Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 58-15 (2024) 486–491

An information theory approach for recursive LPV-ARX model identification via LS-SVM

F. Corrini *, **M. Mazzoleni** *, **F. Ferracuti** **, **L. Cavanini** ***, **F.Previdi** *

* *Department of Management, Information and Production engineering, University of Bergamo, Bergamo (BG), Italy.*
** *Department of Information Engineering, Università Politecnica delle Marche 60131 Ancona, Italy.*
*** *Industrial Systems and Control Ltd, Culzean House, 36 Renfield Street, Glasgow G2 1LU, UK.*

Abstract: When modeling a dynamical system, linear models are always the first choice due to their simplicity. However, many times the system is too complex so that employing linear models results in poor performances. In this context, Linear Parameter Varying (LPV) models allow to represent non-linear input/output relationships while preserving the simple structure of linear models. The method of Least Squares Support Vector Machines (LS-SVM) is one of the most common approaches to identify a LPV model in an ARX formulation (LPV-ARX). However, due to its computational cost, it is difficult to identify a LPV-ARX model using LS-SVM in online applications, where the model must be updated every time new data are collected. An efficient update algorithm has been presented for online identification of such models, where the idea is to update the model only upon certain data that are considered informative. However, this approach requires the tuning of some hyperparameters and in certain conditions can stop updating the model even when data are informative. This paper proposes an information-based algorithm to overcome these drawbacks. Evaluation on simulated and experimental data show the effectiveness of the proposed approach on both identification and computational sides.

Keywords: Online system identification, LPV models, LS-SVM methods.

1. INTRODUCTION

Linear Time Invariant (LTI) models are widely employed in many control applications due to their analytical simplicity and mature identification approaches (Ljung, 1998; Overschee and De Moor, 1996). Nonetheless, there are many situations where the underlying system is not linear. In this context, Linear Parameter Varying (LPV) models are a viable alternative, where the model parameters are not constants but functions of a *scheduling variable* that varies over time (Tóth, 2010). LPV models can represent complex systems while maintaining the interpretability and the intuitive structure proper of linear models (Cavanini et al., 2018; Atoui et al., 2022; Norouzi et al., 2022).

One way to describe a Single Input Single Output (SISO) LPV model is through the LPV-ARX formulation (Piga and Tóth, 2013)

$$y(k) + \sum_{i=1}^{n_a} a_i(\boldsymbol{\rho}(k))y(k-i) = \sum_{j=0}^{n_b} b_j(\boldsymbol{\rho}(k))u(k-j) + e(k) \tag{1}$$

where $k \in \mathbb{N}$ is the discrete time variable, $u : \mathbb{N} \to \mathbb{R}$ is the input signal, $y : \mathbb{N} \to \mathbb{R}$ is the output signal, $\boldsymbol{\rho} : \mathbb{N} \to \mathbb{P} \subseteq \mathbb{R}^{n_\rho}$, $n_\rho \in \mathbb{N}_0$, is the known scheduling variable, $e(k)$ is a white noise and $a_i, b_j : \mathbb{P} \to \mathbb{R}$ are the model coefficients functions that depend only on the current value of $\boldsymbol{\rho}(k)$.

The Least Squares Support Vector Machines (LS-SVM) method provides an efficient way to estimate LPV-ARX models (Tóth et al., 2011; Suykens, 2000). LS-SVM solve a parametric regression problem in a least-squares fashion, so that its parameters can be estimated in closed form thus not requiring iterative optimization algorithms. However, the resolution of the least-squares problem requires the inversion of an $N \times N$ matrix, with N the number of identification data. This computational complexity may pose a problem in online applications when a model must be updated every time a new datum is collected.

To overcome this issue, (Cavanini et al., 2020) presented an efficient update algorithm for recursive identification of LPV-ARX models leveraging the LS-SVM approach. The idea is to update the model only when the new collected data are considered sufficiently informative to aid the model identification, while non-informative data are discarded. The notion of what an informative datum represents is quantified by means of a *similarity function*, that evaluates the degree of similarity of a new data point with respect to all points in the data set currently available for identification. If the new data point is considered sufficiently informative by the similarity function, a recursive

10.1016/j.ifacol.2024.08.576

procedure was proposed to avoid the matrix inversion. Practical scenarios include, for instance, fault diagnosis applications where the system behaviour changes with respect to an operating conditions (Mazzoleni et al., 2017).

While the approach proposed in (Cavanini et al., 2020) is a significant improvement towards a more efficient online estimation of LPV-ARX models with LS-SVM, it still necessitates the definition of which similarity function to employ, with related hyperparameters that might not present specific rules for their tuning. Moreover, the model updating rule proposed in (Cavanini et al., 2020) can, in certain conditions, never trigger the inclusion of new data points so that the update of model is stopped.

