Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-28 (2019) 7–12

A Kernel Principal Component Regressor for LPV System Identification [★]

Paulo Lopes dos Santos [*] **T-P Azevedo Perdicoúlis** [**]

[*] *INESC TEC - Institute for Systems and Computer Engineering, Technology and Science and FEUP - Faculty of Engineering, University of Porto, Rua Dr Roberto Frias, s/n 4200-464 Porto, Portugal (e-mail: pjsantos@fe.up.pt).*
[**] *Universidade de Trás-os-Montes e Alto Douro, Vila Real, and Instituto de Sistemas e Robótica (ISR) Coimbra both in Portugal (e-mail: tazevedo@utad.pt).*

Abstract: This article describes a Kernel Principal Component Regressor (KPCR) to identify Auto Regressive eXogenous (ARX) Linear Parmeter Varying (LPV) models. The new method differs from the Least Squares Support Vector Machines (LS-SVM) algorithm in the regularisation of the Least Squares (LS) problem, since the KPCR only keeps the principal components of the Gram matrix while LS-SVM performs the inversion of the same matrix after adding a regularisation factor. Also, in this new approach, the LS problem is formulated in the primal space but it ends up being solved in the dual space overcoming the fact that the regressors are unknown.

The method is assessed and compared to the LS-SVM approach through 2 Monte Carlo (MC) experiments. Every experiment consists of 100 runs of a simulated example, and a different noise level is used in each experiment,with Signal to Noise Ratios of 20db and 10db, respectively. The obtained results are twofold, first the performance of the new method is comparable to the LS-SVM, for both noise levels, although the required calculations are much faster for the KPCR. Second, this new method reduces the dimension of the primal space and may convey a way of knowing the number of basis functions required in the Kernel. Furthermore, having a structure very similar to LS-SVM makes it possible to use this method in other types of models, e.g. the LPV state-space model identification.

Keywords: System Identification, LPV Systems, Arx, Kernel, Kernel Regression, Least squares, Principal Components, Least Squares Support Vector Machines.

1. INTRODUCTION

Nonparametric approaches present a way to bypass the difficulties associated with the selection of basis functions in Linear Parameter Varying (LPV) system identification. As a result, they have received a significant attention in recent literature (Hsu et al. (2008); Tóth et al. (2011); Laurain et al. (2012); Piga and Tóth (2013); Abbasi et al. (2014); Lopes dos Santos et al. (2014); Rizvi et al. (2015); Mejari et al. (2016); Romano et al. (2016b,a); Rizvi et al. (2018); Lima et al. (2018)). Tóth et al. (2011) proposed a nonparametric Kernel method using a Least Squares Support Vector Machine (LS-SVM) framework (Suykens et al. (2002)) to identify Auto Regressive eXogenous (ARX) LPV models. The LS-SVM solves a linear Least Squares (LS) problem in a computationally efficient way and is capable of capturing difficult nonlinear dependencies. Laurain et al. (2012) came up with an iterative refined instrumented LPV LS-SVM to improve the method under more general noise conditions. In Abbasi et al. (2014), the LS-SVM estimator was modified to cope with noise in the scheduling signal. In Piga and Tóth (2013) and in Mejari et al. (2016), the model order selection of the ARX models estimated by the LS-SVM was addressed. The LS-SVM was also used to identify LPV state-space models. Firstly with a mixed parametric non– parametric approach in Lopes dos Santos et al. (2014), and later with full non–parametric approaches in Rizvi et al. (2015); Romano et al. (2016b,a); Rizvi et al. (2018) and Lima et al. (2018).

The Kernel Principal Component Regressor (KPCR) was proposed by (Schölkopf et al., 1998) and generalises Principal Component Analysis to the estimation of non-linear models. Since then, it has been studied by Rosipal et al. (2000, 2001); Rosipal and Trejo (2001); Hoegaerts et al. (2005); Wibowo and Yamamoto (2012), among many other authors. Although being widely used in classification problems, to the best of the authors knowledge, it still has not been applied to system identification problems. In this paper, a KPCR algorithm for the identification of LPV systems is derived. It is shown to be very similar to the LS-SVM, differing only in the way the LS problem is regularised. KPCR only keeps the Principal Components of the Gram matrix while LS-SVM performs its inversion after adding a regularisation factor. Although only the

[★] This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project: UID/EEA/50014/2019.

Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.12.339

ARX LPV model structure is addressed, being very similar to the LS-SVM algorithm, the herein described algorithm can also be used in other approaches where the LS-SVM was successfully applied.

