the choice of the prior distributions. In General, any prior distribution can be assumed for the parameters (e.g., we have assumed flat prior or uninformative for the bias term). However, since obtaining the posterior distribution requires integration from a term with include the prior, and these integrals might be difficult to calculate, assuming Gaussian priors allows to use the Gaussian approximation of the integral [30, 31].

The hyper-parameters μ_j have the duty to control the complexity of the model. A model with large values for μ_j (low variance on prior distribution of weights) fits data from a smooth function, while a model with small μ_j (large freedom on the prior range of possible α) fits the data from both complex and smooth function. According to the Occam's Razor principle, this parameter should not be too high or too low [38]. One of the most interesting aspects

of the Bayesian approach is that the Occam's Razor principle will be applied automatically by integrating out all the irrelevant variables. In other words, the Bayesian framework automatically prefers simple models that sufficiently explain the data without unnecessary complexity [38] and this property holds even if the prior probability is completely uninformative [30].

In order to obtain the most probable values of the hyper-parameters, the posterior probability (23) should be maximized; or equivalently, its negative logarithm should be minimized. Therefore, the cost function of the second level of inference can be calculated as:

$$\mathcal{J}_2 = -\frac{N}{2}\left(\sum_{j=1}^{n}\log\mu_j + \log\beta\right) + \frac{N-n}{2}\log 2\pi + \mathcal{J}_1(\boldsymbol{\theta}^{MAP};\boldsymbol{\mu},\beta) + \frac{1}{2}\log|\Sigma|. \tag{24}$$

The Hessian matrix Σ can be written as:

$$\Sigma = \begin{pmatrix} M_\mu + \beta H_1 & \beta H_2 \\ \beta H_2^T & \beta H_3 \end{pmatrix}, \tag{25}$$

where

$$[H_{1j}]_{ts} = \left[\frac{\partial^2\mathcal{J}_1}{\partial\alpha_{tj}\partial\alpha_{sj}}\right] = \begin{cases} if\ \ t=s: & \mu_z + \beta\sum_{i=1}^{N} B_{iz}k_z(\boldsymbol{x}_i,\boldsymbol{x}_t)^2 \\ if\ \ t\neq s: & \beta\sum_{i=1}^{N} B_{iz}k_z(\boldsymbol{x}_i,\boldsymbol{x}_t)k_z(\boldsymbol{x}_i,\boldsymbol{x}_s) \end{cases}$$

$$[H_{2j}]_{s1} = [\frac{\partial^2\mathcal{J}_1}{\partial b_j\partial\alpha_{sj}}] = \beta\sum_{i=1}^{N} B_{iz}k_z(\boldsymbol{x}_i,\boldsymbol{x}_s)$$

$$H_{3j} = \frac{\partial^2\mathcal{J}_1}{\partial b_j\partial b_j} = \beta\sum_{i=1}^{N} B_{iz},$$

and where M_μ is a diagonal $(nN\times nN)$ matrix. One of the properties of this Hessian matrix is that, due to the devised formulation, it is sparse and its elements are block-diagonal matrices: $M_\mu = \text{diag}(\mu_1 I_N,...,\mu_n I_N)$, $H_1 = \text{diag}\,(H_{11},...,H_{1n}) \in \mathbb{R}^{nN\times nN}$ and $H_{1j}\in\mathbb{R}^{N\times N}$, $H_2 = \text{diag}\,(H_{21},...,H_{2n}) \in \mathbb{R}^{nN\times n}$ and $H_{2j}\in\mathbb{R}^{N\times 1}$ and $H_3 = \text{diag}\,(H_{31},...,H_{3n})\in\mathbb{R}^{n\times n}$ and $H_{3j}\in\mathbb{R}$ for $j=1,...,n$.

The determinant of the Hessian matrix can be calculated as: $|\Sigma| = |M_\mu + \beta H_a||\beta H_3|$, where $H_a = (H_1 - H_2 H_3^{-1}H_2^T)$. Because of the block-diagonal

nature of the components of Σ, H_a is also a block-diagonal matrix: $H_a = \text{diag}\,(H_{a1}, ..., H_{an})$, which can be expressed as a function of the components of Σ as: $H_{aj} = H_{1j} - H_{2j} H_{3j}^{-1} H_{2j}^T$. Using these notations, the determinant of Σ can be expressed as

$$|\Sigma| = \prod_{j=1}^{n} |\mu_j I_N + \beta H_{aj}| \;\; |\beta H_3|. \tag{26}$$

The logarithm of $|\Sigma|$ can be written in term of the non-zero eigenvalues of H_{aj} as shown below

$$\begin{aligned} \log|\Sigma| = \sum_{j=1}^{n} \left((N - k_j) \log \mu_j + \sum_{l=1}^{k_j} \log\left(\mu_j + \beta \lambda_l(H_{aj})\right) \right) \\ + n \log \beta + \sum_{t=1}^{n} \log \lambda_t(H_3), \end{aligned} \tag{27}$$

where k_j is the number of non-zero eigenvalues of H_{aj}, which is only a function of the kernel and of the training data points.

4.2. Optimal values of the hyper-parameters

In order to calculate the most probable values for the hyper-parameters μ_j^{MAP} and β^{MAP}, the posterior distribution (23) should be maximized; or equivalently the cost function of the second level of inference $\mathcal{J}_2$ should be minimized. This can be done by differentiating $\mathcal{J}_2$ with respect to the mentioned hyper-parameters and solving the resulting equations. The equations for obtaining these hyper-parameters are derived as follows.

Variance of the weights μ_j. The derivative of $\mathcal{J}_2$ (equation (24)) with respect to μ_j is:

$$\frac{\partial \mathcal{J}_2}{\partial \mu_j} = -\frac{N}{2\mu_j} + \frac{1}{2} \frac{\partial \log|\Sigma|}{\partial \mu_j} + \frac{\partial \mathcal{J}_1(\boldsymbol{\theta}^{MAP}; \boldsymbol{\mu}, \beta)}{\partial \mu_j}, \tag{28}$$

where in this equation we have:

$$\begin{aligned} \frac{\partial \mathcal{J}_1(\boldsymbol{\theta}^{MAP}; \boldsymbol{\mu}, \beta)}{\partial \mu_j} &= \frac{1}{2} \parallel \boldsymbol{\alpha}_j^{MAP} \parallel_2^2, \\ \frac{\partial \log|\Sigma|}{\partial \mu_j} &= \frac{(N - k_j)}{\mu_j} + \sum_{l=1}^{k_j} \frac{1}{\mu_j + \beta \lambda_l(H_{aj})}. \end{aligned} \tag{29}$$

