Information and Computation 289 (2022) 104947

Contents lists available at ScienceDirect

Information and Computation

journal homepage: www.elsevier.com/locate/yinco

Hierarchical identification of nonlinear hybrid systems in a Bayesian framework ☆

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

[a] School of Electrical and Computer Engineering, Tarbiat Modares University, Jalal AleAhmad, Nasr bridge, Tehran, Iran
[b] Mechanical Engineering department, Aarhus University, Inge lehmanns gade 10, Aarhus C, 8000, Denmark
[c] Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, OX1 3QD, Oxford, UK
[d] Department of Computer Science at Aalborg University, Selma Lagerlöfs Vej 300, Aalborg East, 9220, Denmark

ARTICLE INFO

Article history:
Received 19 September 2021
Received in revised form 22 July 2022
Accepted 27 July 2022
Available online 2 August 2022

Keywords:
Nonlinear hybrid systems
Switched nonlinear ARX models
Bayesian inference
System identification

ABSTRACT

This paper presents a hierarchical framework for the identification of nonlinear hybrid systems in the form of Switched Nonlinear AutoRegressive models with eXogenous variables (SNARX). The identification is done via three levels of inference, using Bayes' rule. In the first level, model parameters are computed via a Maximum a Posteriori (MAP) estimator. The posterior distribution therein involved depends on hyper-parameters that are tuned in the second level of inference. Such terms determine model complexity, and the Bayesian framework is key in returning values that trade off complexity with accuracy by automatically embodying the Occam's razor principle. Lastly, the third level compares different model structures by means of a quality measure that encompasses data fitness, model complexity, and data classification. The proposed framework is compared with existing relevant methods and is tested on different numerical models, showing promising performance.

1. Introduction

A Hybrid System (HS) is a dynamical system that consists of discrete components with continuous dynamics. For instance, a hybrid model can be useful when the input to the continuous dynamics has some dependencies on discrete components, and similarly the input to the discrete dynamics is determined by the continuous ones [1]. In other words, a HS comprises more than one dynamic sub-systems and the output at a specific time is determined by the governing sub-system at that time. A "switched system" is a special case of a hybrid system, obtained as a dynamical system that switches between several continuous dynamics. In other words, a switched system is formed by neglecting the detailed behaviour of the discrete dynamics and by thus treating all possible switching patterns from a certain class [1]. HSs have attracted considerable attention in the past few years. The theory of hybrid systems enables multi-disciplinary research combining ideas from computer science, software engineering and digital electronics with system theory and control engineering. HSs can be used for modelling a digital control component interacting with a physical system, and also to improve the modelling

☆ This paper was not presented at any IFAC meeting.

* Corresponding author.
E-mail addresses: amadary@mpe.au.dk (A. Madary), momeni_h@modares.ac.ir (H.R. Momeni), aabate@cs.ox.ac.uk (A. Abate), kgl@cs.aau.dk (K.G. Larsen).

https://doi.org/10.1016/j.ic.2022.104947

of physical systems that have both fast and slow behaviour, for example non-smooth mechanisms, impacting rigid bodies, DC-DC converters, etc. [3]. Aside from the mentioned examples, HSs can also be used to model complex nonlinear systems by means of a collection of simpler linear models [3] together with switching criteria. In our framework, a HS in the form of a Switched Auto-Regressive Exogenous (SARX) system can be defined as

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

where $\boldsymbol{x}_i = \begin{bmatrix} y_{i-1} & \dots & y_{i-n_a} & u_{i-1-n_k} & \dots & u_{i-n_b-n_k} \end{bmatrix}$ 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. More information about the methods for obtaining n_b, n_a, and n_k can be found in [25]. The exogenous, time-dependent variable $\lambda_i \in \{1, \dots, n\}$ denotes the discrete mode and it determines which of the n sub-systems is active at that specific time (which means the corresponding dynamics are characterised by the terms f_{λ_i}). If the functions f_{λ_i} are nonlinear, then the resulting system is a Switched Nonlinear ARX (SNARX) system.

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 of the following form [6]:

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

where the weights $\boldsymbol{\alpha}_j = \begin{bmatrix} \alpha_{1j} & \dots & \alpha_{Nj} \end{bmatrix}^T$ and the bias term b_j are the parameters of the j^{th} sub-system, and $k_j(\cdot)$ is a kernel function that satisfies Mercer's condition [27] and represents the model structure $\mathcal{H}_j$. It should be mentioned that this structure emerges from the Representer Theorem in the field of non-parametric function estimation within Reproducing Kernel Hilbert Spaces [37].

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 Hybrid System (NHS) 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 $\lambda_i \in \{1, \dots, n\}$ 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 a HS reduces to a conventional identification of each sub-system; whereas if the dynamics of sub-systems are known, it becomes a classification problem [12].

During the last two decades, various methods have been developed for identification of linear hybrid systems (LHSs). The major categories of these methods are: clustering techniques [19], Bayesian approaches [9], mixed integer programming techniques [10], bounded error approaches [11], algebraic approaches [12], and methods based on Support Vector Regression (SVR) [7]. Other methods, such as sum-of-norm optimization [13] and kernel methods using the hybrid stable spline algorithm [29] have been also developed to identify LHSs. Detailed analysis of LHSs identification methods can be found in [20,30].

In the area of identification of NHSs, much less research has been done. Contributions can be classified in two main categories: methods based on SVR and stochastic techniques. [4] extends the SVR method to the hybrid domain. To do this, a kernel expansion (2) is adopted and an ϵ-insensitive loss function, with margin of tolerance equal to δ (a desired accuracy), is used to measure the estimation error. Furthermore, the number of nN slack variables $\xi_{ij}, i = 1, \dots, N, j = 1, \dots, n$ are introduced in the form of n vectors $\boldsymbol{\xi}_j \in \mathbb{R}^N$. The slack variable ξ_{ij} represents the estimation error for data point i when being estimated by j^{th} subsystem. For a given data point $(\boldsymbol{x}_i, y_i)$, there should be at least one subsystem (say f_j) such that $\xi_{ij} = 0$. This leads to the following nonlinear constrained optimization problem:

$$\min_{\{\boldsymbol{\alpha}_j\}, \{b_j\}, \boldsymbol{\xi}_j} \quad \sum_{j=1}^{n} \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j + C \sum_{i=1}^{N} \prod_{j=1}^{n} \xi_{ij}$$
$$-\boldsymbol{\xi}_j - \delta \mathbf{1} \le \boldsymbol{y} - \boldsymbol{K}_j \boldsymbol{\alpha}_j - b_j \mathbf{1} \le \boldsymbol{\xi}_j + \delta \mathbf{1}, \; j = 1, \dots, n,$$

where $\boldsymbol{K}_j$ is the j^{th} kernel matrix and C is the trade-off coefficient between model complexity and estimation error. Finally modes are assigned using minimum error principle: $\hat{\lambda}_i = \arg \min_{j=1,\dots,n} |y_i - f_j(\boldsymbol{x}_i)|$. In order to overcome the constrained optimization, authors in [6] proposed the *Minimum-of-Errors Estimator* (ME) by using a smooth loss function $\ell(\cdot)$ instead of an ϵ-insensitive loss function. This way, model parameters can be obtained by solving the following nonlinear mixed-integer programming problem:

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

where $\mathcal{R}(\boldsymbol{\alpha}_j)$ is a regularization term (for example second norm $\mathcal{R}(\boldsymbol{\alpha}_j) = \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j$). Similarly, discrete modes can be estimated using $\hat{\lambda}_i = \arg\min_{j=1,\dots,n} \ell(y_i - f_j(\boldsymbol{x}_i))$. To alleviate the computational cost of the mixed integer programming nature of the ME estimator, authors in [6] proposed the *Product-of-Errors Estimator* (PE) as a smooth approximation of the ME. This way, the model parameters can be obtained by solving the following nonlinear optimization:

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

As every other kernel-based SVR method, the performance of the methods in [4,6] will worsen when the number of data points increases. To address this issue, [5,6] tried to generate a smaller data set by using reduced-size kernels techniques, such as feature vector selection, or principal component analysis. Then, one of the PE or ME estimators introduced in [6] is used to perform an initial identification, and to estimate a NHS based on the reduced data set. This step assigns data points of the reduced set to each sub-system. This assignment is then used for a re-estimation of each of the sub-systems using conventional nonlinear estimators. The proposed technique is tested on a NHS with two components (one linear and one nonlinear), using a linear and a Gaussian Radial Basis Function (RBF) kernel respectively. However, since in piece-wise models each sub-model is active only in a specific region of the input space, the proposed procedure may be sub-optimal and select support vectors outside of the active region. Notice that this approach requires investigating how to obtain the sparse representation [6], and moreover the effectiveness of the method should be studied if two similar kernels are used for the identification. In [21], a sparse optimization technique based on $\ell_0 - \ell_1$ norms is used for identification. These techniques are extended into SVR and kernel expansion form in [22].