The paper is organised in the following form: After this introductory Section, the ARX-LPV model structure is presented in Section 2, where the model parameters are described as linear combinations of unknown basis functions and the output predictor is derived in a linear regression form. The KPCR is derived in Section 3. First, the LS problem is solved in the primal space. But, since the regressors are unknown, the Kernel function is introduced and the problem is solved in the dual space. The equations of the model parameters as functions of the dual-space parameters are derived in Section 4. In Section 5 it is shown that the KPCR and LS-SVM are closely related. The KPCR and LS-SVM are compared in Section 6 in a case study consisting of 2 Monte Carlo (MC) experiments that identify an ARX-LPV system with Signal to Noise Ratios (SNR) of 20dB and 10dB. Both produced estimates with similar accuracy. Finally, in Section 7, some conclusions are withdrawn and some directions for future work are outlined.

2. ARX LPV SYSTEMS

A Single-Input-Single-Ouput (SISO) ARX-LPV system is described by

$$\begin{aligned} y_k = {} & b^1(p_k)u_{k-1} + \cdots + b^{n_b}(p_k)u_{k-n_b} - \\ & a^1(p_k)y_{k-1} - \cdots - a^{n_a}(p_k)y_{k-n_a} + e_k. \end{aligned} \tag{1}$$

where e_k is a zero mean white noise sequence and $b^i(p)$, $i = 1, \ldots, nb$ and $a^i(p)$, $i = 1, \ldots, n_a$, are functions $\mathbb{R}^{n_p} \to \mathbb{R}$ of a scheduling signal $p_k \in \mathbb{R}^{n_p}$. An usual assumption in LPV system identification is that both $b^i(p)$ and $a^i(p)$ are linear combinations of a set of basis functions, i.e.,

$$b^i(p) = \sum_{j=1}^{n_\psi} \beta^{i,j}\psi^j(p) = {\boldsymbol{\beta}^i}^T \boldsymbol{\psi}(p), \quad i = 1, \ldots, n_b, \tag{2}$$

$$a^i(p) = \sum_{j=1}^{n_\psi} \alpha^{i,j}\psi^j(p) = {\boldsymbol{\alpha}^i}^T \boldsymbol{\psi}(p), \quad i = 1, \ldots, n_a, \tag{3}$$

with

$$\boldsymbol{\beta}^i = \left[\beta^{i,1} \ \cdots \ \beta^{i,n_\psi}\right]^T \in \mathbb{R}^{n_\psi}, \tag{4}$$

$$\boldsymbol{\alpha}^i = \left[\alpha^{i,1} \ \cdots \ \alpha^{i,n_\psi}\right]^T \in \mathbb{R}^{n_\psi}, \tag{5}$$

$$\boldsymbol{\psi}(p) = \left[\psi^1(p) \ \cdots \ \psi^{n_\psi}(p)\right]^T \in \mathbb{R}^{n_\psi}. \tag{6}$$

Hence, equation (1) becomes

$$\begin{aligned} y_k = {} & {\boldsymbol{\beta}^1}^T \boldsymbol{\psi}(p_k)u_{k-1} + \cdots + {\boldsymbol{\beta}^{n_b}}^T \boldsymbol{\psi}(p_k)u_{k-n_b} - \\ & {\boldsymbol{\alpha}^1}^T \boldsymbol{\psi}(p_k)y_{k-1} - \ldots {\boldsymbol{\alpha}^{n_a}}^T \boldsymbol{\psi}(p_k)y_{k-n_a} + e_k. \end{aligned} \tag{7}$$

Defining

$$\boldsymbol{\beta} = \left[{\boldsymbol{\beta}^1}^T \ \ldots \ {\boldsymbol{\beta}^{n_b}}^T\right]^T \in \mathbb{R}^{n_\psi n_b}, \tag{8}$$

$$\boldsymbol{\alpha} = \left[{\boldsymbol{\alpha}^1}^T \ \cdots \ {\boldsymbol{\alpha}^{n_a}}^T\right]^T \in \mathbb{R}^{n_\psi n_a}, \tag{9}$$

then

$$y_k = \boldsymbol{\beta}^T \begin{bmatrix} \boldsymbol{\psi}(p_k)u_{k-1} \\ \vdots \\ \boldsymbol{\psi}(p_k)u_{k-n_b} \end{bmatrix} - \boldsymbol{\alpha}^T \begin{bmatrix} \boldsymbol{\psi}(p_k)y_{k-1} \\ \cdots \\ \boldsymbol{\psi}(p_k)y_{k-n_a} \end{bmatrix} + e_k. \tag{10}$$