Combining these two equations, the derivative of $\mathcal{J}_2$ with respect to μ_j can be expressed as follows:

$$\frac{\partial \mathcal{J}_2}{\partial \mu_j} = -\frac{k_j}{2\mu_j} + \frac{1}{2}\sum_{l=1}^{k_j} \frac{1}{\mu_j + \beta\lambda_l(H_{aj})} + \frac{1}{2} \parallel \boldsymbol{\alpha}_j^{MAP} \parallel_2^2 . \tag{30}$$

Variance of the noise β. The derivative of $\mathcal{J}_2$ with respect to β can be obtained the same way. It has the following general form

$$\frac{\partial \mathcal{J}_2}{\partial \beta} = \frac{1}{2}\frac{\partial \log|\Sigma|}{\partial \beta} + \frac{\partial \mathcal{J}_1(\boldsymbol{\theta}^{MAP}; \boldsymbol{\mu}, \beta)}{\partial \beta} - \frac{N}{2\beta}. \tag{31}$$

The two partial derivatives in this equation are:

$$\begin{aligned} \frac{\partial \mathcal{J}_1(\boldsymbol{\theta}^{MAP}; \boldsymbol{\mu}, \beta)}{\partial \beta} &= \sum_{i=1}^{N}\sum_{j=1}^{n} \frac{B_{ij}}{2}\left(y_i - f_j(\boldsymbol{x}_i)\right)^2 , \\ \frac{\partial \log|\Sigma|}{\partial \beta} &= \sum_{j=1}^{n}\sum_{l=1}^{k_j} \frac{\lambda_l(H_{aj})}{\mu_j + \beta\lambda_l(H_{aj})} + \frac{n}{\beta}. \end{aligned} \tag{32}$$

Combining these equations with (30), the derivative of $\mathcal{J}_2$ with respect to β can be expressed as:

$$\frac{\partial \mathcal{J}_2}{\partial \beta} = \frac{1}{2}\sum_{j=1}^{n}\sum_{l=1}^{k_j} \frac{\lambda_l(H_{aj})}{\mu_j + \beta\lambda_l(H_{aj})} + \frac{n-N}{2\beta} + \sum_{i=1}^{N}\sum_{j=1}^{n} \frac{B_{ij}}{2}\left(y_i - f_j(\boldsymbol{x}_i)\right)^2 . \tag{33}$$

As also mentioned in [30, 39], $\gamma_j = 1 + \sum_{l=1}^{k_j} \frac{\beta\lambda_l(H_{aj})}{\mu_j + \beta\lambda_l(H_{aj})}$ is the "number of good parameter measurements" for the j^{th} sub-system. Each eigenvalue $\beta\lambda_l(H_{aj})$ determines that how strongly the corresponding parameter has been determined by data, while μ_j measures the effect of the prior on the parameters [30].

It is worth mentioning that we will not perform simultaneous optimization over weights (from Section 3) and hyper parameters μ_j, β. The reason behind this is that both the posterior and likelihood might have skew distributions, so that the maximum likelihood value for the parameters and for the majority of the posterior probabilities might be separated [31].

Remark. Obtaining different values for maximum likelihood and maximum a-posteriori estimation is similar to finding the parameter of a Gaussian distribution (m, σ) from N data points. The maximum likelihood estimation and

the most probable values (obtained by integration over m, i.e. marginalization) for σ are $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2}$ and $\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2}$, respectively. In fact, it is this marginalization that corrects the bias of maximum likelihood estimation [30].

5. Third level of inference: Hyper-parameters

The third level of inference is dedicated to obtaining a *Quality Measure (QM)* for the identification process in order to assess the performance of the identification. There are several parameters that affect the identification of hybrid systems and that contribute to the quality of the identified model. The key components are:

1. Data fitness;
2. Complexity of the model;
3. Number of data assignment to each sub-system.

For conventional, non-hybrid systems, Least Square Support Vector Machines (LS-SVM) [39] incorporates the first two items to rate the identification process and compare different models. But the objective of the identification of hybrid systems is to estimate the parameters of each sub-system and the switching signal simultaneously. Current methods for identification of nonlinear hybrid systems can control the complexity of the model through the trade-off coefficient in the SVR methods, but are not capable of directly incorporating the complexity of the model with data fitness to compare the different identified models or model structures. To make the matter more complicated, the amount of assigned data to each sub-system should also be considered in order to compare the results of different identification procedures. To our knowledge, there is not a unified comprehensive criterion for SVR methods that can include all these items together for hybrid systems.

Another need for having a unified quality measure to compare the results of identification is that the current identification problem for non-linear hybrid systems (including the present research and [35]) is a non-convex optimization

problem which possesses multiple near-optimal solutions. So, not only the choice of various model structures or even different model parameters for a particular sub-system will affect the overall identification process, but also different repetitions for fixed models structures might also result in different identification outcomes. Therefore, it is essential to have a comprehensive criterion to assess the quality of the solutions.

The purpose of the third level of inference is to provide a comprehensive measure to assess the quality of identification of hybrid systems. This measure fulfills the following goals:

- Comparing and selecting different solutions for the identification problem;
- Comparing and selecting different model structures;
- Comparing different model parameters.

It should be noted that comparing different models is a difficult subject, since selecting a model by simply choosing the one with the best data fitness based on criteria such as Mean Square Error (MSE) causes over-fitting, as more complex models always fit better the data. Therefore, choosing the best model by only considering the fitness (for example the maximum likelihood) will result in over-parameterized models with poor generalization. This is where Occam's razor should be used [30]. The third level of inference also provides a tool to assess the effect of choosing a particular model for one sub-system on the overall identification process.