To overcome these limitations, a new online approach based on information theory (Shannon, 1948) is proposed, that we denote as the *information-based approach*. Information theory defines the information of an event that occurs with a certain probability. These concepts are employed to quantify the information brought by a new data point collected online. Since the data space is continuous, it is impossible to compute the probability (and so the information) that new collected data assume a certain value. The problem is solved dividing the data space in folds, and computing the probability that the each new datum falls inside a specific fold, using the histogram method (Scott, 1979) . Then, the information of each new data point is compared to the total information of the data set, to decide if the new data point should be included or not for model updating. As a second minor contribution, this work also presents a modification on the LPV-ARX structure in (1) to improve the online updating performance of the model when input/output data have a time-varying mean value.

This work is organized as follows. Section 2 reviews the LPV-ARX identification problem with LS-SVM and the efficient online update using the similarity function approach of (Cavanini et al., 2020). Section 3 presents the new information-based approach along with a motivational example. Section 4 compares the performance of the information-based approach with respect to the similarity function approach, both on a real and simulated data. Section 5 provides conclusions and future directions.

2. RELATED WORKS

This section reviews the batch and online identification of a SISO LPV-ARX model using LS-SVM (Tóth et al., 2011; Piga and Tóth, 2013; Cavanini et al., 2020).

2.1 LPV-ARX identification with LS-SVM

A parameterized description of (1) is given by

$$y(k) = \sum_{i=1}^{n_g} \boldsymbol{w}_i^\top \boldsymbol{\phi}_i(k)\varphi_i(k) + e(k), \tag{2}$$

where $n_g = n_a + n_b + 1$, $\boldsymbol{\phi}_i : \mathbb{R} \to \mathbb{R}^{n_\phi}, n_\phi \in (0, +\infty)$, represents an unknown *feature map*, $\boldsymbol{w}_i \in \mathbb{R}^{n_\phi \times 1}$, $i \in \{1, ..., n_g\}$, are parameters vectors and

$$\varphi_i(k) := y(k-i), \quad i = 1, ..., n_a, \tag{3a}$$

$$\varphi_\iota(k) := u(k-j), \quad j = 0, ..., n_b, \quad \iota = j + n_a + 1, \tag{3b}$$

such that

$$a_i(\boldsymbol{\rho}(k)) = \boldsymbol{w}_i^\top \boldsymbol{\phi}_i(k), \tag{4a}$$

$$b_j(\boldsymbol{\rho}(k)) = \boldsymbol{w}_\iota^\top \boldsymbol{\phi}_\iota(k). \tag{4b}$$

The model identification using LS-SVM approach consists in solving the following optimization problem

$$\min_{\boldsymbol{w},\boldsymbol{e}} \mathcal{J}(\boldsymbol{w}, \boldsymbol{e}) := \frac{1}{2}\sum_{i=1}^{n_g} \boldsymbol{w}_i^\top \boldsymbol{w}_i + \frac{\gamma}{2}\sum_{k=1}^{N} e(k)^2, \tag{5a}$$

$$\text{s.t.} \quad e(k) = y(k) - \sum_{i=1}^{n_g} \boldsymbol{w}_i^\top \boldsymbol{\phi}_i(k)\varphi_i(k), \tag{5b}$$

where N is the number of data used for the identification and $\gamma \in \mathbb{R}_{>0}$ is a regularization parameter. Problem (5) can be solved with the *Lagrange multiplier method* (Boyd and Vandenberghe (2004)).The Lagrangian function is

$$\mathcal{L}(\boldsymbol{w}, \boldsymbol{e}, \boldsymbol{\alpha}) = \mathcal{J}(\boldsymbol{w}, \boldsymbol{e}) - \sum_{k=1}^{N} \alpha_k \left(\sum_{i=1}^{n_g} \boldsymbol{w}_i^\top \boldsymbol{\phi}_i(k)\varphi_i(k) + e(k) - y(k) \right) \tag{6}$$

with $\alpha_k \in \mathbb{R}$, $k \in \{1, ..., N\}$ being the Lagrangian multipliers. The optimal solution of (5) is obtained when the following conditions are satisfied

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_i} = 0 \to \boldsymbol{w}_i = \sum_{k=1}^{N} \boldsymbol{\phi}_i(k)\varphi_i(k), \tag{7a}$$

$$\frac{\partial \mathcal{L}}{\partial e(k)} = 0 \to \alpha_k = \gamma e(k), \tag{7b}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 \to e(k) = y(k) - \sum_{i=1}^{n_g} \boldsymbol{w}_i^\top \boldsymbol{\phi}_i(k)\varphi_i(k). \tag{7c}$$

Then, from (7a) it follows that

$$y(k) = \sum_{i=1}^{n_g} \left(\sum_{k=1}^{N} \alpha_k \varphi_i(k) \boldsymbol{\phi}_i^\top(k) \right) \boldsymbol{\phi}_i(k)\varphi_i(k) + \gamma^{-1}\alpha_k. \tag{8}$$