Futhermore, collecting the past inputs and the past outputs up to $k - n_b$ and $k - n_a$ in $\mathbf{u}_{k-1}$ and $\mathbf{y}_{k-1}$, respectively, as

$$\mathbf{u}_{k-1} = \left[u_{k-1} \ \cdots \ u_{k-n_b}\right]^T \in \mathbb{R}^{n_b}, \tag{11}$$

$$\mathbf{y}_{k-1} = \left[y_{k-1} \ \cdots \ y_{k-n_a}\right]^T \in \mathbb{R}^{n_b}, \tag{12}$$

the ARX-LPV model can be re-written in the compacted form

$$y_k = \boldsymbol{\beta}^T \mathbf{u}_{k-1} \otimes \boldsymbol{\psi}(p_k) - \boldsymbol{\alpha}^T \mathbf{y}_{k-1} \otimes \boldsymbol{\psi}(p_k) + e_k. \tag{13}$$

Finally, defining the regressor

$$\boldsymbol{\varphi_k} = \left[\mathbf{u}_{k-1}^T \ -\mathbf{y}_{k-1}^T\right]^T \in \mathbb{R}^{n_b+n_a} \tag{14}$$

and the parameter vector

$$\boldsymbol{\theta} = \left[\boldsymbol{\beta}^T \ \boldsymbol{\alpha}^T\right]^T \in \mathbb{R}^{n_\psi(n_b+n_a)}, \tag{15}$$

the model can be rewritten as

$$y_k = \boldsymbol{\theta}^T \left[\boldsymbol{\varphi}_k \otimes \boldsymbol{\psi}(p_k)\right] + e_k = \left[\boldsymbol{\varphi}_k \otimes \boldsymbol{\psi}(p_k)\right]^T \boldsymbol{\theta} + e_k \tag{16}$$

where

$$\hat{y}_k = \left[\boldsymbol{\varphi}_k \otimes \boldsymbol{\psi}(p_k)\right]^T \boldsymbol{\theta} \tag{17}$$

is the predictor of y_k.

3. KERNEL PRINCIPAL COMPONENT REGRESSOR

Let $n = \max(n_b, n_a)$. If there are $N + n$ observations of the triple input-output-scheduling signals, (u_k, y_k, p_k), $k = n + 1, \ldots, n + N$, the outputs, the regressors and the noise are gathered in

$$\mathbf{Y} = \left[y_{n+1} \ \cdots \ y_{n+N}\right]^T \in \mathbb{R}^N, \tag{18}$$

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \boldsymbol{\varphi}_{n+1}^T \otimes \boldsymbol{\psi}(p_{n+1})^T \\ \vdots \\ \boldsymbol{\varphi}_{n+N}^T \otimes \boldsymbol{\psi}(p_{n+N})^T \end{bmatrix} \in \mathbb{R}^{N \times n_\psi(n_b+n_a)}, \tag{19}$$

$$\boldsymbol{\mathcal{E}} = \left[e_{n+1} \ \cdots \ e_{n+N}\right]^T \in \mathbb{R}^N, \tag{20}$$

to have

$$\mathbf{Y} = \boldsymbol{\Upsilon}\boldsymbol{\theta} + \boldsymbol{\mathcal{E}}, \tag{21}$$

$\boldsymbol{\theta}$ may be found by solving this equation in the LS sense. However, this cannot be done if both the basis functions, $\psi_i(p)$, and its number, n_ψ, are unknown. Yet, the problem can be solved if the goal is to find either the output predictor or the functions $b^i(p)$, $i = 1, \ldots, n_b$ or $a^i(p)$, $i = 1, \ldots, n_a$. Hence, consider the singular value decomposition (svd) of $\boldsymbol{\Upsilon}$,

$$\boldsymbol{\Upsilon} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \tag{22}$$

where $\mathbf{U} \in \mathbf{R}^{N \times N}$ and $\mathbf{V} \in \mathbb{R}^{n_\psi(n_b+n_a) \times n_\psi(n_b+n_a)}$ orthonormal matrices whose columns are the left and right singular vectors of $\boldsymbol{\Upsilon}$, respectively, and $\mathbf{S} \in \mathbb{R}^{N \times n_\psi(n_b+n_a)}$ is a diagonal matrix whose elements are the singular values. It is easy to see that

$$\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T = \mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^T \in \mathbb{R}^{N \times N}, \tag{23}$$

where $\boldsymbol{\Sigma}^2 = \mathbf{S}\mathbf{S}^T$, is the eigendecomposition of $\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T$. The columns of $\mathbf{U}$ are the eigenvectors and the entries of the main diagonal of $\boldsymbol{\Sigma}$ are the eigenvalues. This is also the svd because $\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T$ is a semi-definite matrix. From (19), and after a few simple calculations, thus