The posterior distribution of model $\mathcal{H}_j$ will be used as the quality measure for that particular model. Assuming a flat prior for model $\mathcal{H}_j$, the posterior distribution will be proportional to the likelihood $P(\mathcal{D}|\mathcal{H}_j)$, which is the evidence of the model in the previous level. This posterior distribution has the following form

$$P(\mathcal{H}_j|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{H}_j)P(\mathcal{H}_j)}{P(\mathcal{D})}. \tag{34}$$

The evidence $P(\mathcal{D}|\mathcal{H}_j)$ will be obtained by integrating out all the variables

(hyper-parameters $\boldsymbol{\mathcal{X}}_j = [\mu_j \;\; \beta]$):

$$P(\mathcal{D}|\mathcal{H}_j) = \int P(\mathcal{D}|\boldsymbol{\mathcal{X}}_j, \mathcal{H}_j) P(\boldsymbol{\mathcal{X}}_j|\mathcal{H}_j) d\boldsymbol{\mathcal{X}}_j. \tag{35}$$

The evidence can be approximated accurately by a separable normal distribution with confidence intervals σ_{μ_j} and σ_β . These confidence intervals are calculated by differentiating (24) twice with respect to μ_j and β:

$$\begin{aligned} (\sigma_{\mu_j})^2 &= \left(\frac{k_j}{2\mu_j^2} - \frac{1}{2} \sum_{l=1}^{k_j} \frac{1}{(\mu_j + \beta\lambda_l(H_{aj}))^2} \right)^{-1}, \\ (\sigma_\beta)^2 &= \left(\frac{N-n}{2\beta^2} - \frac{1}{2} \sum_{j=1}^{n} \sum_{l=1}^{k_j} \frac{\lambda_l(H_{aj})^2}{(\mu_j + \beta\lambda_l(H_{aj}))^2} \right)^{-1}. \end{aligned} \tag{36}$$

Using these confidence intervals in (35) and assuming a flat prior, the evidence will be calculated as

$$P(\mathcal{D}|\mathcal{H}_j) \approxeq P(\mathcal{D}|\boldsymbol{\mathcal{X}}_j^{MAP}, \mathcal{H}_j) 2\pi\sigma_\beta\sigma_{\mu_j}, \tag{37}$$

where $P(\mathcal{D}|\boldsymbol{\mathcal{X}}^{MAP}, \mathcal{H}_j)$ is calculated by using the most probable values for hyper-parameter which are obtained in the previous level in (23).

Neglecting all the constants in (23) and (37), the posterior distribution for model $\mathcal{H}_j$ is calculated as:

$$P(\mathcal{D}|\mathcal{H}_j) \propto \sqrt{\frac{(\mu_j^{MAP})^N \sigma_{\mu_j}^2 (\beta^{MAP})^{\sum_{i=1}^N B_{ij}} \sigma_\beta^2 exp(-\mathcal{J}_1\left(\boldsymbol{\theta}^{MAP}, \boldsymbol{\mathcal{X}}^{MAP}\right))}{(\mu_j^{MAP})^{(N-k_j)} \prod_{l=1}^{k_j} (\mu_j^{MAP} + \beta^{MAP}\lambda_l(H_{aj})) \beta^{MAP} H_{3j}}}. \tag{38}$$

It is more convenient to use the logarithm of (38) as a measure of quality of model. The *Quality Measure* for model $\mathcal{H}_j$ is expressed as:

$$\begin{aligned} QM(\mathcal{H}_j) = &\log\sigma_{\mu_j} + \log\sigma_\beta + \frac{k_j}{2}\log\mu_j^{MAP} + \frac{\zeta_j - 1}{2}\log\beta^{MAP} - \frac{1}{2}\log\zeta_j \\ &- \frac{1}{2}\sum_{l=1}^{k_j}\left(\mu_j^{MAP} + \beta^{MAP}\lambda_l(H_{aj})\right) - \frac{\mu_j^{MAP}}{4}||\alpha_j||_2^2 \\ &- \frac{\beta^{MAP}}{4}\sum_i^N B_{ij}\left(y_i - f_j(\boldsymbol{x}_i)\right)^2. \end{aligned} \tag{39}$$

In this expression, $\zeta_j = \sum_{i=1}^{N} B_{ij}$ is the number of data points assigned to the model $\mathcal{H}_j$, and k_j is the number of non-zero eigenvalues of the kernel matrix corresponding to $\mathcal{H}_j$.

This quality measure has a unique characteristic: it includes all the relevant components that determine the quality of identification. These components are:

- Model fitness corresponding to $\mathcal{H}_j$: $\sum_i^N B_{ij} \left(y_i - f_j(\boldsymbol{x}_i)\right)^2$ and prior variance of noise $1/\beta^{MAP}$;
- Model Complexity: regularization term $||\alpha_j||_2^2$ and prior variance of the weights μ_j^{MP};
- Uncertainty about the noise and weight variances: $\log \sigma_{\mu_j}, \log \sigma_\beta$;
- Number of the data points assigned to model $\mathcal{H}_j$: ζ_j;
- Characteristics of kernel matrix (eigenvalues) corresponding to $\mathcal{H}_j$.

Summarizing: *this quality measure rewards simple models with the best data fitness and the most assigned data points.* Since every change in one $\mathcal{H}_j$ will alter the identification results for other models, an overall quality measure for identification should be defined to incorporate all the changes in the overall model family $\mathcal{H}$. This quality measure is defined as the summation of QM for all the models:

$$QM_{Overall} = \sum_{j=1}^{n} QM(\mathcal{H}_j). \tag{40}$$

Relation with Minimum Description Length and Akaike criterion. The Minimum Description Length (MDL) tries to select a model that best compresses the data (model with fewer parameters). The MDL can be written in crude from as $L(\mathcal{H}) + L(\mathcal{D}|\mathcal{H})$, where $L(\mathcal{H})$ is the length describing the model $\mathcal{H}$ in bit and $L(\mathcal{D}|\mathcal{H})$ is the length describing the data $\mathcal{D}$ encodes by $\mathcal{H}$ (which can be seen as $-\log P(\mathcal{D}|\mathcal{H})$) [40]. The QM is obtained from the logarithm of (37), which is very similar to the MDL for conventional non-hybrid systems and the Akaike criterion (AIC), which can be seen as the approximation of MDL [30].

6. Case Studies

In this section, several case studies are presented to test the performance of the proposed method. Due to the limited research in the field of identification of nonlinear hybrid systems, and the fact that the models used in this literature are deemed to be fairly simple, the models used for the case studies are devised anew.

First, the performance of the MinLSE approximation introduced in the first level of inference of the proposed method is compared with the PE framework introduced in [15] and used in [24, 25, 2, 35]. The performance comparison is based on the MSE and the percentage of correct data-assignment for identification of a SNARX system.It should be noted that for this comparison, only the first level of inference from the proposed method is used. As for the PE framework, although [24, 25, 2, 35] use dimension reduction and support vector (SV) selection, here for consistency we only implement the optimization formulation using PE, so no SV selection or dimension reduction is done.

Two kinds of SNARX systems with different switching characteristics are identified: an exogenous (or predefined) switch in time, and a state-dependent switch. (Later, the performance of the third level of inference is verified in order to compare different models and to judge the performance of the identification procedure.)