Defining the matrix $\boldsymbol{\Omega}$ such that is (j, k)-th element is

$$\Omega(j, k) := \sum_{i=1}^{n_g} \varphi_i(j)\boldsymbol{\phi}_i^\top(j)\boldsymbol{\phi}_i(k)\varphi_i(k) \quad j, k = 1, ..., N \tag{9}$$

and the vectors $\boldsymbol{Y} := [y(1), ..., y(N)]^\top$, $\boldsymbol{\alpha} := [\alpha_1, ..., \alpha_N]^\top$, (8) can be rewritten as

$$\boldsymbol{Y} = (\boldsymbol{\Omega} + \gamma^{-1}\boldsymbol{I}_N) \cdot \boldsymbol{\alpha} = \widetilde{\boldsymbol{\Omega}} \cdot \boldsymbol{\alpha}, \tag{10}$$

with $\boldsymbol{I}_N$ being the N dimensional identity matrix. Although each $\boldsymbol{\phi}_i$ is unknown, the matrix $\boldsymbol{\Omega}$ can be computed using the so called *kernel trick* (Gedam and Shikalpure (2017)) such that (9) can be rewritten as

$$K_i(\boldsymbol{\rho}(j), \boldsymbol{\rho}(k)) := \boldsymbol{\phi}_i^\top(j)\boldsymbol{\phi}_i(k), \quad i \in \{1, \ldots, n_g\} \tag{11a}$$

$$\Omega(j, k) = \sum_{i=1}^{n_g} \varphi_i(j) K_i(\boldsymbol{\rho}(j), \boldsymbol{\rho}(k))\varphi_i(k), \tag{11b}$$

where (11a) are positive definite kernel functions. In this work, the radial basis function (RBF) kernel is considered for each $i \in \{1, \ldots, n_g\}$ in (11a) so that

$$K_i(\boldsymbol{\rho}(j), \boldsymbol{\rho}(k)) = \exp\left(-\frac{\|\boldsymbol{\rho}(j) - \boldsymbol{\rho}(k)\|_2^2}{\sigma^2} \right), \tag{12}$$

with $\|\cdot\|_2$ being the l_2 norm. Then, $\boldsymbol{\alpha}$ in (10) can be computed as

$$\boldsymbol{\alpha} = \widetilde{\boldsymbol{\Omega}}^{-1}\boldsymbol{Y}, \tag{13}$$

and the LPV-ARX coefficients are computed from (4), (7a) and (11b) as

$$\hat{a}_i(\cdot) = \sum_{k=1}^{N} \alpha_k \varphi_i(k) K_i(\boldsymbol{\rho}(k), \cdot), \tag{14a}$$

$$\hat{b}_j(\cdot) = \sum_{k=1}^{N} \alpha_k \varphi_\iota(k) K_\iota(\boldsymbol{\rho}(k), \cdot). \tag{14b}$$

2.2 Similarity function online update rule

An efficient method to perform the online update of an LS-SVM LPV-ARX is proposed in (Cavanini et al., 2020), where new data are used for model updating only if they are sufficiently informative. The idea is to evaluate the novelty brought by a new datum with respect to all the data present in the identification data set. This is done using a similarity function $P : \mathbb{R}^{n_g} \to \mathbb{R}$ defined as

$$P(\boldsymbol{\varphi}(z)) \coloneqq \frac{1}{N}\sum_{k=1}^{N} \exp\left(-\frac{\|\boldsymbol{\varphi}(z) - \boldsymbol{\varphi}(k)\|_2^2}{\sigma^2}\right), \tag{15}$$

where $\boldsymbol{\varphi}(k) = [\varphi_1(k), ..., \varphi_{n_g}(k)]^\top \in \mathbb{R}^{n_g \times 1}$, $\sigma^2 \in \mathbb{R}_{>0}$ is an hyperparameter that should be tuned for each application, and z defines the time instant when the new datum arrives, so that $z > k$. The model is updated with the new data point $\boldsymbol{\varphi}(z)$ if

$$P(\boldsymbol{\varphi}(z)) \leq \underbrace{\sum_{k=1}^{N} \frac{P(\boldsymbol{\varphi}(k))}{N}}_{\mu} - 3\sqrt{\sum_{k=1}^{N} \frac{(P(\boldsymbol{\varphi}(k)) - \mu)^2}{N-1}}. \tag{16}$$

The update of the model is performed by augmenting the matrix $\widetilde{\boldsymbol{\Omega}}$ in (10) such that

$$\widetilde{\boldsymbol{\Omega}}_+ \coloneqq \begin{bmatrix} \widetilde{\boldsymbol{\Omega}} & \boldsymbol{u} \\ \boldsymbol{u}^\top & a \end{bmatrix}, \tag{17}$$

where $\boldsymbol{u} \coloneqq [u(1) \ldots u(N)]^\top \in \mathbb{R}^{N \times 1}$ and

$$u(k) = \sum_{i=1}^{n_g} \varphi_i(k) K_i(\boldsymbol{\rho}(k), \boldsymbol{\rho}(z)) \varphi_i(z),\ k \in \{1, \ldots, N\},$$

$$a = \sum_{i=1}^{n_g} \varphi_i(z) K_i(\boldsymbol{\rho}(z), \boldsymbol{\rho}(z)) \varphi_i(z) + \gamma^{-1}.$$