$$\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T\right) \circ \left(\boldsymbol{\Psi}\boldsymbol{\Psi}^T\right), \tag{24}$$

where $\circ$ denotes the Hadamard product (element-wise multiplication), also known as the Schur product, and

$$\boldsymbol{\Phi} = \left[\boldsymbol{\varphi}_{n+1} \ \cdots \ \boldsymbol{\varphi}_{n+N}\right]^T \in \mathbb{R}^{N\times(n_a+n_b)} \tag{25}$$

$$\boldsymbol{\Psi} = \left[\boldsymbol{\psi}(p_{n+1}) \ \cdots \ \boldsymbol{\psi}(p_{n+N})\right]^T \in \mathbb{R}^{N\times n_\psi}. \tag{26}$$

The i, j entries of $\boldsymbol{\Psi}\boldsymbol{\Psi}^T$ are the inner products $\boldsymbol{\psi}(p_i)^T\boldsymbol{\psi}(p_j)$. Hence they can be replaced by a Kernel function satisfying Mercer theorem, i.e.,

$$\boldsymbol{\psi}(p_i)^T\boldsymbol{\psi}(p_j) = K^\psi(p_i, p_j), \tag{27}$$

where $K^\psi(\cdot,\cdot)$ is such a Kernel function. By doing this, $\boldsymbol{\Psi}\boldsymbol{\Psi}^T$ can be replaced by the Gram matrix

$$\boldsymbol{\mathcal{K}}_N^\psi = \begin{bmatrix} K^\psi(p_{n+1}, p_{n+1}) & \cdots & K^\psi(p_{n+1}, p_{n+N}) \\ \vdots & \vdots & \vdots \\ K^\psi(p_{n+N}, p_{n+1}) & \cdots & K^\psi(p_{n+N}, p_{n+N}) \end{bmatrix} \in \mathbb{R}^{N\times N}. \tag{28}$$

Notice that $\boldsymbol{\Phi}\boldsymbol{\Phi}^T$ is also a Gram matrix because its elements are inner products too. But here the factors are known, so there is no need to use any other Kernel function to denote them (the inner product is itself a Kernel function known as the linear Kernel). Denoting $\boldsymbol{\Phi}\boldsymbol{\Phi}^T$ as $\boldsymbol{\mathcal{K}}_N^\varphi$, then $\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T$ can be replaced by

$$\boldsymbol{\mathcal{K}}_N = \boldsymbol{\mathcal{K}}_N^\varphi \circ \boldsymbol{\mathcal{K}}_N^\psi, \tag{29}$$

and $\mathbf{U}$ and $\boldsymbol{\Sigma}^2$ defined in (23) can be found from the svd decomposition

$$\boldsymbol{\mathcal{K}}_N = \mathbf{U}\boldsymbol{\Sigma}^2\mathbf{U}^T. \tag{30}$$

As $\boldsymbol{\mathcal{K}}_N$ and $\boldsymbol{\Upsilon}$ have the same column space, they have the same rank. Hence, when rank$\left(\boldsymbol{\mathcal{K}}_N\right) = r < N$, the same happens with $\boldsymbol{\Upsilon}$ and (22) may be replaced by the reduced svd

$$\boldsymbol{\Upsilon} = \mathbf{U}_r\boldsymbol{\Sigma}_r\mathbf{V}_r^T \tag{31}$$

where $\mathbf{U}_r \in \mathbb{R}^{N\times r}$ and $\mathbf{V}_r \in \mathbb{R}^{n_\psi\times r}$ are matrices with the r-first columns of $\mathbf{U}$ and $\mathbf{V}$, and $\boldsymbol{\Sigma}_r \in \mathbb{R}^{r\times r}$ contains the nonzero singular values. The Minimal Norm LS Estimator of $\boldsymbol{\theta}$ is then given by

$$\hat{\boldsymbol{\theta}} = \boldsymbol{\Upsilon}^\dagger\mathbf{Y} = \mathbf{V}_r\boldsymbol{\Sigma}_r^{-1}\mathbf{U}_r^T\mathbf{Y}. \tag{32}$$

Replacing $\boldsymbol{\theta}$ by $\hat{\boldsymbol{\theta}}$ in (17), yields

$$\hat{y}_k = \left[\boldsymbol{\varphi}_k \otimes \boldsymbol{\psi}(p_k)\right]^T\hat{\boldsymbol{\theta}} = \left[\boldsymbol{\varphi}_k \otimes \boldsymbol{\psi}(p_k)\right]^T\mathbf{V}_r\boldsymbol{\Sigma}_r\mathbf{U}_r^T\mathbf{Y}. \tag{33}$$