6.1. Performance Comparison

In this part, the first level of inference of the proposed method is compared with the PE solver in [15] for identification of the following SNARX model:

$$y_i = \begin{cases} -0.4y_{i-1}^2 + 0.5u_{i-1} + e_i & if \;\; \lambda_i = 1 \\ (0.8 - 0.5e^{y_{i-1}^2})y_{i-1} - y_{i-1}^2 + 0.9u_{i-1} + e_i & if \;\; \lambda_i = 2. \end{cases} \tag{41}$$

In [15], the identification is done using the following optimization so-called Product of Error (PE):

$$\min_{\{\boldsymbol{\alpha}_j\},\{b_j\}} \frac{1}{n}\sum_{j=1}^{n}\mathcal{R}(\boldsymbol{\alpha}_j) + \frac{C}{N}\sum_{i=1}^{N}\prod_{j=1}^{n}\ell\left(y_i - \sum_{k=1}^{N}\alpha_{kj}k_j(\boldsymbol{x}_k,\boldsymbol{x}_i) - b_j\right), \tag{42}$$

where $\mathcal{R}(\cdot)$ is the regularization term, $\ell(.)$ is the loss function, C is the trade-off coefficient between model complexity and data fitness, and $k_j(\cdot,\cdot)$ is the kernel function of the j^{th} sub-system. In order to make the comparison, the L_2-norm is used as regularizer ($\mathcal{R}(\boldsymbol{\alpha}_j) = \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j$) and a quadratic loss function is employed. It should be mentioned that since the objective of this part is to compare the performance of the MinLSE with PE, only the first level of inference from the proposed method is used (wihout optimizing the hyper-parameters in the second level of inference). Furthermore, all of the other hyper-parameters (e.g. regularization term) of the solver for the first level of inference and PE framework, along with the kernel type and hyper-parameters, are chosen randomly and are equal for both solvers.

The output of system (41) is measured for $N = 100$ data points generated with a random uniform input u_i in the range of [0 1] starting from a random initial point y_0. The system mode switches from $\lambda = 1$ to $\lambda = 2$ at $i = 41$. The outputs are perturbed with a measurement noise e_i, which is considered to be Gaussian with variance equal to 0.01. Two Gaussian kernels $\mathcal{H}_i(\sigma) = \exp(-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{\sigma})$, with equal parameter (width σ) set to 1 are used for both methods. Towards fairness, hyper-parameter values are set to $\mu_j = \beta = 2$, while $C = 50$ and identification is repeated 200 times. The obtained results for percentage of data-assignment and MSE are shown in the following figures.

As it can be seen from the Figure 3 and Figures 4, the results of the first level of inference in the proposed method is better than the PE method from [2] in terms of MSE and of percentage of correct data assignments. As mentioned before, determining the trade-off coefficient C in the PE framework is not a trivial task and is usually done through cross validation and search methods, whereas the optimal values of hyper-parameters in the proposed method are obtained in the second level of inference, in such a way that the simplest model with the best data-fitness is obtained.

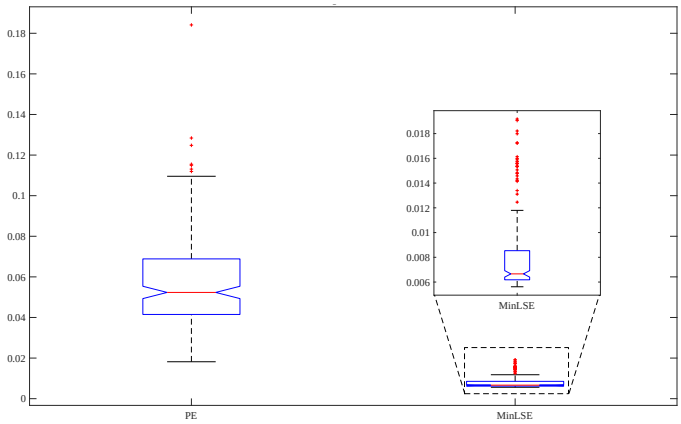


Figure 3: Comparison of MSE results between proposed method (with MinLSE) and PE method.

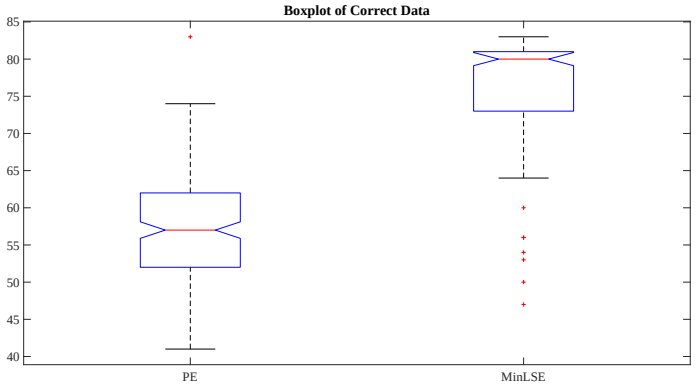


Figure 4: Comparison of percentage of correct data-assignment results between proposed method (with MinLSE) and PE method.

6.2. Identification of switched NARX system

In this part, the performance of the complete method (with all the three levels of inference) is tested on two different NARX systems (in the previous case study in Section 6.1 only the first level of inference was used).

6.2.1. NARX systems with exogenous switching

First, the system in (41) is identified with the set of models $\mathcal{H}$ is chosen as two Gaussian kernels with parameters (width σ) equal to 0.05 and 1: $\mathcal{H} =$

$\{\mathcal{H}_1(0.05), \mathcal{H}_2(1)\}$. After the identification is completed, the optimized hyper-parameters are estimated as $\mu_1 = 108.6957, \mu_2 = 119.0476$ and $\beta = 166.67$, which represent the estimated variances of the weights ($\sigma^2_{\alpha_1} = 0.0092, \sigma^2_{\alpha_2} = 0.0084$) and the estimated variance of the noise ($\sigma^2_e = 0.006$). The identification results are illustrated in Figure 5: 84% of the data points have been assigned correctly and the MSE is only 0.0041.