[23] proposes a randomized approach to identify NHSs. It treats the nonlinear systems as a linear combination of some nonlinear functions, more specifically as a polynomial functional expansion where each element is a monomial up to a given order. It employs two probability distributions to assign each data point to a sub model and to select of the model structure of the local models. This method assumes that the switching signal (or an initial estimation of this signal) is known a-priori, then segments the data into multiple sets associated with different NARX modes, and finally computes the parameters of each Nonlinear ARX (NARX) mode. If there is no available knowledge of the switching signal, the switching time is varied over the range of available data points and it exhaustively generates all possible combinations of modes for each switching time, which requires a Monte Carlo approach. The method has several crucial design parameters that influence the convergence of the algorithm and should be chosen carefully by the user to avoid local optima. In [28], a Gaussian approach and stochastic simulations are used to identify a switched system consisting of one linear and one nonlinear sub-system. In this method sub-systems are modelled as Gaussian process. Furthermore, N unknown state variables are introduced as hyper-parameters that will be estimated using Maximum Likelihood (ML) optimization through Markov Chain Monte Carlo (MCMC) methods. Once the state variables are computed using ML optimization, the sub-models are estimated. [31] uses an Expectation Maximization (EM) framework to identify a specific class of NHSs defined as Switched Markov Nonlinear ARX (SMNARX) systems. Each continuous dynamics is represented by monomials up to a fixed degree. The estimation of the overall models results from estimating the SMNARX parameters, hidden state values and the corresponding transition probabilities using EM techniques. [32] tackles the identification of NHSs by expressing their nonlinear functions as finite-dimensional parameterized polynomial expansions. The use of polynomial expansion in the last two contributions, however, results in the curse of dimensionality. To mitigate this issue, a two-stage iterative method is developed where, in the first stage, the NARX models are identified using a set of switching times, whereas the second stage tries to refine the switching times - these tasks are done with randomized methods. [33] uses an EM technique to identify switched nonlinear systems with multiple Hammerstein models. The method consists of two stages: model selection and parameter identification. The EM method is used to select the models, while the parameters of the models are identified using a weighted multi-innovation least square (LS) algorithm.

While the reviewed polynomial expansion models [23,32] have attractive features and can represent a wide range of nonlinear functions, they are still not as general as kernel-based models. The cited randomized methods [23] are stymied by the presence of several hyper parameters that will affect their outcome and require fine tuning by the user. Furthermore, they rely on a prior knowledge of switching time: without it, the complexity of the problems will greatly increase.

Conversely, the SVR-based techniques are very versatile and can be applied to all kind of nonlinear functions. However, SVR-based techniques have several shortcomings. In SVR-based methods, the output is a point-wise prediction rather than a posterior distribution that can allow to capture uncertainty in the resulting predictions [34]. Furthermore, the coefficient that determines the trade-off between the complexity of the model and the fitness to data 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 (i.e., 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. To choose the best kernels, the identification results for different kernels should be compared with each other. Furthermore, the identification of NHSs with SVR-based method will result in a non-convex optimization problem, which possesses many near optimal solutions [7]. 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

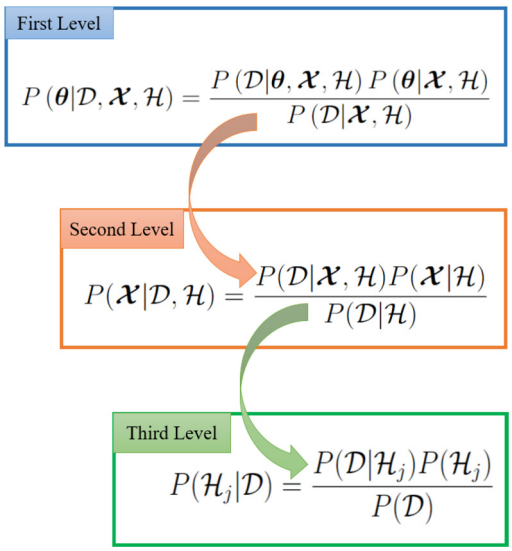

Fig. 1. The three levels of the Bayesian framework.

(minimum error). However, using the fitness criterion alone is not sufficient as more complex models will 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.

This paper aims to address the aforementioned shortcomings of SVR-based techniques in identification of NHSs. Here, a three-level Bayesian framework [2] is introduced for identification of NHSs. 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 sampled in methods such as MCMC.

Compared with [4], 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 [4–6,22], 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. It is worth mentioning that the most important choice in SVR-based methods is the choice of the kernel functions that is done by the user. In [4–6,22], the comparison between different types of kernels or different hyper-parameters for kernels is possible only by benchmarking data-fitness criteria and choosing the kernel with the best data fitness. Instead, the proposed method in this paper provides a unified quality measure to comprehensively compare different kernels and parameters by considering their complexity, uncertainty, data fitness, and mode estimation. It should be finally remarked that it is still the user who decides on the type of the kernel functions.

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 introduced 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 Fig. 1. As it can be seen in Fig. 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:

- The vector of the model parameters, $\boldsymbol{\theta} = [\boldsymbol{\alpha}, \boldsymbol{b}]^T$: here $\boldsymbol{\alpha}$ is the vector of the model 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 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 $\mathcal{H} = \{\mathcal{H}_j | j = 1, ..., n\}$: this is a set containing kernels for each sub-system. These kernels can be of the same type and have the same parameters (for example, Gaussian kernels with the same length); they could be the same kernel types with different parameters for each sub-system (for example, Gaussian kernels with different lengths); or be different kernel types for each sub-system (for example, a Gaussian kernel for one sub-system and a polynomial kernel for another).

3. First level of inference: model parameters

At the first level of inference, the vector of model parameters $\boldsymbol{\theta}$ is calculated through maximizing their posterior probabilities. The conditional posterior probability of the model parameters given the training data set consisting of N points $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$, the vector of hyper-parameters $\boldsymbol{\mathcal{X}}$ and the family of the kernels $\mathcal{H}$ is calculated according to the Bayes' rule

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

where $P(\boldsymbol{\theta}|\boldsymbol{\mathcal{X}}, \mathcal{H})$ is the prior probability distribution of the model parameters. The term $P(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\mathcal{X}}, \mathcal{H})$ is the likelihood of the data points. The denominator of the equation (3) is called the hyper-parameter evidence and is usually ignored in the calculation process of the model parameters since as it will be shown, it is not a function of model parameters [2].

3.1. Prior probability of the model parameters

To calculate the prior distribution of the model parameters, it is assumed that the parameters of each sub-system are independent from those of other sub-systems. Furthermore, $\boldsymbol{\alpha}_j$ and b_j are independent for each sub-system [2,17]. Thus, the conditional probability of the prior distribution over model parameters is:

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

It should be mentioned that while it is not possible to assume that the outputs 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) can be sensibly made.

Next, it is assumed that the prior distribution of $\boldsymbol{\alpha}_j$ of the j^{th} sub-system in (4) has a normal distribution with zero mean and covariance matrix equal to $\mu_j^{-1} I_N$:

$$P\left(\boldsymbol{\alpha}_j|\boldsymbol{\mathcal{X}}, \mathcal{H}\right) = \frac{1}{Z_{\boldsymbol{\alpha}_j}} e^{-\frac{\mu_j}{2}\boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j}, \quad Z_{\boldsymbol{\alpha}_j} = \left(\frac{2\pi}{\mu_j}\right)^{\frac{N}{2}}. \tag{5}$$

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 [2]. In (5), 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 (4) is the prior probability distribution of the bias terms. In this paper, we adopt an uninformative distribution, due to the lack of prior information and for generality [2]. Under the assumption of non-informative prior distribution for b_j and of normal prior distribution (5) for $\boldsymbol{\alpha}_j$, the prior distribution of the model parameters can be written as:

$$P(\boldsymbol{\alpha}, \boldsymbol{b}|\boldsymbol{\mathcal{X}}, \mathcal{H}) = \frac{1}{\prod_{j=1}^{n} Z_{\boldsymbol{\alpha}_j}} e^{\left(\sum_{j=1}^{n} -\frac{\mu_j}{2}\boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j\right)}. \tag{6}$$

