Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 59-15 (2025) 43–48

# A quadratic entropy algorithm for efficient online identification of LPV-ARX models using LS-SVM

**F. Corrini** *, **M. Mazzoleni** *, **M. Scandella** *, **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:** Modeling non-linear systems has always been a challenge in the field of control engineering. Linear Parameter Varying (LPV) models can be a valid choice to model complex systems, since they have a simple linear structure, but time varying coefficients that captures the system dynamics according to a scheduling signal measured from the system. A common approach to identify a LPV system in an ARX form is the Least Squares Support Vector Machines (LS-SVM) method. However, due to its computational complexity, it is difficult to employ such algorithm in online applications, when a model must be identified each time a new datum is collected from the system. An efficient recursive update algorithm has been recently presented in the literature for such cases, where only the most informative data points are selected to update the model, thus generally reducing the required computational effort. However, in certain conditions such algorithm selects too many data points, still leading to an high computational time. In this work, a quadratic entropy based algorithm is proposed to overcome the limitations found in the literature, providing a better trade-off between identification accuracy and computational time.



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

## 1. INTRODUCTION

Modeling non-linear systems has always been a major challenge in the automation field. Industrial applications, such as control engineering (Cavanini et al., 2018) and fault diagnosis (Mazzoleni, 2024) require high model accuracy to perform their task. In this context, linear approximations of the non-linear system may not be enough to capture all the system dynamics and provide the desired performance (Previdi and Lovera, 2004). A valid alternative is represented by Linear Parameter Varying (LPV) models (Piga and Tóth, 2013). LPV models rely on the concept of gain scheduling, i.e. the idea that a non-linear system can be well approximated by multiple Linear Time Invariant (LTI) models which describe the behavior of the system in different operation points. LPV models preserve the simple linear structure of LTI systems, but their coefficients vary over time depending on a *scheduling variable*, a signal measured from the system which describes the current operating condition of the system. A common structure for LPV models is the Single Input Single Output AutoRegressive with eXhogenous inputs (SISO LPV-ARX) one, which represents an LPV model in an ARX fashion with coefficients that depend on the scheduling variable. Least Squares Support Vector Machines (LS-SVM) is an efficient method to identify LPV-ARX models (Tóth et al., 2011). The parameters are estimated in a closed form solving a least squares regression problem, so that no iterative optimization algorithm is required. However, the computational complexity of LS-SVM is cubic in the number of data $N$, since it involves the inversion of an $N \times N$ kernel matrix, where $N$ is the number of identification data. This represents a problem in applications that require the online identification of the system on hardware with limited computation capabilities. For instance, online applications are increasingly important in the industrial manufacturing sector, where one of the aim of the new Industry 5.0 paradigm is resilience, i.e. the possibility to adapt plants to the constantly varying production, economical and social conditions. So, it is important to investigate and propose efficient online identification algorithms for LPV models.

Many approaches are proposed in the literature to improve the efficiency of SVM (Bottou et al., 2007), based on Nystrom method (Williams and Seeger, 2000) or techniques that selects only the most informative data points to update the LPV model (Cavanini et al., 2020). Further, in the field of learning, approaches based on information theory have emerged Han et al. (2011). For example, Corrini et al. (2024) presents a rationale based on the concepts of information and entropy (Shannon, 1948) to select only the most informative data points to perform the online update of the LPV model, and discarding points that do not bring any model improvement. This allows


10.1016/j.ifacol.2025.10.055

to reduce the required computational power; however, the information-based rule of Corrini et al. (2024) considers *separately* the information brought by each regressor of the LS-SVM LPV-ARX model, updating the model even if the newly acquired datum does not bring enough information. This can represent a problem when the LPV-ARX model has a high number of regressors, as too many data points will trigger the update procedure without contributing significantly to the performance of the model, leading to a decrease in the efficiency of the identification algorithm.

In this work, a new recursive identification algorithm is proposed to select the most informative data points to update the LS-SVM LPV-ARX model. The proposed algorithm relies on the concept of *quadratic entropy* (Rényi, 1961). The quadratic entropy has similar meaning to Shannon's one, but it can be estimated through a kernel function that considers jointly all the regressors of the model. As a second contribution, an alternative to the recursive update procedure by Cavanini et al. (2020) is proposed. This new procedure updates the LPV-ARX model not every time a new datum is selected, but rather after an *entire batch* of data is processed. In this way, the update procedure is performed more sparingly, contributing to the reduction of the overall computational time.

This work is organized as follows. Section 2 provides a description of the LS-SVM LPV-ARX identification algorithm, the information-based rule proposed by Corrini et al. (2024) and the recursive update procedure presented by Cavanini et al. (2020). Section 3 provides the formulation of the new methodologies proposed in this work. Section 4 presents a comparison between the new methodologies and the one already proposed in the literature on a simulated example. Finally, Section 5 provides the conclusion.