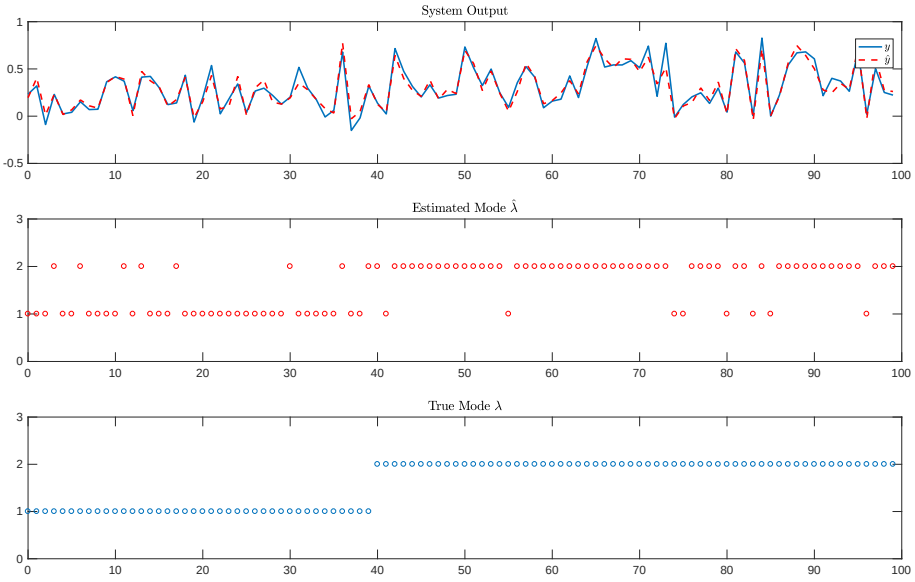


Figure 5: Identification results for exogenous switching NARX model

NARX systems with state-dependent switching. The following system is put under the identification procedure:

$$y_i = \begin{cases} k\sqrt{(\frac{y_{i-1}}{k})^2 + \frac{u_{i-1} - y_{i-1}}{s}} + e_i & y_{i-1} \leq Y \\ 1.3 + \sqrt{u_{i-1}} + \exp(-y_{i-1}) + e_i & y_{i-1} > Y, \end{cases} \tag{43}$$

where $s = 10, k = 0.6$ and threshold $Y = 1.3$. The system switches between two modes according to the value of its output. It is started from a random initial condition y_0 and $N = 100$ data points are generated with a random input uniformly distributed in the range $u_i \in [0\ \ 4]$ and a Gaussian noise with zero mean and standard deviation equals to 0.1. Again, two Gaussian kernels

with parameters set to 0.5 and 1 are chosen: $\mathcal{H} = \{\mathcal{H}_1(0.5), \mathcal{H}_2(1)\}$. The identification is initialized with $\boldsymbol{\mu} = [1\ \ 1]$ and $\beta = 1$, then repeated with the optimized values for hyper-parameters obtained at the second level of inference: $\boldsymbol{\mu} = [10.98\ \ 11.07]$ and $\beta = 33.94$ which corresponds to 0.1716 for the estimated standard deviation of the noise. The MSE reduces from 0.0202 in initial run to 0.0093 after optimizing the hyper-parameters.

The estimated modes and output are shown in Figure 6 and Figure 7. It can be seen from the figures that the overall accuracy of the identification is improved after optimizing the hyper-parameters. Furthermore, the standard deviation of the noise is estimated with relatively good accuracy.

It should be noted that the procedure introduced in [15] is capable of estimating the noise variance by using ν-Support Vector Regression [41] and ϵ-insensitive loss function, which results in the following constrained optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha}_j, \{b_j\}, , \boldsymbol{\xi}_j \geq \mathbf{0}, , \delta \geq 0} \ & \sum_{j=1}^{n} \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j + C \sum_{i=1}^{N} \prod_{j=1}^{n} \xi_{ij} + \nu C N \delta^2 \\ & -\delta \mathbf{1} - \boldsymbol{\xi}_j \leq \boldsymbol{y} - \boldsymbol{K}_j \alpha_{kj} - b_j \leq \delta \mathbf{1} + \boldsymbol{\xi}_j. \end{aligned} \tag{44}$$

The $\boldsymbol{\xi}_j$ are the additional slack variables and δ can be interpreted as standard deviation of the noise. As it can be seen from this equation, the number of weights for each sub-system is equal to the number of the data-points N. In addition to the weights, each sub-system has N slack variables ξ. Furthermore, each sub-system has one bias term, hence the number of the variables for one sub-system is $2N + 1$. Considering the fact that the hybrid system consists of n sub-systems, the total number of system variables will be $n(2N + 1)$ (plus the additional variable δ). As such, the constrained optimization (44) contains $n(2N + 1)$ variables, which is almost twice the number of the variables in the first level of inference. Also to make the matter harder, the optimisation problem in (44) has one additional parameter beside C that should be tuned manually, i.e, ν. Considering that often the solution of a constrained optimization is more difficult and more time consuming, our proposed method obtains this parameter

by solving a set of equations, which can be done with conventional gradient-based methods and without requiring to manually tune any parameters. The model in (43) is identified using the constrained optimization in (44) in 21.4 seconds, while the elapsed time is equal to 6.2 seconds for our proposed method, for which the estimated standard deviation is equal to 0.0743.

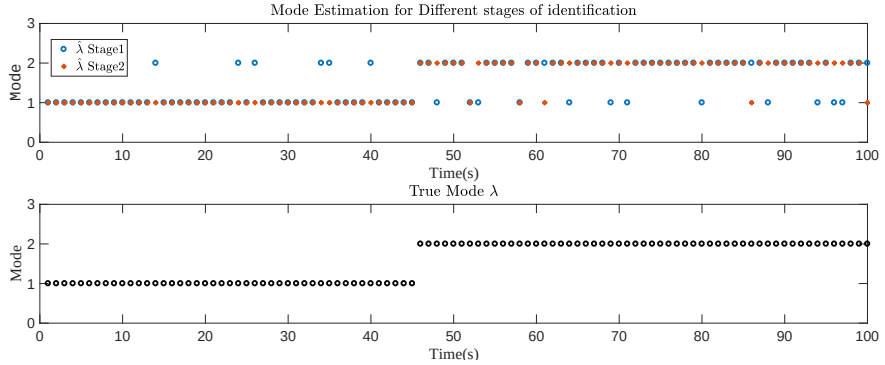


Figure 6: Mode estimation results for state-dependent SNARX model.

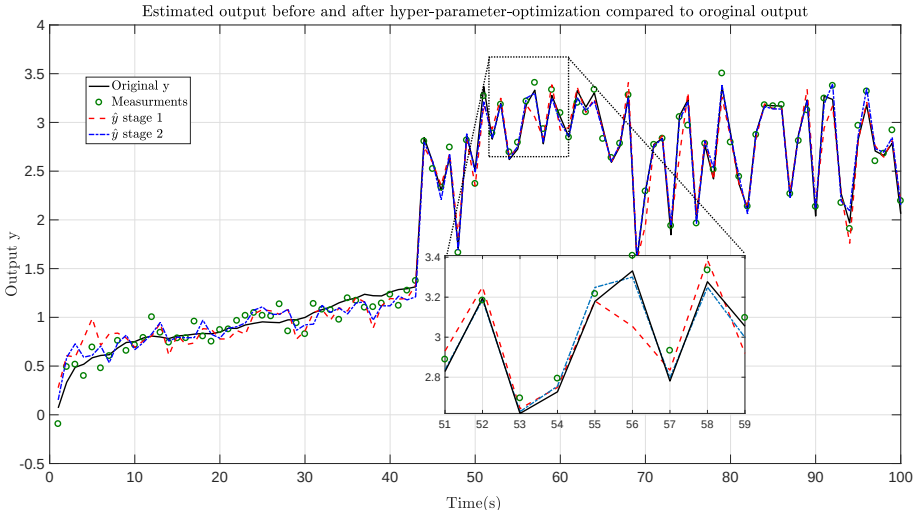


Figure 7: Estimated output results for state-dependent SNARX model.