Yet, $\hat{y}_k$ cannot be calculated because both $\boldsymbol{\psi}(p_k)$ and $\mathbf{V}_r$ are unknown. But if $\boldsymbol{\varphi}_k \otimes \boldsymbol{\psi}(p_k)$ is in the row-space of $\boldsymbol{\Upsilon}$, then it is equal to its orthogonal projection into this space, i.e,

$$\boldsymbol{\varphi}_k \otimes \boldsymbol{\psi}(p_k) = \boldsymbol{\Upsilon}^T\left(\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T\right)^\dagger\boldsymbol{\Upsilon}\left[\boldsymbol{\varphi}_k \otimes \boldsymbol{\psi}(p_k)\right] = \mathbf{V}_r\mathbf{S}_r^{-1}\mathbf{U}_r^T\boldsymbol{\Upsilon}\left[\boldsymbol{\varphi}_k \otimes \boldsymbol{\psi}(p_k)\right]. \tag{34}$$

Using this result in (33) yields,

$$\hat{y}_k = \left[\boldsymbol{\varphi}_k \otimes \boldsymbol{\psi}(p_k)\right]^T\hat{\boldsymbol{\theta}} = \left[\boldsymbol{\varphi}_k \otimes \boldsymbol{\psi}(p_k)\right]^T\boldsymbol{\Upsilon}^T\boldsymbol{\mathcal{K}}_N^\dagger\mathbf{Y}. \tag{35}$$

As

$$\boldsymbol{\Upsilon}\left[\boldsymbol{\varphi}_k \otimes \boldsymbol{\psi}(p_k)\right] = \begin{bmatrix} \boldsymbol{\varphi}_{n+1}^T\boldsymbol{\varphi}_k \\ \vdots \\ \boldsymbol{\varphi}_{n+N}^T\boldsymbol{\varphi}_k \end{bmatrix} \circ \begin{bmatrix} \boldsymbol{\psi}(p_{n+1})^T\boldsymbol{\psi}(p_k) \\ \vdots \\ \boldsymbol{\psi}(p_{n+N})^T\boldsymbol{\psi}(p_k) \end{bmatrix}, \tag{36}$$

then the inner products $\boldsymbol{\psi}(p_i)^T\boldsymbol{\psi}(p_k)$, $i = n, \ldots, n+N$ can be replaced by the Kernel function $K(p_i, p_k)$ and (35) becomes

$$\hat{y}_k = \left(\boldsymbol{\mathcal{K}}_k^\varphi \circ \boldsymbol{\mathcal{K}}_k^\psi\right)^T\boldsymbol{\mathcal{K}}_N^\dagger\mathbf{Y} \tag{37}$$

with

$$\boldsymbol{\mathcal{K}}_k^\psi = \left[\mathcal{K}^\psi(p_{n+1}, p_k) \ \cdots \ \mathcal{K}^\psi(p_{n+N}, p_k)\right]^T, \tag{38}$$

$$\boldsymbol{\mathcal{K}}_k^\varphi = \left[\boldsymbol{\varphi}_{n+1}^T\boldsymbol{\varphi}_k \ \cdots \ \boldsymbol{\varphi}_{n+N}^T\boldsymbol{\varphi}_k\right]^T. \tag{39}$$

Finally, defining

$$\hat{\boldsymbol{\Lambda}} = \boldsymbol{\mathcal{K}}_N^\dagger\mathbf{Y} \in \mathbb{R}^N, \tag{40}$$

the output predictor equation may be rewritten as

$$\hat{y}_k = \left(\boldsymbol{\mathcal{K}}_k^\varphi \circ \boldsymbol{\mathcal{K}}_k^\psi\right)^T\hat{\boldsymbol{\Lambda}}. \tag{41}$$