## 2. PROBLEM STATEMENT AND BACKGROUND

### *2.1 LS-SVM offline identification for LPV-ARX models*

In this section, the LS-SVM identification algorithm for LPV-ARX is presented. Consider a SISO LPV-ARX model as defined by Tóth et al. (2011)

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

where $t \in \mathbb{N}$ 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, $e$ 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 $\boldsymbol{\rho}(t)$. A parameterized description of (1) is given by

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

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

$$\varphi_i(t) \coloneqq y(t-i), \qquad i \in \{1, ..., n_a\},$$
$$\varphi_\iota(t) \coloneqq u(t-j), \qquad j \in \{0, ..., n_b\}, \quad \iota = j + n_a + 1,$$

such that

$$a_i(\boldsymbol{\rho}(t)) = \boldsymbol{w}_i^\top \boldsymbol{\phi}_i(\boldsymbol{\rho}(t)), \quad (2a)$$
$$b_j(\boldsymbol{\rho}(t)) = \boldsymbol{w}_\iota^\top \boldsymbol{\phi}_\iota(\boldsymbol{\rho}(t)). \quad (2b)$$

The parameters of the model can be identified through the LS-SVM method solving the following optimization problem:

$$\min_{\boldsymbol{w},\boldsymbol{e}} \mathcal{J}(\boldsymbol{w}, \boldsymbol{e}) \coloneqq \frac{1}{2}\sum_{i=1}^{n_g} \boldsymbol{w}_i^\top \boldsymbol{w}_i + \frac{\gamma}{2}\sum_{k=1}^{N} e_k^2, \quad (3a)$$

$$\text{s.t.} \quad e_k = y_k - \sum_{i=1}^{n_g} \boldsymbol{w}_i^\top \boldsymbol{\phi}_i(\boldsymbol{\rho}_k)\varphi_{i,k}, \quad (3b)$$

where $\boldsymbol{\varphi}_i \coloneqq [\varphi_{i,1}, ..., \varphi_{i,N}]^\top$, $i \in \{1, \dots, N\}$ and $\boldsymbol{\rho} \coloneqq [\boldsymbol{\rho}_1, \dots, \boldsymbol{\rho}_N]^\top$ are the $N$ data points used for identification, $\boldsymbol{e} \coloneqq [e_1, \dots, e_N]^\top$ and $\boldsymbol{w} \coloneqq [\boldsymbol{w}_1, \dots, \boldsymbol{w}_{n_g}]^\top$ are the decision variables of the optimization problem and $\gamma \in \mathbb{R}_{>0}$ is a regularization parameter. Notice that in this paper, subscripts are used to denote the identification data set, while parentheses indicate signals measured from the system.

According to the Lagrange multiplier method (Boyd and Vandenberghe, 2004), the solution to (3) is obtained in closed form solving the following linear system:

$$\boldsymbol{Y} = (\boldsymbol{\Omega} + \gamma^{-1}\boldsymbol{I}_N) \cdot \boldsymbol{\alpha} = \widetilde{\boldsymbol{\Omega}} \cdot \boldsymbol{\alpha}, \quad (4)$$

where $\boldsymbol{Y} \coloneqq [y_1, ..., y_N]^\top$, $\boldsymbol{I}_N$ is the $N$ dimensional identity matrix, $\boldsymbol{\alpha} \coloneqq [\alpha_1, ..., \alpha_N]^\top$ the Lagrangian multipliers and the $(j,k)$-th element of the matrix $\boldsymbol{\Omega}$ is

$$\Omega_{j,k} \coloneqq \sum_{i=1}^{n_g} \varphi_{i,j}\boldsymbol{\phi}_i^\top(\boldsymbol{\rho}_j)\boldsymbol{\phi}_i(\boldsymbol{\rho}_k)\varphi_{i,k} \quad j,k \in \{1, ..., N\}.$$

Since each feature map $\boldsymbol{\phi}_i$ is unknown, the matrix $\boldsymbol{\Omega}$ is computed using the kernel trick so that

$$\Omega_{j,k} = \sum_{i=1}^{n_g} \varphi_{i,j} K_i\left(\boldsymbol{\rho}_j, \boldsymbol{\rho}_k\right) \varphi_{i,k} \quad (5a)$$

$$K_i(\boldsymbol{\rho}_j, \boldsymbol{\rho}_k) \coloneqq \boldsymbol{\phi}_i^\top(\boldsymbol{\rho}_j)\boldsymbol{\phi}_i(\boldsymbol{\rho}_k), \quad i \in \{1, \dots, n_g\}, \quad (5b)$$

where (5b) are positive definite kernel functions. In this work, the Radial Basis Function (RBF) kernel is considered for each $i \in \{1, \dots, n_g\}$ in (5b) so that

$$K_i\left(\boldsymbol{\rho}_j, \boldsymbol{\rho}_k\right) = \exp\left(-\frac{\|\boldsymbol{\rho}_j - \boldsymbol{\rho}_k\|_2^2}{\sigma^2}\right),$$

with $\|\cdot\|_2$ being the euclidean norm. Then, $\boldsymbol{\alpha}$ in (4) can be computed as

$$\boldsymbol{\alpha} = \widetilde{\boldsymbol{\Omega}}^{-1}\boldsymbol{Y}, \quad (6)$$

and the LPV-ARX coefficients in (2) are computed as

$$\hat{a}_i(\cdot) = \sum_{k=1}^{N} \alpha_k \varphi_{i,k} K_i(\boldsymbol{\rho}_k, \cdot),$$
$$\hat{b}_j(\cdot) = \sum_{k=1}^{N} \alpha_k \varphi_{\iota,k} K_\iota(\boldsymbol{\rho}_k, \cdot).$$

### *2.2 Information-based online identification algorithm*

This section reviews the information-based online identification algorithm presented by Corrini et al. (2024) and the recursive update procedure proposed by Cavanini et al. (2020).