6.3. Model Comparison and Quality Measure

In this part, the nonlinear hybrid system in (41) is studied under several identification tests in order to study the quality measure introduced in the third level of inference, which helps assessing the quality of the identification when the hybrid system is identified several times with the same kernel parameters, and also when different kernel parameters are selected. First, the system is identified 6 times for the same kernels $\mathcal{H}$ with fixed parameters, which produces different identified models due to the non-convex nature of the problem. Then, the Quality Measure (QM) is used to compare and rank the resulting models and to select the best one. Afterwards, the width of the first kernel is changed and the system is identified with different kernel parameters, and the QM is used to assess the effect of different kernel parameters and compare the results.

Different repetition for the same models. As mentioned before, the identification problem of hybrid systems possesses several near optimal solutions due to its non-convex nature. Thus two different runs of the problem with same parameters might produce different answers. Therefore, one should be able to compare different solutions. The performance of the proposed QM for this condition is verified here. For this purpose, (41) is identified 6 times with two fixed Gaussian models $\mathcal{H} = \{\mathcal{H}_1(0.01), \mathcal{H}_2(1)\}$. The results are presented in Table 1.

These results include: regularization terms (complexity of the model), fitness costs, MSE, percentage of correct data assignments, estimated variances of weights and noise (inverse of μ and β respectively) and the model evidence or Quality Measure. It is worth mentioning that since the last study is about the performance of the QM with regards to different kernel parameters, in each part of this section, a different value is selected for the first kernel and finally the effect of selecting these values are compared in the final case study.

At a first glance and considering only data fitness criteria MSE, it seems that *Case 6* results in the best model; but the QM indicates that *Case 3* is the best model, despite having the third-best MSE. The reason mainly lies within the complexity of the model: the total regularization of model, which is

an indication of its complexity, is lower for *Case 3*. This means that *Case 6* tends to closely match the noisy measurements, hence loosing its generalization features [30]. Besides, it has more correct data-assignments. Similar conclusions can be drawn from the other cases. The QM can be used to compare individual sub-systems. For example, QM for $\mathcal{H}_1$ in *Case 5* i.e, $QM_5(\mathcal{H}_1)$ is higher than *Case 4* ($QM_4(\mathcal{H}_1)$) since it has the better generalization and assigns more data correctly. The exact opposite can be said about $\mathcal{H}_2$.

Parameters		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
Regularization	$\mathcal{H}_1$	0.4464	0.4360	0.4346	0.6726	0.5236	0.7994
	$\mathcal{H}_2$	0.1062	0.1670	0.1095	0.1296	0.1525	0.0963
	Total	0.5526	0.6030	0.5441	0.8022	0.6761	0.8957
Fitness	Cost	0.2255	0.2277	0.2274	0.2036	0.2387	0.1697
	MSE	0.0045	0.0046	0.0045	0.0041	0.0048	0.0034
Data Assignment	$\mathcal{H}_1$	68.33	90	71.66	86.66	92.5	71.66
	$\mathcal{H}_2$	80	61.66	80	67.5	51.66	70
	Overall	73	73	75	79	68	71
Hyper-Parameters	$\sigma^2_{\alpha_1}$	0.0134	0.0074	0.0070	0.0094	0.0078	0.0180
	$\sigma^2_{\alpha_2}$	0.0069	0.0089	0.0095	0.0074	0.0093	0.0044
	σ^2_e	0.0108	0.0069	0.0067	0.0071	0.0068	0.0092
QM	$\mathcal{H}_1$	83.34	109.43	95.60	113.72	119.77	86.36
	$\mathcal{H}_2$	105.18	90.38	107.81	76.68	74.99	99.66
	Total	188.52	199.81	203.42	190.40	194.76	186.02

Table 1: Identification results for 6 different repetition with fixed models.

Different model parameters. In this case, the QM (39) is used for ranking the different models. This time, system (41) is identified using 4 different Gaussian kernel parameters for model $\mathcal{H}_1$, while model $\mathcal{H}_2$ has a fixed parameter equal to 1. The parameters for $\mathcal{H}_1$ are [0.01 0.05 0.1 0.5], of which three were investigated earlier in previous case studies. The results are presented in Table 2. All the models have almost the same MSE. For *Case 2* and *Case 3*, despite having almost the same MSE and data-assignments, QM_3 is higher than QM_2: the reason behind this is the lower complexity and better generalization of the

corresponding model.

Parameters		Case 1	Case 2	Case 3	Case 4
		$\mathcal{H}_1(0.01)$	$\mathcal{H}_1(0.05)$	$\mathcal{H}_1(0.1)$	$\mathcal{H}_1(0.5)$
Regularization	$\mathcal{H}_1$	0.5134	0.2231	0.1546	0.2102
	$\mathcal{H}_2$	0.2431	0.1758	0.1937	0.2237
	Total	0.7565	0.3989	0.3483	0.4339
Fitness	Cost	0.2907	0.2883	0.2973	0.29
	MSE	0.0058	0.0058	0.0059	0.0058
Data Assignment	$\mathcal{H}_1$	65	92.5	85	77.5
	$\mathcal{H}_2$	60	68.33	73.33	88.33
	Overall	62	78	78	84
Hyper-Parameters	$\sigma^2_{\alpha_1}$	0.0176	0.0077	0.0056	0.0112
	$\sigma^2_{\alpha_2}$	0.0044	0.0105	0.0100	0.0071
	σ^2_e	0.0091	0.0092	0.0073	0.0088
Quality (QM)	$\mathcal{H}_1$	83.20	104.91	99.84	77.34
	$\mathcal{H}_2$	100.60	92.53	108.20	131.29
	Total	188.82	197.45	208.04	208.63

Table 2: Identification results for 4 different parameters for $\mathcal{H}_1$

The identification in *Case 4* has better quality than in *Case 1*, partly because of more correct data-assignment, but mainly due to the smaller complexity of the model. This can be seen form Figure 8. *Case 1* ($\mathcal{H}_1(0.01)$) tends to match the noise measurements better than *Case 4*. However, it should be mentioned that some of noise will inevitably fit the model, since some components of noise can not be distinguished from real data.