3.2. Likelihood of the first level of inference

The conditional distribution of $P(\mathcal{D}|\boldsymbol{\alpha}, \boldsymbol{b}, \boldsymbol{\mathcal{X}}, \mathcal{H})$ is the likelihood term that can be seen as a model of the system noise that disturbs the measured training data. 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 [7]. The maximum likelihood mode estimation for HSs tries to assign each data point $(\boldsymbol{x}_i, y_i)$ to the 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:

$$\hat{\lambda}_i = \underset{j=1,\dots,n}{\arg\max}\ P(y_i|\boldsymbol{x}_i, \hat{f}_j), \qquad P(y_i|\boldsymbol{x}_i, \hat{f}_j) = \frac{e^{-\ell\left(y_i - \hat{f}_j(\boldsymbol{\alpha}_j, b_j, \boldsymbol{x}_i)\right)}}{Z_\delta}, \tag{7}$$

where $\ell(.)$ is a proper loss function and Z_δ is a normalizing constant that will be made explicit in the sequel. Moreover, $\hat{f}_j$ 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 $y_i - \hat{f}_j(\boldsymbol{\alpha}_j, b_j, \boldsymbol{x}_i)$ in (7) is the prediction error and $P(y_i|\boldsymbol{x}_i, \hat{f}_j)$ is the probability density function (PDF) of the prediction errors [25]. A typical assumption is that the prediction errors are independent (more information regarding this assumption can be found in Chapter 5 of [25]). Hence, the complete likelihood of the data can be written as follows where $Z_\delta = \left(\frac{2\pi}{\beta}\right)^{\frac{1}{2}}$:

$$\begin{aligned} P(\mathcal{D}|\boldsymbol{\alpha}, \boldsymbol{b}, \boldsymbol{\mathcal{X}}, \mathcal{H}) &= \prod_{i=1}^{N} \underset{j=1,\dots,n}{\arg\max}\ P\left(y_i - \hat{f}_j(\boldsymbol{\alpha}_j, b_j, \boldsymbol{x}_i)\right) \\ &= \prod_{i=1}^{N} \frac{1}{Z_\delta} e^{\underset{j}{\arg\max} -\frac{\beta}{2}\left(y_i - \hat{f}_j(\boldsymbol{\alpha}_j, b_j, \boldsymbol{x}_i)\right)^2}. \end{aligned} \tag{8}$$

3.3. Posterior distribution of the model parameters

The posterior probability of the model parameters is calculated by combining the prior distribution of parameters (6) and the complete likelihood of the data (8) as:

$$\begin{aligned} P(\boldsymbol{\alpha}, \boldsymbol{b}|\mathcal{D}, \boldsymbol{\mathcal{X}}, \mathcal{H}) &= \frac{\prod_{i=1}^{N} Z_\delta^{-1} \prod_{j=1}^{n} Z_{\boldsymbol{\alpha}_j}^{-1} e^{-\mathcal{J}_1(\boldsymbol{\alpha}, \boldsymbol{b})}}{P(\mathcal{D}|\boldsymbol{\mathcal{X}}, \mathcal{H})}, \\ \mathcal{J}_1(\boldsymbol{\alpha}, \boldsymbol{b}) &= \sum_{j=1}^{n} \frac{\mu_j}{2} \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j + \frac{\beta}{2} \sum_{i=1}^{N} \underset{j=1,\dots,n}{\arg\min}\ \left(y_i - \hat{f}_j(\boldsymbol{\alpha}_j, b_j, \boldsymbol{x}_i)\right)^2. \end{aligned} \tag{9}$$

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. 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

$$\min_{\boldsymbol{\alpha}, \boldsymbol{b}} \sum_{j=1}^{n} \frac{\mu_j}{2} \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j + \frac{\beta}{2} \sum_{i=1}^{N} \underset{j=1,\dots,n}{\arg\min}\ \left(y_i - \hat{f}_j(\boldsymbol{\alpha}_j, b_j, \boldsymbol{x}_i)\right)^2. \tag{10}$$

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 issue is shared by every kernel-based method, unless some mitigation measures, such as dimension reduction techniques, are used to reduce the number of variables [5]. To mitigate this issue, the authors have extended the presented framework to deal with the identification of NHSs with large scale data sets [36]: whilst for the sake of space and focus we do not discuss the details of this extension, we remark that it promises to provide a reduction of the identification time and to improve the quality of the identified models when presented with large data sets.

After calculating the optimal values for the sub-system parameters through (10), 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 sub-systems is calculated. The data point belongs to the sub-system with the highest probability. Substituting the optimal values of the sub-system parameters obtained earlier in the maximum likelihood estimation (MLE) (7) results in:

$$\hat{\lambda_i} = \underset{j=1,\dots,n}{\arg\max}\ P(y_i|\boldsymbol{x}_i, \hat{f}_j(.; \boldsymbol{\alpha}^{MAP}, \boldsymbol{b}^{MAP})), \tag{11}$$

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

Remark on Equation (10)**.** This equation requires solving a continuous-discrete optimization problem. To avoid the optimization problem on both continuous and discrete variables, [15] 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 [16]. 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

$$MinLSE(x_1, ..., x_n) = -\kappa^{-1} \log \left(\sum_{j=1}^{n} \exp(-\kappa x_j) \right), \tag{12}$$

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:*

$$\min\{x_1, ..., x_n\} - \kappa^{-1} log(n) \leq MinLSE(x_1, ..., x_n) < \min\{x_1, ..., x_n\}. \tag{13}$$

Proof. We will begin by writing $\min_{j=1,...,n} \{x_j\}$ in the following form:

$$\min_{j=1,...,n} \{x_j\} = -\kappa^{-1} \log \left(\exp \left(\max_{j=1,...,n} \{-\kappa x_j\} \right) \right).$$

The logarithm on right hand side has the following upper bound:

$$\begin{aligned} \log \left(\exp \left(\max_{j=1,...,n} \{-\kappa x_j\} \right) \right) &< \log \left(\sum_{j=1}^{n} \exp(-\kappa x_j) \right) \\ &\leq \log \left(n \times \exp(-\kappa x^*) \right) = \log n - \kappa x^*, \end{aligned} \tag{14}$$

where x^* is the minimum of $\{x_1, ..., x_n\}$. By multiplying (14) with $-\kappa^{-1}$, the lower and upper bounds in (13) are obtained. It should be evident that with a proper κ, this lower bound can be made sufficiently small. □

Using the MinLSE function, the optimization problem (10) is re-written as follows:

$$\min_{\boldsymbol{\alpha}, \boldsymbol{b}} \ \mathcal{J}_1 = \min_{\boldsymbol{\alpha}, \boldsymbol{b}} \ \sum_{j=1}^{n} \frac{\mu_j}{2} \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j + \frac{\beta}{2} \sum_{i=1}^{N} \underset{j=1,...,n}{MinLSE} \left(\left(y_i - \hat{f}_j(\boldsymbol{\alpha}_j, b_j, \boldsymbol{x}_i) \right)^2 \right), \tag{15}$$

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 [2]. The posterior can be approximated locally with a Gaussian distribution as:

$$P(\boldsymbol{\theta}|\mathcal{D}, \boldsymbol{\mathcal{X}}, \mathcal{H}) \approxeq P(\boldsymbol{\theta}^{MAP}|\mathcal{D}, \boldsymbol{\mathcal{X}}, \mathcal{H}) \exp \left(-\frac{1}{2} \triangle \boldsymbol{\theta}^T \Sigma \triangle \boldsymbol{\theta} \right), \tag{16}$$

where $\boldsymbol{\theta}^{MAP} = [\boldsymbol{\alpha}^{MAP}, \boldsymbol{b}^{MAP}]^T$ and $\triangle\boldsymbol{\theta} = \boldsymbol{\theta} - \boldsymbol{\theta}^{MAP}$. In (16), Σ is the Hessian matrix, namely $\Sigma = -\bigtriangledown^2 \log P(\boldsymbol{\alpha}, \boldsymbol{b}|\mathcal{D}, \boldsymbol{\mathcal{X}}, \mathcal{H})$, and the covariance 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 [2], since the chosen prior is in fact conjugate to the likelihood, hence the posterior is of the same class [38].

After the most probable values of parameters have been obtained, the mode estimation will be done using (11) and values of λ_i are calculated for each data point. The estimated modes can be encoded in a discrete variable B_{ij} as follows:

$$\begin{aligned} &B_{ij} \in \{0, 1\}, \ \ \forall i = 1, ..., N \ \ j = 1, ..., n, \\ &\text{s.t} \ \ B_{ij} = 1 \ \text{ iff } \ \lambda_i = j \ \text{ and } \ \sum_{j=1}^{n} B_{ij} = 1, \end{aligned} \tag{17}$$

which encodes each data point to a sub-system. Introducing this discrete variable into (10), the cost function $\mathcal{J}_1$ can be re-written as

$$\mathcal{J}_1 = \sum_{j=1}^{n} \frac{\mu_j}{2} \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j + \frac{\beta}{2} \sum_{i=1}^{N} \sum_{j=1}^{n} B_{ij} \left(y_i - \hat{f}_j(\boldsymbol{\alpha}_j, b_j, \boldsymbol{x}_i) \right)^2 . \tag{18}$$

Remark. The first term in this equation is called *regularization*, which expresses the kind of desired smoothness from the resulting model [2]. 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 $(1/\beta)$. The necessity of obtaining the optimal values of the hyper parameters is that, even for the case of dynamical non-HSs, 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 [2, 17,18] (depending on the values of model parameters and the ratio β/μ_j). For HSs, this is even more important, since the purpose is not only to fit models on the data, but also to estimate the switching sequence. Improper values for μ_j and 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 MCMC methods, this separation becomes important [17].