As seen above, it is assumed that $\boldsymbol{\varphi}_k \otimes \boldsymbol{\psi}(p_k)$ is in the row-space of $\boldsymbol{\Upsilon}$. Hence, for accurate ouput predictions, rank$(\boldsymbol{\Upsilon})$ = rank$(\boldsymbol{\mathcal{K}}_N) << N$. But, due to noise conditions the rank of these matrices might be to high. Therefore, only the principal components of $\boldsymbol{\mathcal{K}}_N$ are taken and this approximated by

$$\hat{\boldsymbol{\mathcal{K}}}_N = U_{n_k}\Sigma_{n_k}U_{n_k}^T \tag{42}$$

where $U_{n_k} \in \mathbb{R}^{N\times n_k}$ is a matrix whose columns are the n_k first singular vectors of $\boldsymbol{\mathcal{K}}_N$ and $\Sigma_{n_k} \in \mathbb{R}^{n_k\times n_k}$ a diagonal matrix with the first n_k singular values. A small n_k decreases the variance of the estimates but also increases the bias. Thus, the use of the principal components acts as a regularisation. Like the regularisation factor in a Ridge regression, the number of principal components is considered as a hyper-parameter. Notice, however, that finding n_k is similar to the model order selection in a LS problem. Hence, methods for determining model order such as the Aikaike information criterion (AIC), the Akaike final prediction error (FPE), etc., can be considered to find n_k.

4. MODEL PARAMETERS

From (40), $\hat{\boldsymbol{\Lambda}}$ is the minimal norm LS estimator of $\boldsymbol{\Lambda}$ such that

$$\boldsymbol{Y} = \boldsymbol{\mathcal{K}}_N\boldsymbol{\Lambda} + \boldsymbol{\mathcal{E}}. \tag{43}$$

As $\boldsymbol{\mathcal{K}}_N$ stands for $\boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T$, this equation may be written as

$$\boldsymbol{Y} = \boldsymbol{\Upsilon}\boldsymbol{\Upsilon}^T\boldsymbol{\Lambda} + \boldsymbol{\mathcal{E}}. \tag{44}$$

Comparing with (21) it can be concluded that

$$\boldsymbol{\theta} = \boldsymbol{\Upsilon}^T\boldsymbol{\Lambda}, \tag{45}$$

i.e. $\hat{\boldsymbol{\Lambda}}$ is the solution of the LS problem defined by (21) in a dual space.

From the definitions of $\boldsymbol{\Upsilon}$, $\boldsymbol{\varphi_k}$, $\mathbf{u}_{k-1}$, and $\mathbf{y}_{k-1}$, in (19), (14), (11) and (12),

$$\boldsymbol{\theta} = \left[\boldsymbol{\varphi}_{n+1} \otimes \boldsymbol{\psi}(p_{n+1}) \ \cdots \ \boldsymbol{\varphi}_{n+N} \otimes \boldsymbol{\psi}(p_{n+N})\right] \boldsymbol{\Lambda} = \tag{46}$$

$$\begin{bmatrix} \mathbf{u}_n \otimes \boldsymbol{\psi}(p_{n+1}) \cdots & \mathbf{u}_{n+N-1} \otimes \boldsymbol{\psi}(p_{n+N}) \\ \mathbf{y}_n \otimes \boldsymbol{\psi}(p_{n+1}) \cdots & \mathbf{y}_{n+N-1} \otimes \boldsymbol{\psi}(p_{n+N}) \end{bmatrix} \boldsymbol{\Lambda} =$$

$$\begin{bmatrix} u_n\boldsymbol{\psi}(p_{n+1}) & \cdots & u_{n+N-1}\boldsymbol{\psi}(p_{n+N}) \\ \vdots & \vdots & \vdots \\ u_{n-n_b+1}\boldsymbol{\psi}(p_{n+1}) & \cdots & u_{n+N-n_b}\boldsymbol{\psi}(p_{n+N}) \\ -y_n\boldsymbol{\psi}(p_{n+1}) & \cdots & -y_{n+N-1}\boldsymbol{\psi}(p_{n+N}) \\ \vdots & \vdots & \vdots \\ -y_{n-n_a+1}\boldsymbol{\psi}(p_{n+1}) & \cdots & -y_{n+N-n_a}\boldsymbol{\psi}(p_{n+N}) \end{bmatrix} \boldsymbol{\Lambda}.$$

Moreover, from (15),

$$\boldsymbol{\beta}^i = \left[u_{n-i+1}\boldsymbol{\psi}(p_{n+1}) \ \cdots \ u_{n+N-n_b}\boldsymbol{\psi}(p_{n+N})\right] \boldsymbol{\Lambda} = \sum_{j=n+1}^{n+N} \lambda_j u_{j-i} \boldsymbol{\psi}(p_j), \qquad i = 1, \ldots, n_b$$

$$\boldsymbol{\alpha}^i = -\left[y_{n-i+1}\boldsymbol{\psi}(p_{n+1}) \ \cdots \ y_{n+N-n_b}\boldsymbol{\psi}(p_{n+N})\right] \boldsymbol{\Lambda} = -\sum_{j=n+1}^{n+N} \lambda_j y_{j-i} \boldsymbol{\psi}(p_j), \qquad i = 1, \ldots, n_a.$$