Let $X \coloneqq \{x_1, x_2, ..., x_M\}$ be a set of discrete *events* and $p(x_s)$, $x_s \in X$ the probability mass function (pmf) associated to $X$. Then, the *information* brought by the occurrence of the event $x_s$ is

$$I(x_s) = \log \frac{1}{p(x_s)}. \tag{7}$$

The average of the information (7) is called the *entropy*

$$H(X) = \sum_{s=1}^{M} p(x_s) I(x_s), \tag{8}$$

while the *information variance* reads as

$$V(X) = \sum_{s=1}^{M} p(x_s)[I(x_s) - H(X)]^2. \tag{9}$$

The idea behind the information-based rule proposed by Corrini et al. (2024) is to estimate a pmf for each regressor $\boldsymbol{\varphi}_i$ and use them to compute the information brought by each regressor of a new collected datum. If at least a regressor brings more information than a certain threshold, then the new datum is used to update the model, otherwise it is discarded. Each pmf is estimated by dividing the data samples into bins (all with equal length) and computing the probability that a data point falls into a specific bin. This is done using the histogram method (Scott, 1979).

Consider a new data point $\boldsymbol{\varphi}(t) \coloneqq [\varphi_1(t), \dots, \varphi_{n_g}(t)]$ collected at the time $t$. For each regressor $\varphi_i$ of the new datum, an histogram can be estimated by computing its constant bin width as in (Scott, 1979):

$$h_i^{(t)} \simeq 3.49 \lambda_i^{(t)} N^{-\frac{1}{3}}, \quad i \in \{1, \dots, n_g\},$$

where $\lambda_i^{(t)}$ is

$$\lambda_i^{(t)} = \sqrt{\frac{1}{N-1} \sum_{k=1}^{N} \left( \varphi_{i,k}^{(t-1)} - \frac{1}{N} \sum_{k=1}^{N} \varphi_{i,k}^{(t-1)} \right)^2},$$

where $\boldsymbol{\varphi}_i^{(t-1)} \coloneqq [\varphi_{i,1}^{(t-1)}, \dots, \varphi_{i,N}^{(t-1)}]$, $i \in \{1, \dots, n_g\}$ represents the online identification data set at the time $t-1$, i.e. the available data set before time $t$ with $N$ elements. Then, defining

$$r_{i,0}^{(t)} = \min\left(\boldsymbol{\varphi}_i^{(t-1)}\right) - \frac{h_i^{(t)}}{2},$$

each bin is obtained as the set $R_{i,s}^{(t)} \coloneqq [r_{i,s}^{(t)}, r_{i,s}^{(t)} + h_i^{(t)})$ where

$$r_{i,s}^{(t)} = r_{i,0}^{(t)} + s \cdot h_i^{(t)},$$

where $s \in \mathbb{Z}$ is the index of the $s$-th bin of the histogram. Finally, the estimated pmf of $\boldsymbol{\varphi}_i$ is obtained as

$$p_i^{(t)}(s) \coloneqq \frac{N_{i,s}^{(t)}}{N},$$

where $N_{i,s}^{(t)}$ is the number of points that fall in the bin $R_{i,s}^{(t)}$. Then, using (8) and (9), the entropy and the information variance of each $\boldsymbol{\varphi}_i^{(t)}$ can be computed as [1]

$$H_i^{(t)} = \sum_{s=0}^{M_i^{(t)}-1} p_i^{(t)}(s) \log \frac{1}{p_i^{(t)}(s)},$$

$$V_i^{(t)} = \sum_{s=0}^{M_i^{(t)}-1} p_i^{(t)}(s) \left( \log \frac{1}{p_i^{(t)}(s)} - H_i^{(t)} \right)^2,$$

with $M_i^{(t)} = \left\lceil (\max(\boldsymbol{\varphi}_i^{(t-1)}) - r_{i,0}^{(t)}) / h_i^{(t)} \right\rceil$ being the number of bins that contain all data points starting from $r_{i,0}^{(t)}$.

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

$$I_i^{(t)} = \log \frac{1}{p_i\left(s_i^{(t)}\right)},$$

where $s_i^{(t)}$ is defined such that

$$\varphi_i(t) \in R_{i,s}^{(t)}, \quad s = s_i^{(t)}.$$

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

$$\exists i \in \{1, ..., n_g\} \quad \text{s.t.} \quad I_i^{(t)} > H_i^{(t)} + 3\sqrt{V_i^{(t)}}, \tag{10}$$

i.e. if there is at least a regressor 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 $p_i^{(t)}(s_i^{(t)}) = 0$, i.e. if the data set does not contain any data point similar to the new one, then $I_i^{(t)}$ is assumed to be infinite, since it can be useful to update the model with a datum that is not similar to data measured until that moment. Once the data point is selected to be included in the model according to (10), the update of the model is performed by augmenting the matrix $\widetilde{\boldsymbol{\Omega}}$ in (4) such that

$$\widetilde{\boldsymbol{\Omega}}_+^{(t)} \coloneqq \begin{bmatrix} \widetilde{\boldsymbol{\Omega}}^{(t-1)} & \boldsymbol{u} \\ \boldsymbol{u}^\top & a \end{bmatrix}, \tag{11}$$

where $\boldsymbol{u} \coloneqq [u_1, \dots, u_N]^\top \in \mathbb{R}^{N \times 1}$ and

$$u_k = \sum_{i=1}^{n_g} \varphi_{i,k}^{(t-1)} K_i(\boldsymbol{\rho}_k^{(t-1)}, \boldsymbol{\rho}(t)) \varphi_i(t), \; k \in \{1, \dots, N\},$$