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:

$$P(\boldsymbol{\mathcal{X}}|\mathcal{D}, \mathcal{H}) = \frac{P(\mathcal{D}|\boldsymbol{\mathcal{X}}, \mathcal{H}) P(\boldsymbol{\mathcal{X}}|\mathcal{H})}{P(\mathcal{D}|\mathcal{H})}, \tag{19}$$

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 [17] (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 [2]. 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 $P(\mathcal{D}|\boldsymbol{\mathcal{X}}, \mathcal{H})$ 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 $\boldsymbol{\theta} = [\boldsymbol{\alpha} \ \ \boldsymbol{b}]^T$. The evidence of the first level is calculated by marginalizing over model parameters using the following integral [2]:

$$P(\mathcal{D}|\boldsymbol{\mathcal{X}}, \mathcal{H}) = \int P(\mathcal{D}|\boldsymbol{\theta}, \boldsymbol{\mathcal{X}}, \mathcal{H}) P(\boldsymbol{\theta}|\boldsymbol{\mathcal{X}}, \mathcal{H}) d\boldsymbol{\theta}. \tag{20}$$

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 $\triangle\boldsymbol{\theta}$ [17]. The best fit likelihood is multiplied by the Occam's factor, which is less than one and penalises model $\mathcal{H}$ for having parameter $\boldsymbol{\theta}$:

$$\underbrace{P(\mathcal{D}|\boldsymbol{\mathcal{X}}, \mathcal{H})}_{\text{Evidence}} \approx \underbrace{P(\mathcal{D}|\boldsymbol{\theta}^{MAP}, \boldsymbol{\mathcal{X}}, \mathcal{H})}_{\text{Best Fit Likelihood}} \ \ \underbrace{P(\boldsymbol{\theta}^{MAP}|\boldsymbol{\mathcal{X}}, \mathcal{H}) \triangle \boldsymbol{\theta}}_{\text{Occam's Factor}} . \tag{21}$$

The known Occam's Razor principle states that a model should be sufficiently complex to fit the data or, in other words, that a model should not be overtly complex. Complex models which possess a lot of parameters that can take values in a broad interval will be typically penalized with a large Occam factor, compared to simple models [2]. The Occam factor rewards simpler models. This factor also penalizes models that need to be tuned finely to fit the data [2]. In other words, it encourages models that require rough precision on their parameters [2]. The integral (20) can be approximated locally as a Gaussian distribution with covariance matrix Σ, as follows:

$$P(\mathcal{D}|\boldsymbol{\mathcal{X}}, \mathcal{H}) = \prod_{j=1}^{n} Z_{\boldsymbol{\alpha}_j}^{-1} \prod_{i=1}^{N} Z_{\delta}^{-1} e^{-\mathcal{J}_1(\boldsymbol{\theta}^{MAP})} (2\pi)^{\frac{n(N+1)}{2}} |\Sigma|^{-\frac{1}{2}}, \tag{22}$$

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 (15).

Remark on the choice of 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 which includes the prior, and these integrals might be difficult to calculate, assuming Gaussian priors allow to use the Gaussian approximation of the integral [2,17].

The hyper-parameters μ_j 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 [14]. One of the most interesting aspects of the Bayesian approach is that it embodies the Occam's razor principle, that is, this framework automatically selects simpler models. In other words, the Bayesian framework automatically prefers models that sufficiently explain the data without unnecessary complexity [14] and this property holds even if the prior probability is completely uninformative [2].

To obtain the most probable values of the hyper-parameters, the posterior probability (22) should be maximized. Thus, the cost function of the second level can be calculated as:

$$\mathcal{J}_2 = -\frac{N}{2}\left(\sum_{j=1}^{n}\log\mu_j + \log\beta\right) + \frac{N-n}{2}\log 2\pi + \mathcal{J}_1(\boldsymbol{\theta}^{MAP};\boldsymbol{\mu},\beta) + \frac{1}{2}\log|\Sigma|. \tag{23}$$

The Hessian matrix Σ can be written as:

$$\Sigma = \begin{pmatrix} M_\mu + \beta H_1 & \beta H_2 \\ \beta H_2^T & \beta H_3 \end{pmatrix}, \tag{24}$$

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

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

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

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: $H_{aj} = H_{1j} - H_{2j}H_{3j}^{-1}H_{2j}^T$, $j = 1, ..., n$. Using these notations, the determinant of Σ can be expressed as

$$|\Sigma| = \prod_{j=1}^{n} |\mu_j I_N + \beta H_{aj}|\ \ |\beta H_3|. \tag{25}$$

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

$$\log|\Sigma| = \sum_{j=1}^{n}\left((N-k_j)\log\mu_j + \sum_{l=1}^{k_j}\log\left(\mu_j + \beta\lambda_l(H_{aj})\right)\right) + n\log\beta + \sum_{t=1}^{n}\log\lambda(H_{3t}), \tag{26}$$

where k_j is the number of non-zero eigenvalues of H_{aj}, which is only a function of the kernel and training data points. It should be mentioned that $\lambda(\cdot)$ denotes the eigenvalues of the matrix, while $\lambda_l(\cdot)$ represents the l^{th} eigenvalue of the matrix.

4.2. Optimal values of the hyper-parameters

To calculate the most probable values for the hyper-parameters μ_j^{MAP} and β^{MAP}, the posterior distribution (22) 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 (23)) with respect to μ_j can be expressed as:

$$\frac{\partial\mathcal{J}_2}{\partial\mu_j} = -\frac{k_j}{2\mu_j} + \frac{1}{2}\sum_{l=1}^{k_j}\frac{1}{\mu_j + \beta\lambda_l(H_{aj})} + \frac{1}{2}\parallel\boldsymbol{\alpha}_j^{MAP}\parallel_2^2. \tag{27}$$

Variance of the noise β**.** Using the same procedure, the derivative of $\mathcal{J}_2$ with respect to β is:

$$\frac{\partial \mathcal{J}_2}{\partial \beta} = \frac{1}{2}\sum_{j=1}^{n}\sum_{l=1}^{k_j}\frac{\lambda_l(H_{aj})}{\mu_j + \beta\lambda_l(H_{aj})} + \frac{n-N}{2\beta} + \sum_{i=1}^{N}\sum_{j=1}^{n}\frac{B_{ij}}{2}\left(y_i - f_j(\boldsymbol{x}_i)\right)^2. \tag{28}$$

As also mentioned in [2,8], $\gamma_j = 1 + \sum_{l=1}^{k_j}\frac{\beta\lambda_l(H_{aj})}{\mu_j + \beta\lambda_l(H_{aj})}$ is the number of good parameter measurements for the j^{th} sub-system. Each eigenvalue $\beta\lambda_l(H_{aj})$ determines how strongly the corresponding parameter has been determined by data, while μ_j measures the effect of the prior on the parameters [2]. 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 MLE for the parameters and for the majority of the posterior probabilities might be separated [17].

Remark. Obtaining different values for MLE and maximum a-posteriori estimation is similar to finding the parameter of a Gaussian distribution (m, σ) from N data points. The MLE 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 MLE [2]. □

5. Third level of inference: quality measure

The third level of inference is dedicated to obtaining a *quality measure (QM)* for the identification process to assess the performance of the identification. There are several parameters that affect the identification of HSs and that contribute to the quality of the identified model. The key components include *Data fitness*, *Complexity of the model*, and *the Number of data assignment to each sub-system*.

For conventional, non-hybrid systems, Least-Square Support Vector Machines (LS-SVM) method proposed in [8] incorporates the first two items to rate the identification process and compare different models. But the objective of the identification of HSs is to estimate the parameters of each sub-system and the switching signal simultaneously. Current methods for identification of NHSs can control the complexity of the model using 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 structures. To make the matter more complicated, the amount of assigned data to each sub-system should also be considered to compare the results of different identification procedures. To our knowledge, there is not a unified comprehensive criterion for SVR methods that integrates all these items together for HSs.

Another need for having a unified QM to compare the results of identification is that the current identification problem for NHSs (including the present research and [15]) 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: with such measure, solutions that are closer to the optimal will have a better score.

The purpose of the third level of inference is to provide a comprehensive measure to assess the quality of identification of HSs. 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 task, 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 (e.g. MLE) will result in over-parameterized models with poor generalization. This is where the Occam's razor principle should be used [2]. 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

$$P(\mathcal{H}_j|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{H}_j)P(\mathcal{H}_j)}{P(\mathcal{D})}. \tag{29}$$

The evidence $P(\mathcal{D}|\mathcal{H}_j)$ will be obtained by integrating out all the variables (hyper-parameters $\boldsymbol{\mathcal{X}}_j = [\mu_j \ \ \beta]$):

$$P(\mathcal{D}|\mathcal{H}_j) = \int P(\mathcal{D}|\boldsymbol{\mathcal{X}}_j, \mathcal{H}_j) P(\boldsymbol{\mathcal{X}}_j|\mathcal{H}_j) d\boldsymbol{\mathcal{X}}_j. \tag{30}$$

The evidence can be approximated accurately by two independent normal distributions with confidence intervals σ_{μ_j} and σ_β. These confidence intervals are calculated by differentiating (23) twice with respect to μ_j and β.