The inverse of (17) can be computed efficiently using the bordering method (Householder (1953))

$$\widetilde{\boldsymbol{\Omega}}_+^{-1} = \begin{bmatrix} \widetilde{\boldsymbol{\Omega}}^{-1} + \frac{\widetilde{\boldsymbol{\Omega}}^{-1}\boldsymbol{u}\boldsymbol{u}^\top\widetilde{\boldsymbol{\Omega}}^{-1}}{q} & -\frac{\widetilde{\boldsymbol{\Omega}}^{-1}\boldsymbol{u}}{q} \\ -\frac{\boldsymbol{u}^\top\widetilde{\boldsymbol{\Omega}}^{-1}}{q} & \frac{1}{q} \end{bmatrix}, \tag{18}$$

with $q = a - \boldsymbol{u}^\top\widetilde{\boldsymbol{\Omega}}^{-1}\boldsymbol{u}$. To keep the dimension of the data set constant, when a new datum is considered for model updating, the oldest one is discarded using a First In First Out (FIFO) rule, i.e.

$$\boldsymbol{\varphi}_i = [\boldsymbol{\varphi}_i(2:N); \varphi_i(z)], \quad \boldsymbol{Y} = [\boldsymbol{Y}(2:N); y(z)], \tag{19a}$$

$$\boldsymbol{\rho} = [\boldsymbol{\rho}(2:N); \boldsymbol{\rho}(z)], \tag{19b}$$

and the matrix $\widetilde{\boldsymbol{\Omega}}_+^{-1}$ is reduced accordingly such that

$$\widetilde{\Omega}^{-1}(i,j) = \widetilde{\Omega}_+(i+1, j+1) - \frac{\widetilde{\Omega}_+(i+1,1)\widetilde{\Omega}_+(1,j+1)}{\widetilde{\Omega}_+(1,1)}, \tag{20}$$

with $\widetilde{\Omega}^{-1}(i,j)$ the (i,j)-th element of $\widetilde{\boldsymbol{\Omega}}^{-1}$. In this way, the new value of $\boldsymbol{\alpha}$ in (13) can be computed with the new data set (19) and the new $\widetilde{\boldsymbol{\Omega}}^{-1}$ computed using (20).

The next section describes the proposed improvements for online identification of LPV-ARX models using LS-SVM.

3. PROPOSED METHODOLOGIES

This section presents two methodological improvements with respect to (Cavanini et al., 2020). The main contribution is an alternative update rule to (16). A second minor contribution includes the modeling of the possible time-varying mean of output data through a time-varying intercept term of the model.

3.1 Information-based online update rule

As discussed in the introduction, the method of (Cavanini et al., 2020) presented in Section 2.2 requires the user to employ a similarity function (15) that depends of an hyperparameter σ^2 which value has to be set based on the specific application. Moreover, the next example shows how the approach presented in Section 2.2, under some circumstances, might interrupt the model updating upon new data, thus resulting in bad identification performance. In this particular case, the problem can be mitigated by increasing the value of the hyperparameter σ^2 in (15). However, it is not certain if the new value of σ^2 in (15) would be a good value also for the kernel function in (12).

Example 1. Consider the noiseless LPV-ARX system

$$y(k) = (0.7 + \cos\rho(k))y(t-1) - (0.4 + \sin\rho(k))y(t-2) + (0.4 + \sin\rho(k))u(t-1), \tag{21}$$

with scheduling variable

$$\rho(k) = \beta\rho(k-1) + \frac{\cos(\rho(k-1))}{10}, \tag{22}$$

To measure the goodness of the model, define the Best Fit Ratio (BFR) as

$$\text{BFR} \coloneqq 100 \times \max\left(0, 1 - \frac{\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}\|_2}{\|\boldsymbol{Y} - \bar{y}\mathbf{1}_N\|_2}\right), \tag{23}$$

where $\hat{\boldsymbol{Y}}$ is the $N \times 1$ vector of outputs estimated by the model, $\bar{y} = mean(\boldsymbol{Y})$ with $mean(\cdot)$ the average function operating on the elements of the vector $\boldsymbol{Y}$, and $\mathbf{1}_N$ a $N \times 1$ vector of all ones. To identify the model, $N = 1000$ data points are simulated with $\beta = 0.8$ and the input signal in Figure 1. The kernel function and similarity function are chosen as the RBF in (12) with $\sigma^2 = 100$, while model orders in (1) to $n_a = 2$, $n_b = 1$ (the correct orders), and $\gamma = 0.1$ in (5). The BFR on identification data is 98%. Then, 15000 data points are simulated to test the online update algorithm of Section 2.2, using the input signal in Figure 1 and