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

The inverse of (11) can be computed efficiently using the bordering method (Householder (1953))

$$\boldsymbol{\Theta}^{(t)} = \begin{bmatrix} \boldsymbol{P}^{(t-1)} + \dfrac{\boldsymbol{P}^{(t-1)} \boldsymbol{u} \boldsymbol{u}^\top \boldsymbol{P}^{(t-1)}}{q} & -\dfrac{\boldsymbol{P}^{(t-1)} \boldsymbol{u}}{q} \\ -\dfrac{\boldsymbol{u}^\top \boldsymbol{P}^{(t-1)}}{q} & \dfrac{1}{q} \end{bmatrix},$$

with $q \coloneqq a - \boldsymbol{u}^\top \boldsymbol{P}^{(t-1)} \boldsymbol{u}$, $\boldsymbol{P} \coloneqq \widetilde{\boldsymbol{\Omega}}^{-1}$ and $\boldsymbol{\Theta} \coloneqq \widetilde{\boldsymbol{\Omega}}_+^{-1}$.

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. To this end, define the function $\Delta_k : \mathbb{R}^{N+1} \to \mathbb{R}^N$ as

$$\Delta_L(\boldsymbol{x}, \tilde{x}) \coloneqq \begin{bmatrix} \boldsymbol{I}_{L-1} & \boldsymbol{0}_{L-1 \times N-L+1} \\ \boldsymbol{0}_{N-L \times L} & \boldsymbol{I}_{N-L} \\ \boldsymbol{0}_{1 \times L} & \boldsymbol{0}_{1 \times N-L} \end{bmatrix} \boldsymbol{x} + \begin{bmatrix} \boldsymbol{0}_{(N-1) \times 1} \\ 1 \end{bmatrix} \tilde{x},$$

[1] In the case $p_i^{(t)}(s) = 0$, the domain of the functions $p_i^{(t)}(s) \log \frac{1}{p_i^{(t)}(s)}$ and $p_i^{(t)}(s) \left( \log \frac{1}{p_i^{(t)}(s)} \right)^2$ is extended to include 0 to ensure their continuity, i.e. they map 0 to 0.

with $\boldsymbol{x} \in \mathbb{R}^N$, $\tilde{x} \in \mathbb{R}$, $\boldsymbol{I}_k$ the $k$ dimensional identity matrix and $\mathbf{0}_{j\times k}$ the $j \times k$ matrix with all entries equal to zero. Then, the data set is updated such that

$$\begin{aligned}\boldsymbol{\varphi}_i^{(t)} &= \Delta_1(\boldsymbol{\varphi}_i^{(t-1)}, \varphi_i(t)),\\ \boldsymbol{Y}^{(t)} &= \Delta_1(\boldsymbol{Y}^{(t-1)}, y(t)),\\ \rho_{j,k}^{(t)} &= \Delta_1(\rho_{j,k}^{(t-1)}, \rho_{j,k}(t)),\end{aligned}$$

where $\boldsymbol{\rho}_k(t) = \{\rho_{1,k}(t), \dots \rho_{n_\rho,k}(t)\}$. The matrix $\boldsymbol{\Theta}^{(t)}$ is reduced accordingly such that

$$P_{i,j}^{(t)} = \Theta_{i+1,j+1}^{(t)} - \frac{\Theta_{i+1,1}^{(t)}\Theta_{1,j+1}^{(t)}}{\Theta_{1,1}^{(t)}}, \quad i,j \in \{1,\dots,N\} \tag{12}$$

with $\Theta_{i,j}^{(t)}$ the $(i,j)$-th element of $\boldsymbol{\Theta}^{(t)}$. In this way, the new value of $\boldsymbol{\alpha}$ in (6) can be calculated with the new data set and the new $\widetilde{\boldsymbol{\Omega}}^{-1} = \boldsymbol{P}$ computed using (12).

## 3. CONTRIBUTIONS

### *3.1 Quadratic entropy online identification algorithm*

The information-based update algorithm selects a new datum when the condition (10) is triggered. In the case of high order models with many regressors, it can happen that too many data points are selected for the update. This is due to the fact that (10) can be triggered by just a single regressor that brings enough information, even if all the other ones are not useful for the model. These high number of updates increases the computational time while not improving significantly the model. In this context, considering all the regressors jointly might reduce the number of updates, and thus increase the computational performance. This is the purpose of the new *quadratic entropy online update algorithm*, which is based on the quadratic entropy instead of the classical Shannon's entropy.