Assuming a flat prior the evidence will be calculated as:

$$P(\mathcal{D}|\mathcal{H}_j) \approxeq P(\mathcal{D}|\boldsymbol{\mathcal{X}}_j^{MAP}, \mathcal{H}_j) 2\pi\sigma_\beta\sigma_{\mu_j}, \tag{31}$$

where σ_β and σ_{μ_j} are confidence intervals of the hyper-parameters β and μ_j, respectively. Furthermore, $P(\mathcal{D}|\boldsymbol{\mathcal{X}}^{MAP}, \mathcal{H}_j)$ is calculated by using the most probable values for hyper-parameter which are obtained in the previous level in (22).

The QM for model $\mathcal{H}_j$ is obtained by taking the logarithm of the posterior distribution $P(\mathcal{H}_j|\mathcal{D})$ while neglecting all the constants in (22) and (31) as expressed below:

$$\begin{aligned} QM(\mathcal{H}_j) = {} & \log\sigma_{\mu_j} + \log\sigma_\beta + \frac{k_j}{2}\log\mu_j^{MAP} + \frac{\zeta_j - 1}{2}\log\beta^{MAP} - \frac{1}{2}\log\zeta_j \\ & - \frac{1}{2}\sum_{l=1}^{k_j}\left(\mu_j^{MAP} + \beta^{MAP}\lambda_l(H_{aj})\right) - \frac{\mu_j^{MAP}}{4}||\alpha_j||_2^2 - \frac{\beta^{MAP}}{4}\sum_i^N B_{ij}\left(y_i - f_j(\boldsymbol{x}_i)\right)^2. \end{aligned} \tag{32}$$

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

- Model fitness for $\mathcal{H}_j$: $\sum_i^N B_{ij}\left(y_i - f_j(\boldsymbol{x}_i)\right)^2$ and prior variance of noise $1/\beta^{MAP}$;
- Model Complexity: regularization term $||\alpha_j||_2^2$ and prior variance of weights μ_j^{MAP};
- 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 QM 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 QM for identification should be defined to incorporate all the changes in the overall model family $\mathcal{H}$. This new QM is defined as the summation of the QM for all the models:

$$QM_{Overall} = \sum_{j=1}^n QM(\mathcal{H}_j). \tag{33}$$

Relationship 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 form 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 $-\log P(\mathcal{D}|\mathcal{H})$) [26]. The QM is obtained from the logarithm of (31), which is very similar to the MDL for non-HSs and the Akaike criterion (AIC), which can be seen as the approximation of MDL [2].

6. Case studies

In this section, several case studies are presented to test the performance of the proposed method. Due to the limited results in the field of identification of NHSs, and the fact that the models used in the existing literature are deemed to be fairly simple, the models used for the case studies are devised anew. All simulations are implemented on Matlab R2019a and run on a laptop with Intel Core i5 1.6 GHz CPU and 8 GB of memory. First, the performance of the proposed method in the first level of inference is compared with the PE framework introduced in [7] and used in [4–6,15]. 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 [4–6,15] use dimension reduction and support vector (SV) selection, here for consistency we only implement the optimization formulation using PE. Thus, no SV selection or dimension reduction is done. Furthermore, all the values (Kernel types, kernel parameters, trade-off coefficients) are the same and all the algorithms start from the same initial values.

Afterwards, the performance of the all three levels of inference is studied by identifying two kinds of SNARX systems with different switching characteristics. The systems under study are considered to have: an exogenous (or predefined) switch in time, and a state-dependent switch. The identification is initialized from hyper-parameters in the first level. The hyper-parameters are optimized in the second level and the system is identified again with the optimized values. Finally, the best

identified model is chosen using the QM obtained in the third level. Later, the performance of the third level of inference is verified to compare different models arising from the usage of different kernels or different repetitions of the identification algorithm. Finally, the dependence of the algorithms on the size of the data set is reported. The procedure presented in Algorithm 1 illustrates the explained procedure for identification of NHSs using the proposed hierarchical Bayesian method. It is worth recalling that the goal of this problem is to *jointly* identify the parameters of individual sub-systems as well as the switching signal.

Algorithm 1 Identification of NHSs using hierarchical Bayesian method.

1: *Collect N data points from the HS under study*
2: *Set up the data set in appropriate format* $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$
3: *Form the set of the kernels* $\mathcal{H} = \{\mathcal{H}_j | j = 1, ..., n\}$ *by selection of n kernels*
4: *Initialize the identification from initial hyper-parameter values by applying the first level of inference and obtain model parameters* $\boldsymbol{\alpha}^{MAP}$, $\boldsymbol{b}^{MAP}$ *and discrete mode* $\hat{\lambda}_i$
5: *Use the results of the first level to optimize the hyper-parameters* $\boldsymbol{\mathcal{X}} = [\boldsymbol{\mu}, \beta]$ *using the second level*
6: *Plug the optimized hyper-parameters back in the first level and recalculate the model parameters and discrete modes*
7: *Solve the third level to calculate the quality measure and select the best identified model/kernel structure*

6.1. Performance comparison

In this part, the first level of inference of the proposed method is compared with the PE solver in [7] for identification of the following SNARX model:

$$y_i = \begin{cases} -0.4y_{i-1}^2 + 0.5u_{i-1} + e_i & \text{if } \lambda_i = 1 \\ (0.8 - 0.5e^{y_{i-1}^2})y_{i-1} - y_{i-1}^2 + 0.9u_{i-1} + e_i & \text{if } \lambda_i = 2. \end{cases} \tag{34}$$

For this purpose, the cost function of the first level of inference that includes "MinLSE" approximation, is compared with the cost function of PE framework that used "Product of Errors" as the approximation of minimum function. In [7], the identification is done using the following optimization so-called PE:

$$\min_{\{\boldsymbol{\alpha}_j\},\{b_j\}} \frac{1}{n}\sum_{j=1}^{n}\mathcal{R}(\boldsymbol{\alpha}_j) + \frac{C}{N}\sum_{i=1}^{N}\prod_{j=1}^{n}\ell\left(y_i - \sum_{k=1}^{N}\alpha_{kj}k_j(\boldsymbol{x}_k, \boldsymbol{x}_i) - b_j\right), \tag{35}$$

where $\mathcal{R}(\cdot)$ is the regularization term, $\ell(\cdot)$ 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.

The L_2-norm is used as regulariser ($\mathcal{R}(\boldsymbol{\alpha}_j) = \boldsymbol{\alpha}_j^T\boldsymbol{\alpha}_j$) and a quadratic loss function $\ell(\cdot)$ is employed. To have a fair comparison, all of the hyper-parameters (e.g. regularization term) of the solver for the first level of inference and the PE framework, along with the kernel type and hyper-parameters, are chosen randomly and are equal for both solvers. 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, without optimizing the hyper-parameters in the second level of inference.

The output of system (34) 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 $C = 100$ 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 Fig. 2 and Fig. 3, the results of the first level of inference in the proposed method are better than the PE method from [6] 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.

6.2. Identification of a 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). The identification is initialized with $\boldsymbol{\mu} = [1\ 1]$ and $\beta = 1$, then these hyper-parameters are optimized in the second level and the identification is repeated with the new optimized values. Each system is identified several times and the best identified system is chosen using the QM of the third level, as studied in detail in Section 6.3. The reported values correspond to the best that have been identified according to their QM.