The two instances *Case 3* and *Case 4* have the same quality. Whilst the later assigns more data correctly, since it is less general than *Case 3*, it is not rated as "significantly better". Furthermore, it can be seen from Figure 9 that *Case 3* ($\mathcal{H}_1(0.1)$) performs better than *Case 4* ($\mathcal{H}_1(0.5)$). This is confirmed by

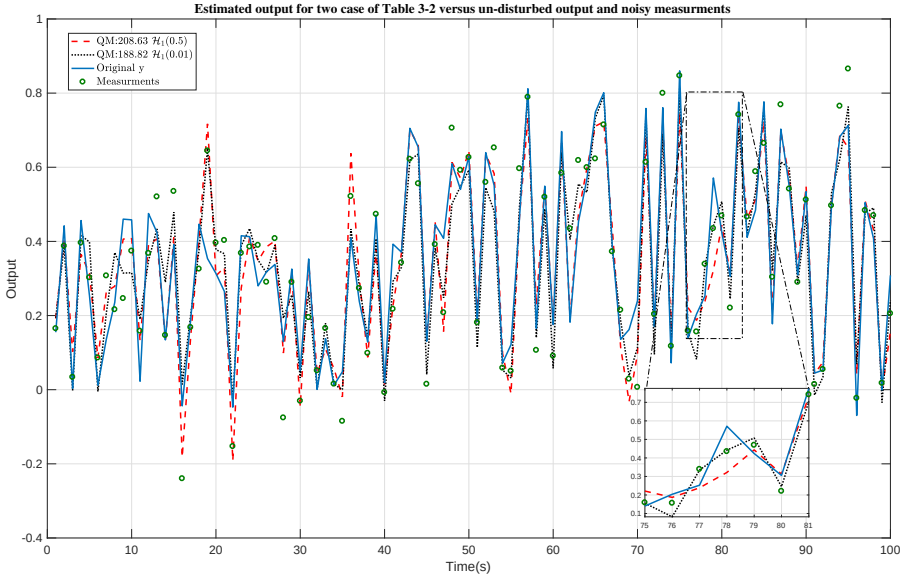


Figure 8: Comparison between *Case 1* and *Case 4* from Table 2.

Table 2, where $QM_3(\mathcal{H}_1)$ is higher than $QM_4(\mathcal{H}_1)$.

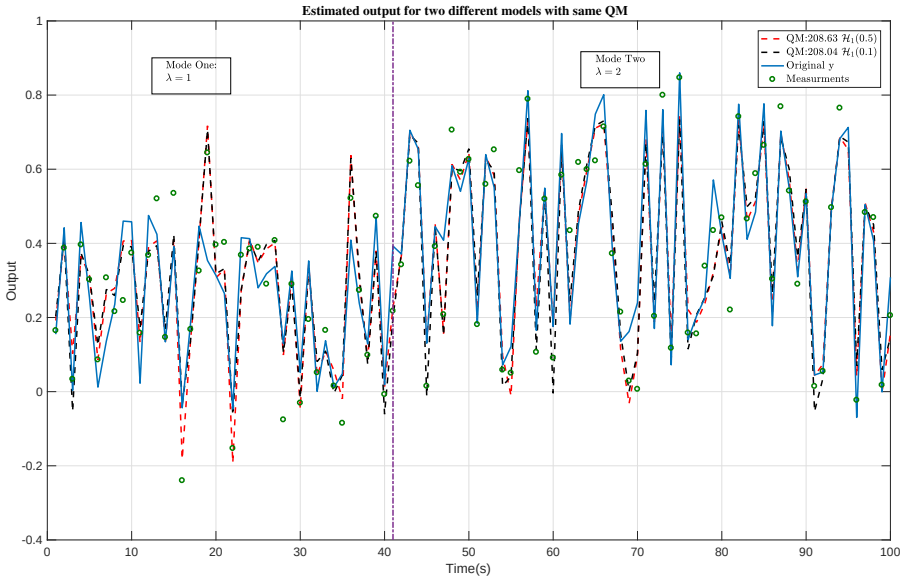


Figure 9: Comparison between *Case 3* and *Case 4* from Table 2.

.

7. Discussion and Conclusions

In this paper, a three-level Bayesian framework for identification of nonlinear hybrid systems has been presented. The parameters of the model (weights and bias terms) are inferred in the first level. At the second level, the variance of the prior distribution for weights, which also controls the complexity of the model, along with the estimated variance of the noise, are calculated. The obtained values from this level cause the output model to be complex enough to fit the data, but not too complex that it looses generalization features. The third level of inference provides a quality measure, in order to compare different models resulting from identification by incorporating all the key ingredients in identification of hybrid systems in a single unified criterion. These ingredients are: model complexity, data fitness, and amount of assigned data points to each sub-system. It can also be used to obtain the best values for the parameters in a given model structure. This framework also gives a probability distribution for prediction, which can be sampled from. The performance of the proposed method has been tested on nonlinear systems with satisfactory results. In addition, the introduced quality measure derived in the third level has been assessed. The results have shown that the quality measure includes all the criteria for assessing the quality of the identification and can be used to choose the best resulting models.

Future work. Future work will focus on extending the proposed framework to multi-output SNARX system and exploiting sparseness to the framework in order to make it suitable for large data-sets. We will also attempt to add robustness to the proposed method with respect to outlier data by considering a different and robust distribution for the likelihood of the data.

References

[1] J. Lunze, F. Lamnabhi-Lagarrigue, Handbook of hybrid systems control: Theory, tools, applications, Cambridge University Press, 2009.

[2] F. Lauer, G. Bloch, R. Vidal, Nonlinear hybrid system identification with kernel models, in: 49th IEEE Conference on Decision and Control, CDC 2010, IEEE, 2010, pp. 696–701.

[3] A. J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and computing 14 (3) (2004) 199–222.

[4] A. L. Juloski, W. Heemels, G. Ferrari-Trecate, R. Vidal, S. Paoletti, J. Niessen, Comparison of four procedures for the identification of hybrid systems, in: International Workshop on Hybrid Systems: Computation and Control, Springer, 2005, pp. 354–369.

[5] Y. Ma, R. Vidal, Identification of deterministic switched ARX systems via identification of algebraic varieties, in: International Workshop on Hybrid Systems: Computation and Control, Springer, 2005, pp. 449–465.