The continuous quadratic entropy is defined as follows (Rényi, 1961):

$$H^2(X) = -\log\left(\int p(x_s)^2\right).$$

During the online identification process, the entropy of the data set can be approximated as showed by Girolami (2002) as

$$H^2(\boldsymbol{\varphi})^{(t)} \approx -\log\left(\frac{1}{N^2}\sum_{k=1}^{N}\sum_{j=1}^{N}\Gamma_{k,j}^{(t-1)}\right), \tag{13a}$$

$$\Gamma_{k,j}^{(t-1)} = K(\widetilde{\boldsymbol{\varphi}}_k^{(t-1)}, \widetilde{\boldsymbol{\varphi}}_j^{(t-1)}), \quad k,j \in \{1,\dots,N\} \tag{13b}$$

$$\widetilde{\boldsymbol{\varphi}}_k^{(t-1)} = [\varphi_{1,k}^{(t-1)}, \dots, \varphi_{n_g,k}^{(t-1)}], \quad k \in \{1,\dots,N\} \tag{13c}$$

with $K(\cdot,\cdot)$ a positive definite kernel function [2].

The idea of the proposed algorithm is to select data points that increase the quadratic entropy (13a), or equivalently decrease the sum of the elements in $\Gamma_{k,j}^{(t-1)}$. This is done computing the quantity $\eta_k$ that represents the contribution of the datum $\widetilde{\boldsymbol{\varphi}}_k^{(t-1)}$ to (13a):

$$\eta_k^{(t-1)} \coloneqq \sum_{j=1}^{N}\Gamma_{k,j}^{(t-1)}, \qquad k \in \{1,\dots,N\}.$$

When a new datum $\boldsymbol{\varphi}(t)$ is collected, the quantity $\mu^{(t)}$ is computed as follows:

$$\begin{aligned}\xi_k^{(t)} &\coloneqq K(\widetilde{\boldsymbol{\varphi}}_k^{(t-1)}, \boldsymbol{\varphi}(t)), \qquad k \in \{1,\dots,N\},\\ \mu^{(t)} &\coloneqq K(\boldsymbol{\varphi}(t), \boldsymbol{\varphi}(t)) + \sum_{k=1}^{N}\xi_k^{(t)}.\end{aligned}$$

The quantity $\zeta_k^{(t)} \coloneqq \mu^{(t)} - \xi_k^{(t)}$ represents the hypothetical contribution of the datum $\boldsymbol{\varphi}(t)$ when it is inserted in the data set while removing the data point $\widetilde{\boldsymbol{\varphi}}_k^{(t)}$.

Define the quantity

$$\delta_k^{(t)} \coloneqq \eta_k^{(t-1)} - \zeta_k^{(t)},$$

as the reduction of (13a) obtained substituting $\widetilde{\boldsymbol{\varphi}}_k^{(t-1)}$ with $\boldsymbol{\varphi}(t)$. Define also

$$\hat{k} \coloneqq \arg\max_k \delta_k^{(t)}, \tag{14}$$

as the index of the data in the data set such that substituting $\widetilde{\boldsymbol{\varphi}}_{\hat{k}}^{(t-1)}$ with $\boldsymbol{\varphi}(t)$ leads to the maximal reduction of (13a). In this way, the less informative datum is removed from the data set optimizing the reduction of (13a) with respect to using a FIFO procedure, as done in the information-based algorithm of Section 2.2.

Finally, the datum $\boldsymbol{\varphi}(t)$ is selected by the algorithm if the following condition is triggered:

$$\delta_{\hat{k}}^{(t)} > 0 \ \wedge \ \zeta_{\hat{k}}^{(t)} \leq q_\eta^{(t-1)}, \tag{15}$$

where $q_\eta^{(t-1)}$ is the first percentile of $\boldsymbol{\eta}^{(t-1)} = [\eta_1^{(t-1)}, \dots, \eta_N^{(t-1)}]$. In other words, (15) is triggered when selecting $\boldsymbol{\varphi}(t)$ leads to a decrease of (13a) and the contribution $\zeta_{\hat{k}}^{(t)}$ will be significantly lower than the contribution of the other points already in the data set. Every time a new datum is selected, the values in $\boldsymbol{\eta}^{(t)}$ can be recursively computed as follows:

$$\eta_k^{(t)} = \eta_k^{(t-1)} - K(\widetilde{\boldsymbol{\varphi}}_{\hat{k}}^{(t-1)}, \widetilde{\boldsymbol{\varphi}}_k^{(t-1)}) + \underbrace{K(\boldsymbol{\varphi}(t), \widetilde{\boldsymbol{\varphi}}_k^{(t-1)})}_{\xi_k^{(t)}}.$$

Once the new datum $\boldsymbol{\varphi}(t)$ is selected to be included in the model according to (15), the update procedure is the same as in Section 2.2, with the new data set that becomes:

$$\boldsymbol{\varphi}_i^{(t)} = \Delta_{\hat{k}}(\boldsymbol{\varphi}_i^{(t-1)}, \varphi_i(t)), \tag{16a}$$

$$\boldsymbol{Y}^{(t)} = \Delta_{\hat{k}}(\boldsymbol{Y}^{(t-1)}, y(t)), \tag{16b}$$

$$\boldsymbol{\rho}^{(t)} = \Delta_{\hat{k}}(\boldsymbol{\rho}^{(t-1)}, \boldsymbol{\rho}(t)), \tag{16c}$$

$$\boldsymbol{\eta}^{(t)} = \Delta_{\hat{k}}(\boldsymbol{\eta}^{(t)}, \zeta_{\hat{k}}^{(t)}), \tag{16d}$$

and (12) that has to be adjusted to remove the $\hat{k}$-th data point found by (14) instead of the first one.