$$\beta = \begin{cases} 0.8 & k \leq 3750 \\ 0.75 & 3750 < k \leq 7500 \\ 0.7 & k > 7500 \end{cases} \tag{24}$$

so that at the time stamp $k = 3750$ the scheduling variable (22) changes behaviour, while at time stamp $k = 7500$ both the input and the scheduling variable change behaviour. However, as shown in the bottom plot of Figure 2, the algorithm based on (16) stops updating the model at the time stamp $k = 4506$, as the right side of (16) becomes negative while the left side of (16) is always positive. Thus, the online identification scheme of Section 2.2 loses the capability to update the model, and this results in a loss of performance (BFR $= 84\%$) as the system changes its behaviour. The figure also shows the results of the proposed update rule scheme based on information theory, and it will be discussed next.

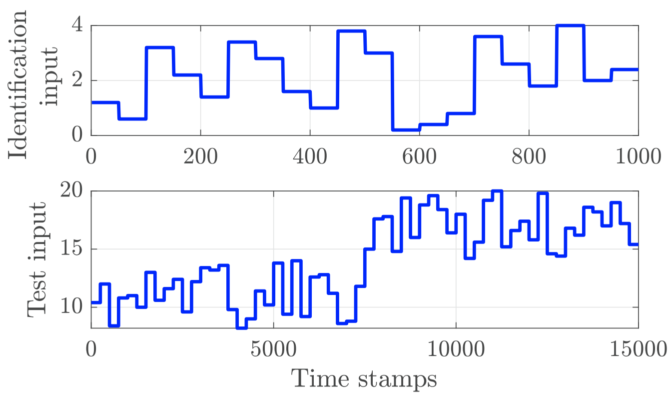

Fig. 1. Identification and test input signals of Example 1.

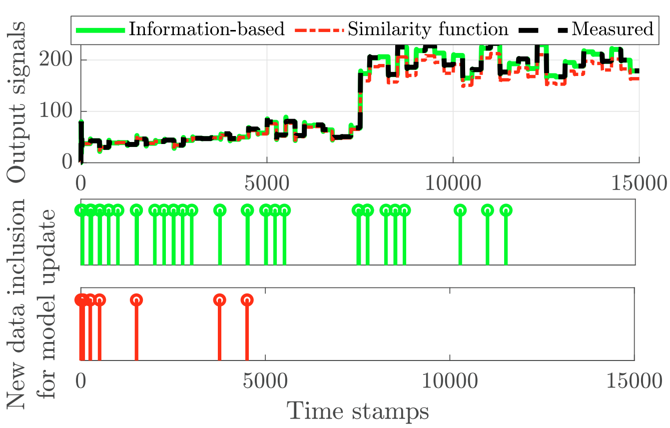

Fig. 2. Comparison between the similarity based approach (16) and the information-based one (40) on Example 1. Top: Estimated output of the online identified models. (Red) Updated rule (16); (Green) Proposed update rule (40); (Black) real output. Bottom: time stamps when the update rules (40) and (16) choose to update the model, respectively.

To overcome these limitations, a new information-based updating rule is proposed in place of (16).

Define $X = \{x_1, x_2, ..., x_M\}$ as a set of *events* and $p(x_s)$, $x_s \in X$ the probability mass function (pmf) associated to X. The *information* brought by the occurrence of the event x_s is (Shannon, 1948)

$$I(x_s) = \log \frac{1}{p(x_s)}. \tag{25}$$

In this context, X is called the *source*. The average information brought by a source is called the *entropy*

$$H(X) = \sum_{s=1}^{M} p(x_s) I(x_s), \tag{26}$$

while the *information variance* reads as

$$V(X) = \sum_{s=1}^{M} p(x_s)[I(x_s) - H(X)]^2. \tag{27}$$

The rationale of the proposed information-based approach, as a building block for a new update rule, is to divide the data samples into bins, and compute the probability that a data point falls into that bin. This pmf can be used to compute the information brought by a new datum collected online, and, based on this information, select or not the datum for model updating.