Finally, from the definitions of $b^i(p)$ and $a^i(p)$ in (2) e (3), and of $K^{\psi}(\cdot,\cdot)$ in (27), hence

$$b^i(p) = {\boldsymbol{\beta}^i}^T \boldsymbol{\psi}(p) = \sum_{j=n+1}^{n+N} \lambda_j u_{j-i} K^{\psi}(p_j, p), \tag{47}$$

$$a^i(p) = {\boldsymbol{\alpha}^i}^T \boldsymbol{\psi}(p) = -\sum_{j=n+1}^{n+N} \lambda_j y_{j-i} K^{\psi}(p_j, p). \tag{48}$$

5. RELATION WITH THE LS-SVM ESTIMATOR

The LS-SVM proposed by Tóth *et. al.*, to identify ARX-LPV models solves the following problem

$$\min_{\boldsymbol{\theta},\boldsymbol{\mathcal{E}}} \mathcal{J}(\boldsymbol{\theta},\boldsymbol{\mathcal{E}}) = \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\theta} + \frac{\gamma}{2}\boldsymbol{\mathcal{E}}^{\boldsymbol{T}}\boldsymbol{\mathcal{E}} \tag{49}$$

$$\text{such that} \qquad \boldsymbol{\mathcal{E}} = \mathbf{Y} - \boldsymbol{\Upsilon}\boldsymbol{\theta}.$$

in a dual space, by finding the vector $\boldsymbol{\Lambda}$ of Lagrange multipliers, that optimises the Lagrangian

$$\mathcal{L} = \left(\boldsymbol{\theta}, \boldsymbol{\mathcal{E}}, \hat{\boldsymbol{\Lambda}}\right) = \mathcal{J}(\boldsymbol{\theta},\boldsymbol{\mathcal{E}}) - \hat{\boldsymbol{\Lambda}}^T \left(\boldsymbol{\mathcal{E}} + \boldsymbol{\Upsilon}\boldsymbol{\theta} - \mathbf{Y}\right). \tag{50}$$

The solution is

$$\hat{\boldsymbol{\Lambda}} = \left(\boldsymbol{\mathcal{K}}_N + \frac{1}{\gamma}\mathbf{I}_n\right)^{-1} \mathbf{Y}. \tag{51}$$

which is similar to the $\hat{\boldsymbol{\Lambda}}$ estimate of the KPCR in (40). Moreover, both the predictor and model parameters are given by (41), (47) and (48). Hence, it is expectable the KPCR and LS-SVM estimators to be almost equal. However, the KPCR is computationally more efficient because

(1) The regularisation consists in adding or removing principal components to the Gram matrix without having to invert it.
(2) The number of principal components can be found by the AIC or FPE Akaike criteria. Hence, it is not necessary to simulate the system to find the hyper-parameters.

6. CASE STUDY

In this section, the KPCR and LS-SVM performance are compared using the following simulated example

$$y_k + a_1(p_k)y_{k-1} = b_1(p_k)u_{k-1} + b_2(p_k)u_{k-2} + e_k \tag{52}$$

with

$$b_1(p) = \begin{cases} -0.5, & p < -0.5 \\ p, & -0.5 \leq p \leq 0.5 \\ 0.5, & p > 0.5 \end{cases} \tag{53}$$

$$b_2(p) = -0.2p^2 \tag{54}$$

$$a_1(p) = 0.1\frac{\sin(\pi^2 p)}{\pi^2 p}. \tag{55}$$

MC experiments with 100 runs and SNR=20dB and SNR=10dB, respectively, have been carried out to compare the statistical properties and reconstruction capabilities of the algorithms. Two versions of the KPCR algorithm have been implemented. Both versions use the radius basis function (RBF) as the Kernel function, i.e.,

$$K(p_i, p_j) = \exp\left(-\frac{1}{\sigma^2}(p_i - p_j)^2\right). \tag{56}$$

The difference is the way in which the hyper-parameters n_k, the number principal components, and σ are tuned. In both cases they are selected from a pre-specified grid but using different criteria:

KPCR1 - minimises the Akaike FPE index defined as

$$FPE = \boldsymbol{\mathcal{E}}^T\boldsymbol{\mathcal{E}}\frac{N + n_k}{N - n_k}, \tag{57}$$

where $\boldsymbol{\mathcal{E}} = \mathbf{Y} - \boldsymbol{\mathcal{K}}_N\hat{\boldsymbol{\Lambda}} \in \mathbb{R}^N$ are the residuals, using the training data.