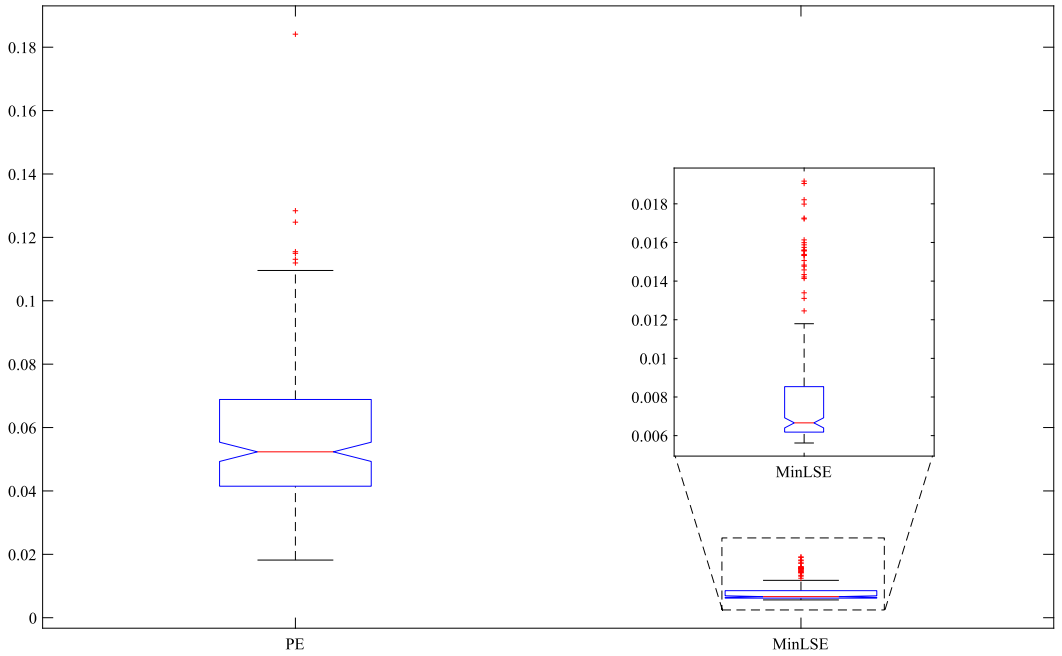

Fig. 2. Comparison of MSE results between proposed method (with MinLSE) and PE method.

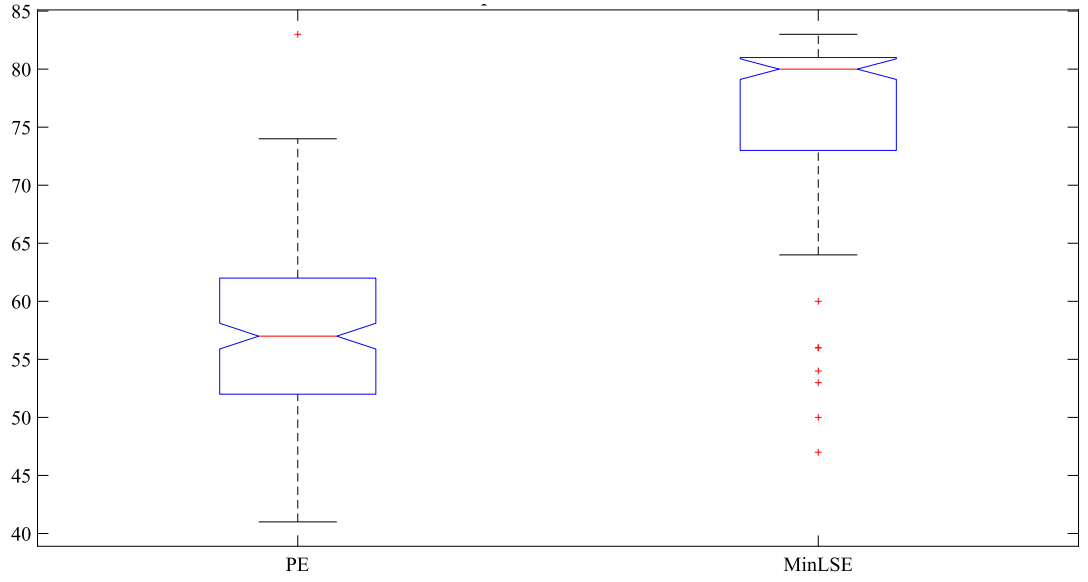

Fig. 3. Comparison of percentage of correct data-assignment between proposed method (with MinLSE) and PE method.

6.2.1. NARX systems with exogenous switching

First, the system in (34) is identified with the set of models $\mathcal{H}$ 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 results of the identification show that 84% of the data points have been assigned correctly and the MSE is only 0.0041.

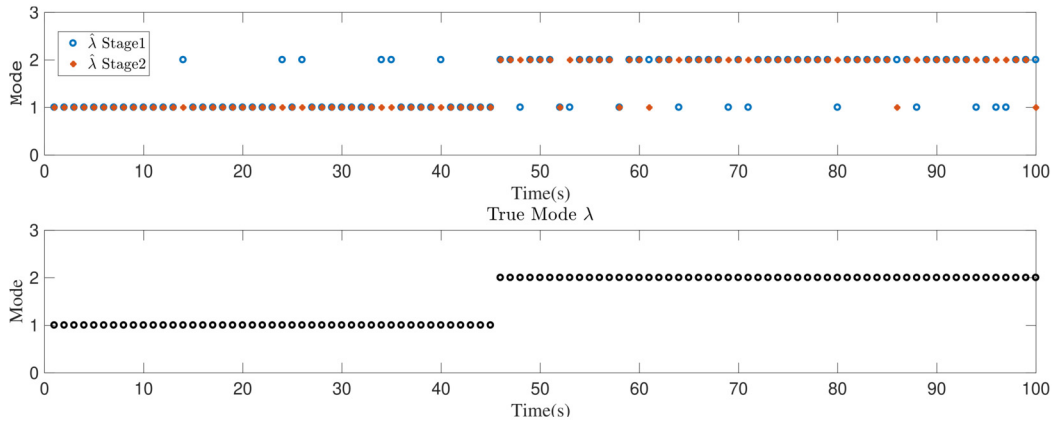

Fig. 4. Mode estimation results for state-dependent SNARX model.

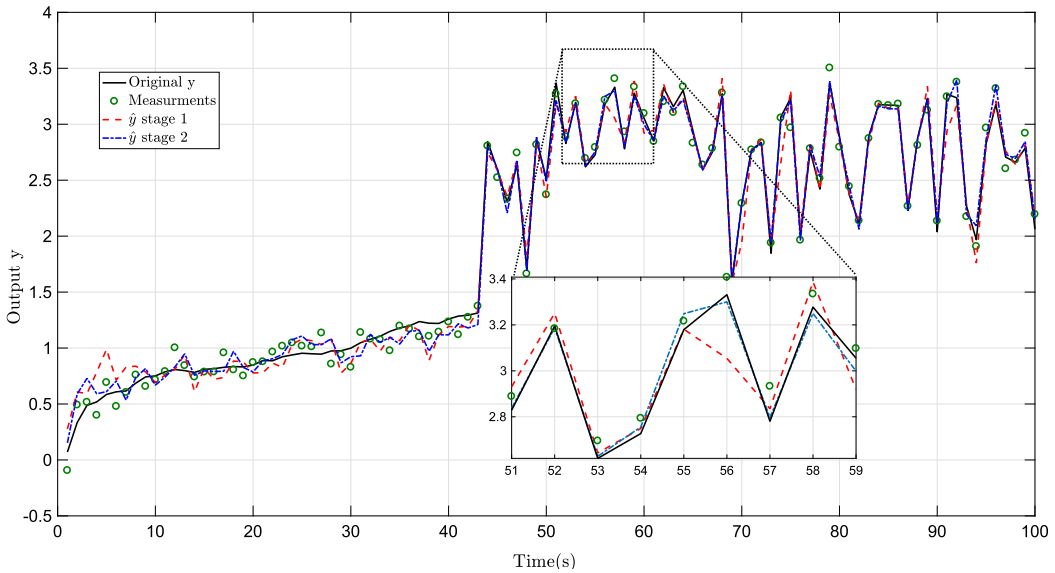

Fig. 5. Estimated output for state-dependent SNARX model.

6.2.2. NARX systems with state-dependent switching

The following system is considered:

$$y_i = \begin{cases} k\sqrt{(\frac{y_{i-1}}{k})^2 + \frac{u_{i-1} - y_{i-1}}{s}} + e_i & \text{if} \quad y_{i-1} \leq Y \\ 1.3 + \sqrt{u_{i-1}} + \exp(-y_{i-1}) + e_i & \text{if} \quad y_{i-1} > Y, \end{cases} \tag{36}$$

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 (SD) equals to 0.1. Again, two Gaussian kernels with parameters set to 0.5 and 1 are chosen: $\mathcal{H} = \{\mathcal{H}_1(0.5), \mathcal{H}_2(1)\}$. Starting from the initialized values, the identification is repeated with the optimized values for hyper-parameters obtained at the second level of inference. The optimized hyper-parameters for the best identified system are: $\boldsymbol{\mu} = [10.98\ \ 11.07]$ and $\beta = 33.94$ which corresponds to 0.1716 for the estimated SD 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 Fig. 4 and Fig. 5. It can be seen from the figures that the overall accuracy of the identification is improved after optimizing the hyper-parameters. Furthermore, the SD of the noise is estimated with relatively good accuracy.

It should be noted that the procedure introduced in [7] is capable of estimating the noise variance by using ν-SVR [24] and ϵ-insensitive loss function, which results in the following constrained optimization problem:

Table 1
Identification results for 6 repetitions with fixed models.