This pmf can be estimated by the histogram method (Scott, 1979). Defining $f(x)$, $x \in \mathbb{R}$, a probability density function (pdf), the histogram method is an efficient way to estimate $f(x)$ from a data sample as

$$\hat{f}(x; n_s, h_s, r_s) = \frac{n_s}{N h_s}, \tag{28}$$

where r_s and h_s are, respectively, the *starting point* and the *bin width* of the histogram bar such that $x \in \mathbb{S}_{r_s,h_s}$ with $\mathbb{S}_{r_s,h_s} = [r_s, r_s + h_s)$, and n_s is the number of data points that fall in $\mathbb{S}_{r_s,h_s}$. Despite the method is designed to estimate a pdf, in this work it is used to obtain a pmf representing the probability that $x \in \mathbb{S}_{r_s,h_s}$, as

$$p(s) = \int_{r_s}^{r_s+h_s} \hat{f}(x; n_s, h_s, r_s)\, dx = \frac{n_s}{N}. \tag{29}$$

In this way, a new collected data point will be associated with the event of falling in a certain bin of the histogram, and its information will be computed using the estimated pmf in (29). The choice of each h_s is crucial to estimate a precise histogram. In (Scott, 1979) it is proposed to use *equally spaced histograms* (ESH), i.e. $h_s = h \quad \forall s$. To measure the error made by the histogram approximating the real pdf (28), the *mean squared error* in a single point x is defined as

$$\text{MSE}(x) = \mathbb{E}[(f(x) - \hat{f}(x))^2]. \tag{30}$$

A global measure of the error of the histogram approximating the pdf (28) is given by the *integrated mean squared error*

$$\text{IMSE} = \int_{-\infty}^{+\infty} \text{MSE}(x)\, dx = \int_{-\infty}^{+\infty} \mathbb{E}[(f(x) - \hat{f}(x))^2]\, dx. \tag{31}$$

It can be shown in (Scott, 1979) that the IMSE for an ESH can be defined as

$$\text{IMSE}(h) = \frac{1}{nh} + \frac{1}{12} h^2 \mathcal{R}\left(\frac{\partial f}{\partial x}\right) + O\left(\frac{1}{n} + h^3\right), \tag{32}$$

where

$$\mathcal{R}\left(\frac{\partial f}{\partial x}\right) = \int_{-\infty}^{+\infty} \left(\frac{\partial f}{\partial x}\right)^2 dx. \tag{33}$$

As $\dfrac{\partial f}{\partial x}$ is almost always unknown, (Scott, 1979) suggests to use the *normal reference rule*, i.e. assuming that $f(x)$ is Gaussian (even if it is not). This results in

$$\mathcal{R}\left(\frac{\partial f}{\partial x}\right) = \frac{1}{4\sqrt{\pi}\lambda^3}, \tag{34}$$

where λ is the standard deviation of the data. The optimal bin width h^* is computed by minimizing (32) and considering (34), leading to

$$h^* = \left(\frac{24\sqrt{\pi}\lambda^3}{n}\right)^{\frac{1}{3}} \simeq 3.49\lambda n^{-\frac{1}{3}}. \tag{35}$$

Note that (32) does not depend on any r_s but only on h. This means that the *starting point* r_0 of the histogram can be decided arbitrarily, and then the start of each bin can be obtained as $r_s = r_0 + s \cdot h^*$. The pmf can be now computed using (29).

The idea of the proposed information-based approach is to use the histogram method to estimate a total of n_g probability mass functions (pfms) $p_i(s)$, $i = \{1, ..., n_g\}$, using as data sample the features vector

$$\boldsymbol{\varphi}_i = [\varphi_i(1), ..., \varphi_i(N)]^\top, \tag{36}$$

defined based on the currently available identification data set. Each $\boldsymbol{\varphi}_i$ in (36) is used to compute the respective h_i using (35) and $r_0^i = \min(\boldsymbol{\varphi}_i) - h_i/2$. Then, using (26) and (27), the entropy and the information variance of each $\boldsymbol{\varphi}_i$ can be computed as

$$H_i = \sum_{s=0}^{M^i-1} p_i(s) \log \frac{1}{p_i(s)}, \tag{37a}$$

$$V_i = \sum_{s=0}^{M^i-1} p_i(s) \left(\log \frac{1}{p_i(s)} - H_i\right)^2, \tag{37b}$$

with $M^i = \lceil (\max(\boldsymbol{\varphi}_i) - r_0^i)/h_i \rceil$ being the number of bins that contain all data points starting from r_0^i. Note that, when $p_i(s) = 0$, the expression inside the $\log(\cdot)$ does not make sense, but since it is multiplied by 0, then we assume in the implementation that the whole product is null.

The information brought by the i-th feature $\varphi_i(z)$ of new data point $\boldsymbol{\varphi}(z)$ is defined as

$$I_i = \log \frac{1}{p_i(s_{z,i})}, \tag{38}$$

where $s_{z,i}$ is defined such that

$$\varphi_i(z) \ \in [r^i_{s_{z,i}}, r^i_{s_{z,i}} + h). \tag{39}$$

Then, the new data point $\boldsymbol{\varphi}(z)$ is employed to updated the model if

$$\exists i = 1, ..., n_g \quad s.t. \quad I_i > H_i + 3\sqrt{V_i}, \tag{40}$$

i.e. if there is at least a feature in the new data point that brings more information than the average information in the data set, plus three times its standard deviation. If any $p_i(s_{z,i}) = 0$, i.e. if the data set does not contain any data point similar to the new one, then I_i is assumed to be infinite, since it can be useful to update the model with a datum that was never seen until that moment. Once the data point is selected to be included in the model according to (40), the update procedure is the same as in (17)-(20).