The quadratic entropy online identification algorithm solves the problem of considering separately the features of the new data point by employing a kernel function. In this

[2] in this work, it coincides with the kernel function (5b) used for the LS-SVM identification algorithm.

way, only the most informative data points are selected and the required computational time is reduced.

### 3.2 Batch online update procedure

In this section, an improvement of the recursive update procedure in (11)-(12) is proposed that further reduces the required computational time. This is done updating the matrix $\boldsymbol{\Omega}$ only after an entire batch of data has been collected, and not every time a new datum triggers (15).

Consider a batch of data collected online $\boldsymbol{\varphi}_W(t) \coloneqq [\boldsymbol{\varphi}(t), \dots, \boldsymbol{\varphi}(t+W-1)]$ of dimension $W$. During the collection of each $\boldsymbol{\varphi}(t+k)$, $k \in \{0, \dots, W-1\}$, the quadratic entropy algorithm is used to update the data sets (16) according to the rule in (15). However, the matrix $\boldsymbol{P} = \boldsymbol{\Omega}^{-1}$ is not updated, having $\boldsymbol{P}^{(k)} = \boldsymbol{P}^{(k-1)}$ for $k \in \{t, \dots, t+W-1\}$. After processing the entire batch of data, define $m$ as the number of data points selected by the quadratic entropy algorithm from $\boldsymbol{\varphi}_W(t)$. Notice that $m$ is not the number of time that (15) is triggered, since some data points can be selected and then discarded during the evaluation of the same batch. If $m > 0$, i.e. if at least a data point is selected by the algorithm, the matrix $\boldsymbol{P}$ is updated with the $m$ selected data according to the following procedure. Define the matrix

$$\widetilde{\boldsymbol{\Omega}}_{+}^{(t_W)} = \begin{bmatrix} \widetilde{\boldsymbol{\Omega}}^{(t_W-1)} & \boldsymbol{U} \\ \boldsymbol{U}^\top & \boldsymbol{A} \end{bmatrix} = \begin{bmatrix} \widetilde{\boldsymbol{\Omega}}^{(t)} & \boldsymbol{U} \\ \boldsymbol{U}^\top & \boldsymbol{A} \end{bmatrix},$$

where $t_W \coloneqq t+W-1$, $\boldsymbol{U}$ is the $N \times m$ matrix and $\boldsymbol{A}$ the $m \times m$ matrix such that

$$\begin{aligned} U_{k,j} &= \sum_{i=1}^{n_g} \varphi_{i,k}^{(t)} K_i(\boldsymbol{\rho}_k^{(t)}, \boldsymbol{\rho}_j^{(t_W)}) \varphi_{i,j}^{(t_W)}, \\ &k \in \{1, \dots, N\}, \quad j \in \{N-m+1, \dots, N+m\}, \\ A_{k,j} &= \sum_{i=1}^{n_g} \varphi_{i,k}^{(t_W)} K_i(\boldsymbol{\rho}_j^{(t_W)}, \boldsymbol{\rho}_j^{(t_W)}) \varphi_{i,j}^{(t_W)}, \\ &k, j \in \{N-m+1, \dots, N+m\}, \quad k \neq j, \\ A_{k,k} &= \sum_{i=1}^{n_g} \varphi_{i,k}^{(t_W)} K_i(\boldsymbol{\rho}_k^{(t_W)}, \boldsymbol{\rho}_k^{(t_W)}) \varphi_{i,k}^{(t_W)} + \gamma^{-1}, \\ &k \in \{N-m+1, \dots, N+m\}. \end{aligned}$$

The matrix $\boldsymbol{\Theta}^{t_W}$ is computed using the "extended" bordering method (Wang et al., 2020):

$$\boldsymbol{\Theta}^{(t_W)} = \begin{bmatrix} \boldsymbol{P}^{(t)} + \boldsymbol{P}^{(t)}\boldsymbol{U}\boldsymbol{Q}\boldsymbol{U}^\top\boldsymbol{P}^{(t)} & -\boldsymbol{P}^{(t)}\boldsymbol{U}\boldsymbol{Q} \\ -\boldsymbol{Q}\boldsymbol{U}^\top\boldsymbol{P}^{(t)} & \boldsymbol{Q} \end{bmatrix},$$

where $\boldsymbol{Q} = \left(\boldsymbol{A} - \boldsymbol{U}^\top\boldsymbol{P}^{(t)}\boldsymbol{U}\right)^{-1}$. The matrix $\boldsymbol{\Theta}$ is reduced as following:

$$\boldsymbol{P}^{(t_W)} = \boldsymbol{\Theta}_{NN} - \boldsymbol{\Theta}_{Nm}\boldsymbol{\Theta}_{mm}^{-1}(\boldsymbol{\Theta}_{Nm})^\top$$

where $\boldsymbol{\Theta}_{NN}$ is obtained removing from $\boldsymbol{\Theta}^{(t_W)}$ the $m$ rows and columns related to the discarded data points, $\boldsymbol{\Theta}_{Nm}$ is obtained removing from $\boldsymbol{\Theta}^{(t_W)}$ the $m$ rows related to the discarded data points and $N$ columns related to the remaining data points, and $\boldsymbol{\Theta}_{mm}$ selecting from $\boldsymbol{\Theta}^{(t_W)}$ the $m$ rows and columns related to the discarded data points. The new matrix $\boldsymbol{P} = \widetilde{\boldsymbol{\Omega}}^{-1}$ is used to compute $\boldsymbol{\alpha}$ as in (6).