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

$$\begin{aligned} \min_{\boldsymbol{\alpha}_j,\{b_j\},\boldsymbol{\xi}_j\geq\mathbf{0},\delta\geq 0} \quad & \sum_{j=1}^{n}\boldsymbol{\alpha}_j^T\boldsymbol{\alpha}_j + C\sum_{i=1}^{N}\prod_{j=1}^{n}\xi_{ij} + \nu CN\delta^2 \\ & -\delta\mathbf{1}-\boldsymbol{\xi}_j \leq \boldsymbol{y}-\boldsymbol{K}_j\alpha_{kj}-b_j \leq \delta\mathbf{1}+\boldsymbol{\xi}_j. \end{aligned} \tag{37}$$

The $\boldsymbol{\xi}_j$ are the additional slack variables and δ can be interpreted as SD 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 HS 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 (37) 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 (37) 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 (36) is identified using the constrained optimization in (37) in 21.4 seconds, while the elapsed time is equal to 6.2 seconds for our proposed method, for which the estimated SD is equal to 0.0743.

6.3. Model comparison and quality measure

In this part, the NHS in (34) is studied under several identification tests to study the QM introduced in the third level, which helps assessing the quality of the identification when the HS 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 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.

6.3.1. Different repetitions for the same models

As mentioned before, the identification problem of HSs possesses several near optimal solutions due to its non-convex nature. Thus, two different runs of the problem with the same parameters might result in 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, (34) 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 QM. 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 [2]. 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

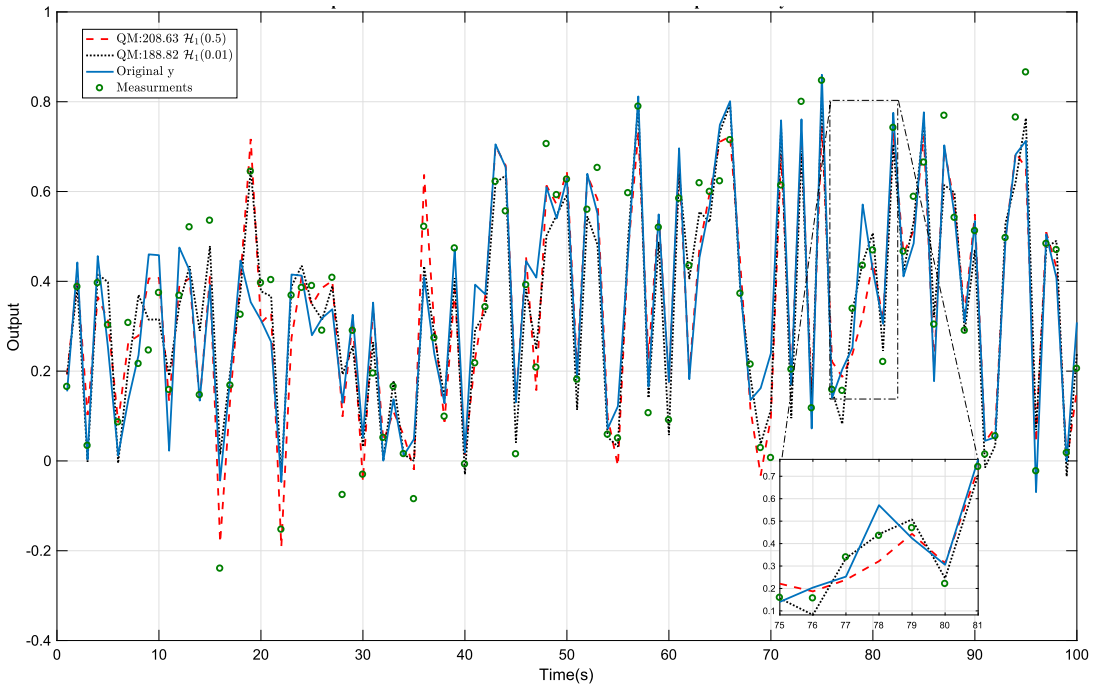

Fig. 6. Comparison between *Case 1* and *Case 4* from Table 2.

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 $\mathcal{H}_2$.

6.3.2. Different model parameters

In this case, the QM in (32) is used for ranking different models. This time, the system in (34) 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 is the lower complexity and better generalization of the corresponding model.

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 from Fig. 6. *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". Table 2 confirms that *Case 3* ($\mathcal{H}_1(0.1)$) performs better than *Case 4* ($\mathcal{H}_1(0.5)$) as $QM_3(\mathcal{H}_1)$ is higher than $QM_4(\mathcal{H}_1)$.

6.4. Dependence on number of data points

Next, the system (36) is identified several times with different number of data points to investigate the effect of the size of the data set on the identification time. The size of the data points is $N = \{100, 200, 400, 800, 1000\}$. These simulations are run using the hardware mentioned before. The optimization in the first level is performed using Matlab *fminunc* function. For each data set, the identification was performed 20 times. The minimum, average, and maximum identification time for the first level of inference corresponding to the size of data sets are shown in Fig. 7. The second level of inference requires only a fraction of a second to be completed (ranging from $t = 0.019s$ for $N = 100$ to $t = 0.3s$ for $N = 1000$). To overcome this issue, the presented framework is extended in [36] to improve the identification time for large data sets.

This is done by splitting the identification problem into a mode estimation part, and a part on regression. More specifically, a reduced data-set $\mathcal{D}$ is constructed from the original large data-set $\mathcal{S}$ using feature vector selection technique presented in [35]. Then the presented method is used as a pre-identification, to assign all the data in $\mathcal{D}$ and create a set of labels $\mathcal{L}$. Each label is in fact the corresponding mode of a data point. The reduced data-set $\mathcal{D}$ and its corresponding label set $\mathcal{L}$ are used as a training set for a classifier [14], that later classifies all the data in the original $\mathcal{S}$ to a sub-system. Having assigning the data to sub-systems, the continuous sub-systems can be estimated using a regressor. Fig. 8 demonstrates

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

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

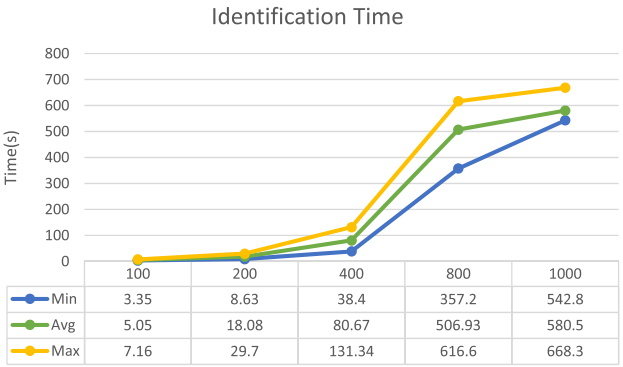

Fig. 7. Effect of the number of the data-points on the identification time.

the aforementioned expansion of the proposed method to large data sets. This method is used to identify the following model [36]:

$$y_i = \begin{cases} y_i = -0.9y_{i-1} + 0.5u_{i-1} + e_i & \text{if } \lambda_i = 1 \\ y_i = (0.8 - 0.5exp(-y_{i-1}^2))y_{i-1} & \\ \quad -0.9y_{i-1}^2 + 0.9u_{i-1} + e_i & \text{if } \lambda_i = 2 \end{cases} \tag{38}$$

$N = 3200$ data points were generated from this system starting from a random initial condition y_0, with a random input uniformly distributed in the range $u_i \in [0\ 4]$ and a Gaussian noise with zero mean and standard deviation equal to 0.1. Two kernels were used for this purpose: a linear kernel and a Gaussian kernel with parameter set to 0.22. While the full identification took $3671.02s$, the extended method solved the problem in just $7.71s$ [36].

Since the pre-identification forms the foundation for which the data are assigned to sub-systems, every measure that can improve the quality of the pre-identification will hugely improve the overall performance. To demonstrate this, system (38) is identified two times: once with an initial guess for the hyper parameters, without optimizing them; and a second time, from the same initial hyper-parameters but with optimized values. The initial guess for the hyper-parameters are $\boldsymbol{\mu} = [1\ \ 1]$ and $\beta = 100$. After optimizing them in the second level of the pre-identification, their values changed to $\boldsymbol{\mu} = [44.0102\ \ 53.7600]$ and $\beta = 1414$. The identification results are reported in Table 3.

It can be seen that optimizing the hyper-parameters in the second level of inference greatly improves the performance of the identification, both in therms of MSE and of data assignment.