The proposed method solves the two main problems of the similarity function algorithm, since there are no functions or parameter to tune (as the bin width of the histogram is automatically obtained with (35)), and there are no risks that the algorithm stops updating (as unseen data have infinite information and so they are always included).

3.2 Time varying intercept

Model (1) assumes that output data have a constant mean (usually zero). However, in practical applications, the online identification of the model must deal also with a time-varying average behaviour of the input/output signals. To deal with these situations, the LPV-ARX formulation in (1) is extended by adding a time-varying intercept term $m(\boldsymbol{\rho}(k))$ as

$$y(k) + \sum_{i=1}^{n_a} a_i(\boldsymbol{\rho}(k))y(k-i) = \sum_{j=0}^{n_b} b_j(\boldsymbol{\rho}(k))u(k-j) + m(\boldsymbol{\rho}(k)) + e(k), \tag{41}$$

Considering (2), n_g is redefined such that $n_g = n_a + n_b + 2$, i.e. a new parameter is added to the model. Then, (3) is extended such that

$$\varphi_{n_g}(k) = 1, \tag{42}$$

and in a similar way (4) is extended such that

$$m(\boldsymbol{\rho}(k)) = \boldsymbol{w}_{n_g}^\top \boldsymbol{\phi}_{n_g}(k). \tag{43}$$

At each time k, the new coefficient can be estimated as

$$\hat{m}(\cdot) = \sum_{k=1}^{N} \alpha_k K_{n_g}(\boldsymbol{\rho}(k), \cdot). \tag{44}$$

The new coefficient behaves like an intercept that varies with time, and helps to capture the variation of the mean of the data during the online identification of the model.

4. EXPERIMENTAL RESULTS

The presented methodologies are tested on the data collected from a Spark-Ignite (SI) engine, typically used in aircraft applications (Cavanini et al., 2020). The input signal u is the throttle angle, while the output y is the engine speed. The nonlinear mathematical model is present in (Weeks and Moskwa, 1995). The scheduling variable ρ is set as $\rho(k) = y(k-1)$. Two different data sets are provided: one is used for model identification while the second for evaluation. Identification and test input signals are reported in Figure 3. The sampling time is $T_s = 0.015s$. The RBF in (12) is chosen as kernel function and the selected values for the hyperparameters are $\sigma = 20$, $\gamma = 0.1$, $n_a = 2$ and $n_b = 1$ as in (Cavanini et al., 2020). Note that the system is set to be strictly proper so that $y(k)$ does not depend on $u(k)$ but only on $u(k-1)$. The model (1) is compared to the one with the intercept term in (41), without and with the online update rule based on the similarity function (16) and the proposed information-based one (40). Figure 4 and Table 1 show the test performance, and, only for the online identified models, the computational time and the number of data points used to update the model. It can be seen that the model (41) has higher performances than the classical one in (1). The performances of (41) are the same to (1) if an update algorithm is employed. However, the computational time is way smaller with the proposed update rule (40). This is due to the fact that the update condition (16) has $\mathcal{O}(N^2)$ as its computational cost, while the computation of (40) is only $\mathcal{O}(N)$, as the highest computation is done for (29) that counts the number of data points that falls in each bin of the histogram.

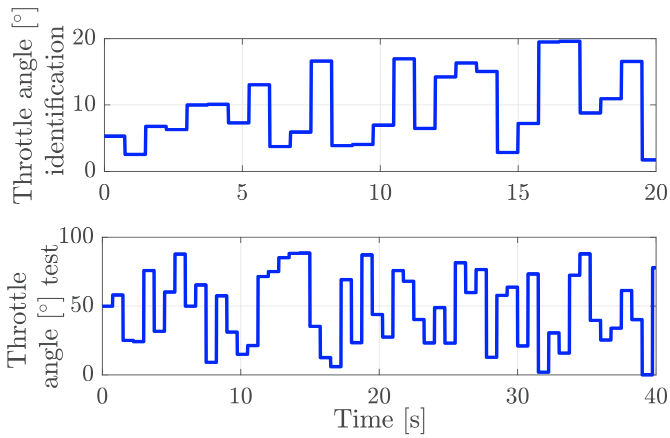

Fig. 3. Aircraft data set identification and test inputs.