The batch online update procedure is computationally more efficient with respect to the recursive update procedure in (11)-(12) since, whatever the value of $m$ is, the number of subtractions required to update the matrix $\widetilde{\boldsymbol{\Omega}}^{-1}$ will always be $N^2$, while the naive algorithm requires $mN^2$ subtractions ($N^2$ for each new datum selected by (15)). Since $m \ll N$, the computational cost required to invert the matrix $\boldsymbol{\Theta}_{mm}$, which is $\mathcal{O}(m^3)$, can be ignored. However, this approach introduces a delay in the update of the model which can lead to a decrease of the performance. Because of this, there is a trade-off between the accuracy of the model and the computational time that depends on the decided value of the parameter $W$.

## 4. SIMULATION RESULTS

In this section, the performance of the methodologies proposed in Section 3 are evaluated on a simulated example and compared with the literature presented in Section 2.2. Consider the following deterministic LPV-ARX system:

$$y(t) = \boldsymbol{c}^\top(t)\bar{\boldsymbol{y}}(t) + \boldsymbol{d}^\top(t)\bar{\boldsymbol{u}}(t),$$

where

$$\begin{aligned} \bar{\boldsymbol{y}}(t) &\coloneqq [y(t-5), \dots, y(t-1)], \\ \bar{\boldsymbol{u}}(t) &\coloneqq [u(t-5), \dots, u(t-1)], \\ \boldsymbol{c}(t) &\coloneqq 0.05 \cdot [-0.05, 0.5, -0.4, -0.3, 0.3] + \\ &\quad + 0.02 \cdot [\sin(\tau), \cos(\tau), \sin(\tau), \cos(\tau), \sin(\tau)], \\ \boldsymbol{d}(t) &\coloneqq 0.05 \cdot [-0.04, 0.2, -0.1, -0.5, 0.8] + \\ &\quad + 0.02 \cdot [\cos(\tau), \sin(\tau), \cos(\tau), \sin(\tau), \cos(\tau)], \end{aligned}$$

with $\tau$ a value that depends on the time stamp $t$. A total of 15000 samples are simulated considering

$$\tau = \begin{cases} \frac{t}{1500}, & t < 7500, \\ \frac{t}{750}, & t \geq 7500. \end{cases}$$

The first 200 samples are discarded to remove transient effects in the data. Then, the sample is divided in two sets, one for the identification of the model (first 2300 data points) and one for validation (the remaining 12500 data points). Figure 1 shows the train and test input signal. It is possible to see that the range covered by the test input signal is wider with respect to the train input signal. Because of this, an online identification algorithm is necessary to have good performance in conditions of the system that are not covered by the train data set. To identify the model, the following values for the hyperparameters are selected: $\gamma = 100$, $\sigma = 20$, $n_a = n_b = 5$ (the correct orders). The scheduling variable $\rho$ is set to be $\rho(t) = y(t-1)$. For the batch update algorithms, the selected value for the batch size is $W = 20$. To measure the goodness of the model, define the Best Fit Ratio (BFR) as

$$\text{BFR} \coloneqq 100 \times \max\left(0, 1 - \frac{\|\boldsymbol{Y} - \hat{\boldsymbol{Y}}\|_2}{\|\boldsymbol{Y} - \bar{y}\boldsymbol{1}_N\|_2}\right), \tag{17}$$

where $\hat{\boldsymbol{Y}}$ is the $N \times 1$ vector of outputs estimated by the model, $\bar{y} = \text{mean}(\boldsymbol{Y})$ with $\text{mean}(\cdot)$ the average function operating on the elements of the vector $\boldsymbol{Y}$, and $\boldsymbol{1}_N$ a $N \times 1$ vector of all ones.

Figure 2 shows that all online algorithms performs very well, while not employing an online algorithm results in very poor identification performance. Table 1 summarizes

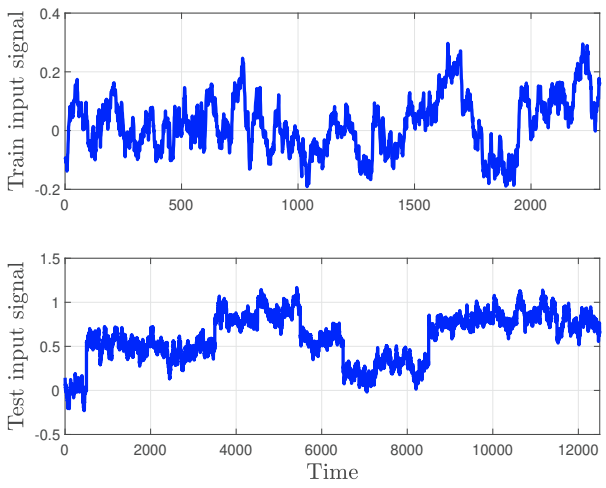


Fig. 1. Train (top) and test (bottom) input signal for the simulated example.