7. Discussion and conclusions

In this paper, a three-level Bayesian framework for identification of NHSs 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, is 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 QM to compare different models resulting from identification by incorporating all the key ingredients in identification of HSs in a single unified criterion. These ingredients

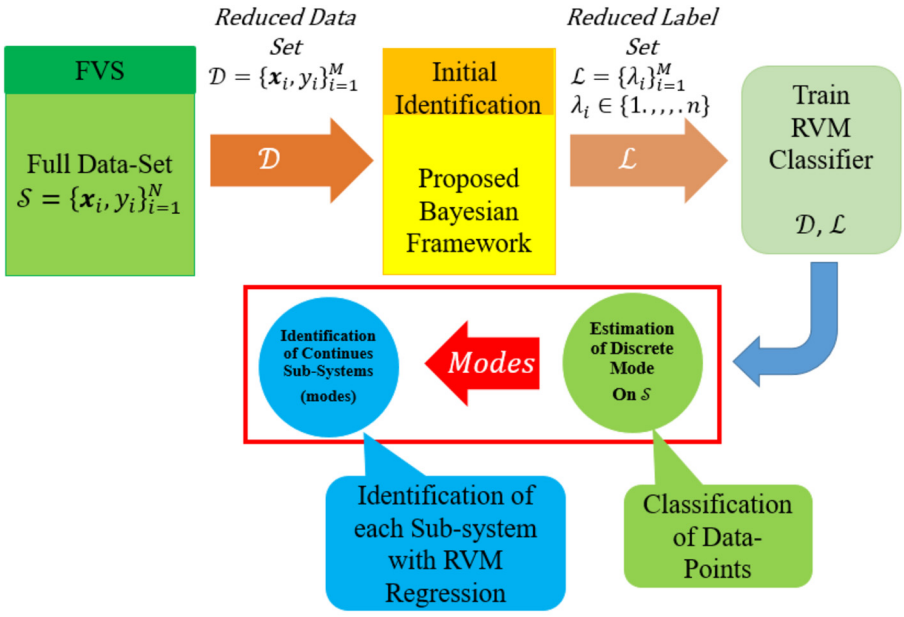

Fig. 8. Extension of the proposed method to deal with large data sets.

Table 3
Effects of the optimized hyper-parameters.

	Hyper-parameters	
	Not-optimized	Optimized
Correct mode assignment on $\mathcal{D}$(%)	81.34	98.34
Classifier accuracy on $\mathcal{D}$(%)	79.32	91.56
Classifier accuracy on $\mathcal{S}$(%)	75.45	84.25
Time for data reduction	16.6	16.6
Time for identification	3.72	4.35
MSE	0.044	0.023

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 QM derived in the third level has been assessed. The results have shown that the QM includes all the criteria for assessing the quality of the identification and can be used to choose the best resulting models.

The presented method can be extended to NHSs generating large data sets [36]: whilst these details are beyond the scope of this work, we remark that [36] can significantly reduce the identification time and outperform existing methods [5] with regard to identification accuracy and mode estimation.

Future work will focus on extending the proposed framework to multi-output SNARX systems. 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.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] D. Liberzon, Switching in Systems and Control, Vol. 190, Birkhauser, Boston, 2003.
[2] D.J.C. MacKay, Bayesian interpolation, Neural Comput. 4 (3) (1992) 415–447.
[3] J. Lunze, F. Lamnabhi-Lagarrigue, Handbook of Hybrid Systems Control: Theory, Tools, Applications, Cambridge University Press, 2009.
[4] F. Lauer, G. Bloch, Switched and piecewise nonlinear hybrid system identification, in: Proceedings of the International Workshop on Hybrid Systems: Computation and Control (HSCC), Berlin, 2008, pp. 330–343.
[5] G. Bloch, F. Lauer, Reduced-size kernel models for nonlinear hybrid system identification, IEEE Trans. Neural Netw. 22 (12) (2011) 2398–2405.

[6] F. Lauer, G. Bloch, R. Vidal, Nonlinear hybrid system identification with kernel models, in: Proceedings of the 49th IEEE Conference on Decision and Control, CDC 2010, 2010, pp. 696–701.
[7] F. Lauer, From support vector machines to hybrid system identification, Ph.D. dissertation, Université Henri Poincaré-Nancy I, 2008.
[8] T. Van Gestel, J.A.K. Suykens, G. Lanckriet, A. Lambrechts, B.D.E. Moor, J. Vandewalle, Bayesian framework for least-squares support vector machine classifiers, Gaussian processes, and kernel Fisher discriminant analysis, Neural Comput. 14 (5) (2002) 1115–1147.
[9] A.L. Juloski, S. Weiland, W.P.M.H. Heemels, A Bayesian approach to identification of hybrid systems, IEEE Trans. Autom. Control 50 (10) (2005) 1520–1533.
[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, A. Garulli, S. Paoletti, A. Vicino, A bounded-error approach to piecewise affine system identification, IEEE Trans. Autom. Control 50 (10) (2005) 1567–1580.
[12] Y. Ma, R. Vidal, Identification of deterministic switched ARX systems via identification of algebraic varieties, in: Proceedings of the International Workshop on Hybrid Systems: Computation and Control (HSCC), 2005, pp. 449–465.
[13] A. Hartmann, J.M. Lemos, R.S. Costa, J. Xavier, S. Vinga, Identification of switched ARX models via convex optimization and expectation maximization, J. Process Control 28 (2015) 9–16.
[14] 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.
[15] F. Lauer, R. Vidal, G. Bloch, A product-of-errors framework for linear hybrid system identification, in: Proceedings of the IFAC Volumes, vol. 42(10), 2009, pp. 563–568.
[16] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[17] D.J.C. MacKay, Probable networks and plausible predictions: a review of practical Bayesian methods for supervised neural networks, Netw. Comput. Neural Syst. 6 (3) (1995) 469–505.
[18] S.S. Keerthi, C.J. Lin, Asymptotic behaviors of support vector machines with Gaussian kernel, Neural Comput. 15 (7) (2003) 1667–1689.
[19] 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.
[20] S. Paoletti, A.L.J. Juloski, G. Ferrari-Trecate, R. Vidal, Identification of hybrid systems: a tutorial, Eur. J. Control 13 (2) (2007) 242–260.
[21] L. Bako, K. Boukharouba, S. Lecoeuche, An ℓ_0-ℓ_1 norm based optimization procedure for the identification of switched nonlinear systems, in: Proceedings of the 49th IEEE Conference on Decision and Control (CDC), 2010, pp. 4467–4472.
[22] 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 (HSCC), 2013, pp. 33–42.
[23] F. Bianchi, M. Prandini, L. Piroddi, A randomized approach to switched nonlinear systems identification, in: Proceedings of the 18th IFAC Symposium on System Identification (SYSID 2018), vol. 51(15), 2018, pp. 281–286.
[24] B. Schölkopf, A.J. Smola, R. Williamson, P.L. Bartlett, New support vector algorithms, Neural Comput. 12 (5) (2000) 1207–1245.
[25] L. Ljung, System Identification. Signal Analysis and Prediction, Birkhäuser, Boston, MA, 1998, pp. 163–173.
[26] P. Grünwald, A Tutorial Introduction to the Minimum Description Length Principle, Advances in Minimum Description Length: Theory and Applications, 2005, pp. 3–81.
[27] A.J. Smola, B. Schölkopf Bernhard, A tutorial on support vector regression, Stat. Comput. 14 (3) (2003) 199–222.
[28] A. Scampicchio, A. Giaretta, G. Pillonetto, Nonlinear hybrid systems identification using kernel-based techniques, in: Proceedings of the 18th IFAC Symposium on System Identification (SYSID 2018), vol. 51(15), 2018, pp. 269–274.
[29] G. Pillonetto, A new kernel-based approach to hybrid system identification, Automatica 70 (2016) 21–31.
[30] A. Garulli, S. Paoletti, A. Vicino, A survey on switched and piecewise affine system identification, in: Proceedings of the IFAC Volumes, vol. 45(16), 2012, pp. 344–355.
[31] A. Brusaferri, M. Matteucci, A. Spinelli, Estimation of switched Markov polynomial NARX models, arXiv preprint arXiv:2009.14073, 2020.
[32] F. Bianchi, M. Prandini, L. Piroddi, A randomized two-stage iterative method for switched nonlinear systems identification, Nonlinear Anal. Hybrid Syst. 35 (2020) 100818.
[33] C. Xiujun, H. Hongwei, W. Lin, X. Zhengqing, Identification of switched nonlinear systems based on EM algorithm, in: Proceedings of the 39th Chinese Control Conference (CCC), 2020, pp. 1337–1342.
[34] M.E. Tipping, The relevance vector machine, Adv. Neural Inf. Process. Syst. (2000) 652–658.
[35] G. Baudat, A. Fatiha, Feature vector selection and projection using kernels, Neurocomputing 55 (1–2) (2003) 21–38.
[36] A. Madary, H.R. Momeni, A. Abate, K.G. Larsen, A Bayesian framework for large-scale identification of nonlinear hybrid systems, in: Proceedings of the 7th IFAC Conference on Analysis and Design of Hybrid Systems, July 7-9, 2021, Brussels, Belgium, 2021.
[37] I. Steinwart, A. Christmann, Support Vector Machines, Springer Science & Business Media, 2008.
[38] R. Schlaifer, H. Raiffa, Applied Statistical Decision Theory, 1st ed., Division of Research, Harvard Business School, 1961.