[6] G. Ferrari-Trecate, M. Muselli, D. Liberati, M. Morari, A clustering technique for the identification of piecewise affine systems, Automatica 39 (2) (2003) 205–217.

[7] H. Nakada, K. Takaba, T. Katayama, Identification of piecewise affine systems based on statistical clustering technique, Automatica 41 (5) (2005) 905–913.

[8] A. L. Juloski, S. Weiland, W. Heemels, A Bayesian approach to identification of hybrid systems, IEEE Transactions on Automatic Control 50 (10) (2005) 1520–1533.

[9] Y. Lu, S. Khatibisepehr, B. Huang, A variational Bayesian approach to identification of switched arx models, in: Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on, IEEE, 2014, pp. 2542–2547.

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

[11] A. Bemporad, J. Roll, L. Ljung, Identification of hybrid systems via mixed-integer programming, in: Decision and Control, 2001. Proceedings of the 40th IEEE Conference on, Vol. 1, IEEE, 2001, pp. 786–792.

[12] A. Bemporad, A. Garulli, S. Paoletti, A. Vicino, A greedy approach to identification of piecewise affine models, in: International Workshop on Hybrid Systems: Computation and Control, Springer, 2003, pp. 97–112.

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

[14] R. Vidal, S. Soatto, Y. Ma, S. Sastry, An algebraic geometric approach to the identification of a class of linear hybrid systems, in: Decision and Control, 2003. Proceedings. 42nd IEEE Conference on, Vol. 1, IEEE, 2003, pp. 167–172.

[15] F. Lauer, From support vector machines to hybrid system identification, Ph.D. thesis, Université Henri Poincaré-Nancy I (2008).

[16] A. Hartmann, J. M. Lemos, R. S. Costa, J. Xavier, S. Vinga, Identification of switched ARX models via convex optimization and expectation maximization, Journal of Process Control 28 (2015) 9–16.

[17] G. Pillonetto, A new kernel-based approach to hybrid system identification, Automatica 70 (2016) 21–31.

[18] M. Petreczky, L. Bako, S. Lecoeuche, Minimality and identifiability of sarx systems, IFAC Proceedings Volumes 45 (16) (2012) 541–546.

[19] M. Petreczky, L. Bako, J. H. van Schuppen, Identifiability of discrete-time linear switched systems, in: Proceedings of the 13th ACM international

conference on Hybrid systems: computation and control, ACM, 2010, pp. 141–150.

[20] V. Breschi, A. Bemporad, D. Piga, Identification of hybrid and linear parameter varying models via recursive piecewise affine regression and discrimination, in: 2016 European Control Conference (ECC), IEEE, 2016, pp. 2632–2637.

[21] A. L. Juloski, S. Paoletti, J. Roll, Recent techniques for the identification of piecewise affine and hybrid systems, in: Current trends in nonlinear systems and control, Springer, 2006, pp. 79–99.

[22] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, R. Vidal, Identification of hybrid systems: A tutorial, European journal of control 13 (2-3) (2007) 242–260.

[23] A. Garulli, S. Paoletti, A. Vicino, A survey on switched and piecewise affine system identification, IFAC Proceedings Volumes 45 (16) (2012) 344–355.

[24] F. Lauer, G. Bloch, Switched and piecewise nonlinear hybrid system identification, in: International Workshop on Hybrid Systems: Computation and Control, Springer, 2008, pp. 330–343.

[25] G. Bloch, F. Lauer, et al., Reduced-size kernel models for nonlinear hybrid system identification, IEEE Transactions on Neural Networks 22 (12) (2011) 2398–2405.

[26] L. Bako, K. Boukharouba, S. Lecoeuche, An ℓ_0-ℓ_1 norm based optimization procedure for the identification of switched nonlinear systems, in: Decision and Control (CDC), 2010 49th IEEE Conference on, IEEE, 2010, pp. 4467–4472.

[27] V. L. Le, F. Lauer, L. Bako, G. Bloch, Learning nonlinear hybrid systems: from sparse optimization to support vector regression, in: Proceedings of the 16th international conference on Hybrid systems: computation and control, ACM, 2013, pp. 33–42.

[28] F. Bianchi, M. Prandini, L. Piroddi, A randomized approach to switched nonlinear systems identification, IFAC-PapersOnLine 51 (15) (2018) 281–286.

[29] A. Scampicchio, A. Giaretta, G. Pillonetto, Nonlinear hybrid systems identification using kernel-based techniques, IFAC-PapersOnLine 51 (15) (2018) 269–274.

[30] D. J. MacKay, Bayesian interpolation, Neural computation 4 (3) (1992) 415–447.

[31] D. J. MacKay, Probable networks and plausible predictions: a review of practical Bayesian methods for supervised neural networks, Network: computation in neural systems 6 (3) (1995) 469–505.

[32] T. Van Gestel, J. A. Suykens, D.-E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor, J. Vandewalle, Financial time series prediction using least squares support vector machines within the evidence framework, IEEE Transactions on neural networks 12 (4) (2001) 809–821.

[33] J. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, J. Vandewalle, Least squares support vector machines.,(world scientific publishing: Singapore).

[34] L. Ljung, System identification, in: Signal analysis and prediction, Springer, 1998, pp. 163–173.

[35] F. Lauer, R. Vidal, G. Bloch, A product-of-errors framework for linear hybrid system identification, in: Proc. of the 15th IFAC Symp. on System Identification (SYSID), Saint-Malo, France, 2009.

[36] S. Boyd, L. Vandenberghe, Convex optimization, Cambridge University Press, 2004.

[37] S. S. Keerthi, C.-J. Lin, Asymptotic behaviors of support vector machines with Gaussian kernel, Neural computation 15 (7) (2003) 1667–1689.

[38] M. E. Tipping, Bayesian inference: An introduction to principles and practice in machine learning, in: Advanced lectures on machine Learning, Springer, 2004, pp. 41–62.

[39] T. V. Gestel, J. A. Suykens, G. Lanckriet, A. Lambrechts, B. D. Moor, J. Vandewalle, Bayesian framework for least-squares support vector machine classifiers, Gaussian processes, and kernel Fisher discriminant analysis, Neural computation 14 (5) (2002) 1115–1147.

[40] P. Grünwald, A tutorial introduction to the minimum description length principle, Advances in minimum description length: Theory and applications (2005) 3–81.

[41] B. Schölkopf, A. J. Smola, R. C. Williamson, P. L. Bartlett, New support vector algorithms, Neural computation 12 (5) (2000) 1207–1245.