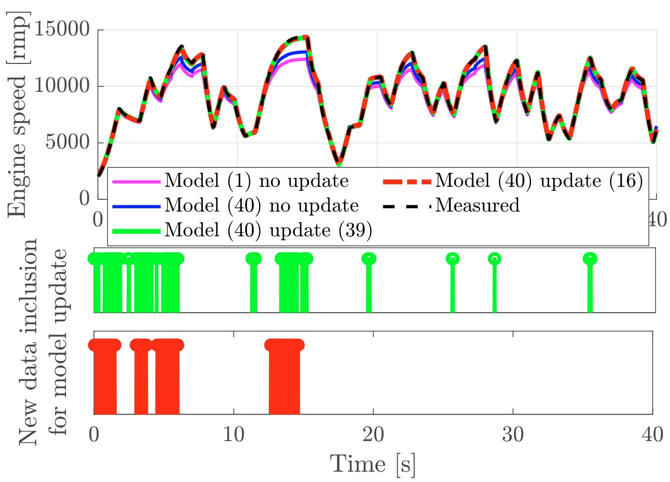

Fig. 4. Models comparison on the aircraft evaluation data set. Top: Estimated output. (Magenta) model (1) without online update, (Blue) model (41) without online update, (Red) model (41) without online update based on (16),(Green) model (41) without online update based on (40), (Black) measured output. Bottom: time stamps when update rules (40) and (16) choose to update the model, respectively.

Model	BFR	Time	Data
Model (1) no update	70.73%		
Model (41) no update	81.34%		
Model (41) online updated (16)	96.75%	76s	334
Model (41) online updated (40)	**97.05%**	**7s**	319

Table 1. Test performance on the aircraft evaluation data set.

The online updating rules (16) and (40) are also compared on Example 1 in Section 3.1, by considering the model (1). The estimated outputs of the two approaches are presented in Figure 2. From Table 2, it can be seen that the proposed information-based approach (40) performs better in terms of BFR as it does not stop updating the model with respect to the similarity function approach (16).

Model	BFR	Time	Data
Model (1) online updated (16)	83.85%	19s	79
Model (1) online updated (40)	**99.25%**	**9s**	288

Table 2. Test performance on Example 1.

5. CONCLUSION

In this paper, a new rule based on information theory for the online update of LS-SVM LPV-ARX models has been introduced. The proposed approach has been tested on an experimental aircraft data set and on a simulated example. In contrast to the existing approach, the new algorithm is computationally more efficient and does not suffer of stall problems. Also, it does not require any tuning of hyperparameters or functions, so that it can be employed without any previous analysis. An improvement of the standard LPV-ARX model formulation is proposed to face a time-varying mean value during online model updating.

REFERENCES

Atoui, H., Sename, O., Milanés, V., and Martinez, J.J. (2022). Lpv-based autonomous vehicle lateral controllers: A comparative analysis. *IEEE Transactions on Intelligent Transportation Systems*, 23(8), 13570 – 13581.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Cavanini, L., Cimmini, G., and Ippoliti, G. (2018). Computationally efficient model predictive control for a class of linear parameter-varying systems. *IET Control Theory & Application*, (12), 1384–1392.

Cavanini, L., Ferracuti, F., Longhi, S., and Monteriù, A. (2020). Ls-svm for lpv-arx identification: Efficient online update by low-rank matrix approximation. *International Conference on Unmanned Aircraft Systems*.

Gedam, A.G. and Shikalpure, S.G. (2017). Direct kernel method for machine learning with support vector machine. In *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 1772–1775.

Householder, A.S. (1953). *Principles of numerical analysis*. McGraw-Hill.

Ljung, L. (1998). *System Identification: Theory for the User*. Pearson.

Mazzoleni, M., Maccarana, Y., Previdi, F., Pispola, G., Nardi, M., Perni, F., and Toro, S. (2017). Development of a reliable electro-mechanical actuator for primary control surfaces in small aircrafts. In *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 1142–1147. doi:10.1109/AIM.2017.8014172.

Norouzi, A., Shahpouri, S., Gordon, D., Winkler, A., Nuss, E., Abel, D., Andert, J., Shahbakhti, M., and Koch, C.R. (2022). Machine learning integrated with model predictive control for imitative optimal control of compression ignition engines. *IFAC-PapersOnLine*, 55(24), 19–26. 10th IFAC Symposium on Advances in Automotive Control AAC.

Overschee, P. and De Moor, B. (1996). *Subspace Identification for Linear Systems*. Springer New York, NY.

Piga, D. and Tóth, R. (2013). Lpv model order selection in an ls-svm setting. In *52nd IEEE Conference on Decision and Control*, 4128–4133.

Scott, D.W. (1979). On optimal and data-based histograms. *Biometrika*, 66(3), 605–610.

Shannon, C.E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27(3), 379–423, 623–656.

Suykens, J.A.K. (2000). *Least Squares Support Vector Machines*. World Scientific Pub Co Inc.

Tóth, R. (2010). *Modeling and Identification of Linear Parameter-Varying Systems*. Springer Berlin, Heidelberg.

Tóth, R., Laurain, V., Zheng, W.X., and Poolla, K. (2011). Model structure learning: A support vector machine approach for lpv linear-regression models. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 3192–3197.

Weeks, R.W. and Moskwa, J.J. (1995). Automotive engine modeling for real-time control using matlab/simulink. *SAE Transactions*, 104, 295–309.