| Model | BFR | Time | Data |
|---|---|---|---|
| Section 2.1 algorithm | 0 | | |
| Section 2.2 algorithm | **95.70**% | 40.29 s | 1205 |
| Section 3.1 algorithm | 95.28% | 19.08 s | **381** |
| Section 3.1+3.2 algorithm, $W = 5$ | 95.22% | 16.57 s | **381** |
| Section 3.1+3.2 algorithm, $W = 20$ | 95.19% | **10.93** s | **381** |

Table 1. Performance on test data of the methodologies presented in Section 2 and 3.

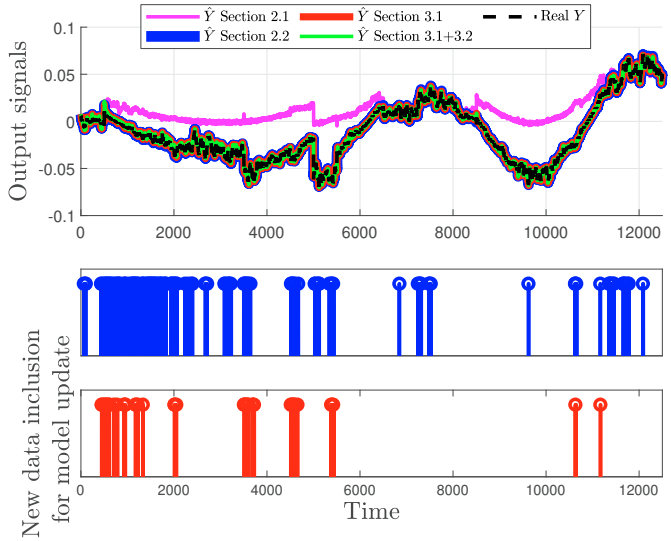


Fig. 2. Test output signal compared to model predictions (top), time stamp in which the models are updated (bottom).

the performance of each algorithm in terms of BFR (17), computational time and number of selected data points. Simulation results show that the three online identification algorithms have similar BFR, however the quadratic entropy update algorithm proposed in Section 3.1 has lower computational time since it selects only the most informative data points, while the algorithm presented in Section 2.2 more points without improving significantly the model. The new batch update procedure proposed in Section 3.2 reduces drastically the computational time. Identification performance decreases as it decreases $W$, but not significantly.

## 5. CONCLUSION

In this work, a new algorithm based on the concept of quadratic entropy is proposed to reduce the computational time required for recursive online identification of LS-SVM LPV-ARX models. As a second contribution, an improvement of the recursive update procedure presented by Cavanini et al. (2020) is provided. The proposed methodologies have been tested on a simulated example. Results show that the new algorithms reduce significantly the computational time with respect to what has already been proposed in literature.

## ACKNOWLEDGMENT

This work was partially funded by the EU European Defence Fund Project KOIOS (EDF-2021-DIGIT-R-FL-KOIOS). KOIOS is a project Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Defence Fund. Neither the European Union nor the granting authority can be held responsible for them.

## REFERENCES

Bottou, L., Chapelle, O., DeCoste, D., and Weston, J. (2007). *Large-Scale Kernel Machines*. MIT Press. doi:10.7551/mitpress/7496.001.0001.

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 Appl.*, (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. *Int. Conf. Unmanned Aircr. Syst.*

Corrini, F., Mazzoleni, M., Ferracuti, F., Cavanini, L., and Previdi, F. (2024). An information theory approach for recursive LPV-ARX model identification via LS-SVM. *20th IFAC Symp. Syst. Identif.*

Girolami, M. (2002). Orthogonal Series Density Estimation and the Kernel Eigenvalue Problem. *Neural Comput.*, 14(3), 669–688.

Han, M., Liang, Z., and Li, D. (2011). Sparse kernel density estimations and its application in variable selection based on quadratic renyi entropy. *Neurocomputing*, 74(10), 1664–1672.

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

Mazzoleni, M. (2024). Design of supervision solutions for industrial equipment: Schemes, tools and guidelines for the user. *J. Ind. Inf. Integr.*

Piga, D. and Tóth, R. (2013). LPV model order selection in an ls-svm setting. In *52nd IEEE Conf. Decis. Control*.

Previdi, F. and Lovera, M. (2004). Identification of non-linear parametrically varying models using separable least squares. *Int. J. Control*, 77(16), 1382–1392.

Rényi, A. (1961). On measures of entropy and information.

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

Shannon, C.E. (1948). A mathematical theory of communication. *Bell Syst. Tech. J.*, 27, 379–423.

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 *50th IEEE Conf. Decis. Control Eur. Control Conf.*

Wang, H., Guo, Y., and Zhang, H. (2020). A method of Ultra-Large-Scale matrix inversion using block recursion. *Information*, 11(11).

Williams, C. and Seeger, M. (2000). Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press.