KPCR2 - minimises

$$J_v = \boldsymbol{\mathcal{E}}_v^T\boldsymbol{\mathcal{E}}_v \tag{58}$$

where $\boldsymbol{\mathcal{E}}_v$ is the simulation error over a validation data set. The LS-SVM algorithm also uses the RBF in (56) as the Kernel function. Also, the hyper-parameters, γ, the regularisation factor, and σ, are selected from a pre-specified grid (with the same range of σ) and minimising the **KPCR2** criterion (58).

1500 input-output data points were generated in every run of each MC experiment. The input and scheduling signals, u_k and p_k, were zero mean independent white noise sequences uniformly distributed in the interval $[-1, 1]$ and the equation error, e_k, was a zero mean gaussian white noise with a variance determined by the SNR level. The first 750 data points were the training data, and the other half was used to calculate the hyper-parameters of the **KPCR2** and **LS-SVM** algorithms. Additionally, a set of 750 noiseless data points was generated with the input and scheduling signals being characterised as before but independently generated. This set has been used to access the accuracy of the estimated models.

Table 1 displays the minimum, average and maximum values of the Best Fit Rate (BFR) Index defined as

$$BFR := 100\% \max\left(1 - \frac{\|y - \hat{y}\|_2}{\|y - \bar{y}\|}, 0\right),$$

where y is the output of third data set, $\bar{y}$ the mean of y, and $\hat{y}$ the simulated output with the estimated model.

Table 1. BFRs of the MC experiments.

	BFR			
	SNR	Min	Avg	Max
KPCR1	20dB	94.2550%	97.1928%	97.9748%
	10dB	86.2668%	93.2127%	95.7933%
KPCR2	20dB	96.4880 %	97.4691 %	98.2624%
	10dB	91.7072%	93.7132%	95.6421%
LS-SVM	20dB	96.8511%	97.5330%	98.1455%
	10dB	92.2561%	94.0650%	96.0537%

This table shows that all estimated models accurately simulated the system revealing a high performance index. **KPCR1** estimates the model much faster, although it produces slightly lower BFRs. Figures 1-3 display the real and estimated values of the parameters $b_1(p)$, $b_2(p)$ and $a_1(p)$, respectively, for SNR=10dB. Solid black lines are the real values, dashed red lines the **KPCR1** estimates (average, average plus/minus standard deviation), dotted blue lines the **KPCR2** and dash-dotted yellow lines the **LS-SVM**. All the algorithms show comparable accuracy in the estimates of $b_1(p)$ and $b_2(p)$. In Figure 3, it can be observed that the estimates of $a_1(p)$ exhibit more variability, but, as shown in Table 1, the precision of the simulations has not been affected, denoting a low sensitivity of the model to this parameter. It can also be seen in this Figure that the **LS-SVM** algorithm yielded the most biased estimates. This larger bias was compensated by a lower variance. **KPCR1** has the lowest bias and the higher variance which is due to a larger number of retained principal components.

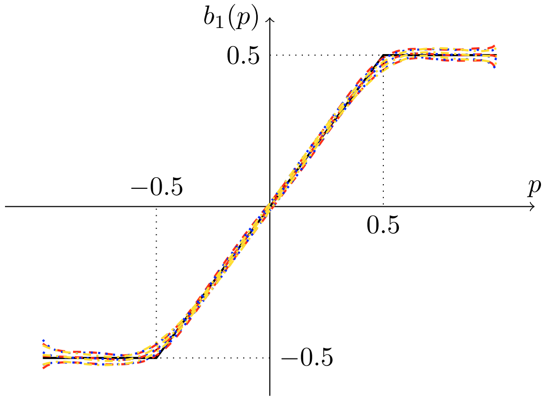

Fig. 1. Parameter $b_1(p)$, SNR=10dB.

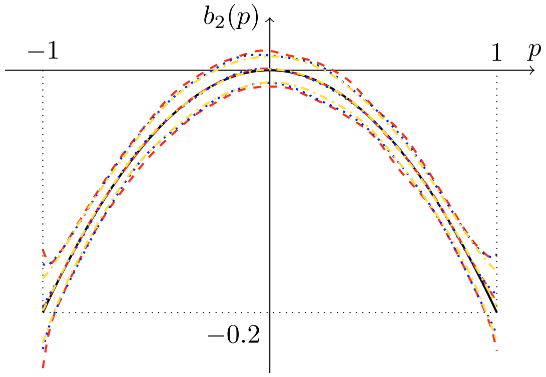

Fig. 2. Parameter $b_2(p)$, SNR=10dB.

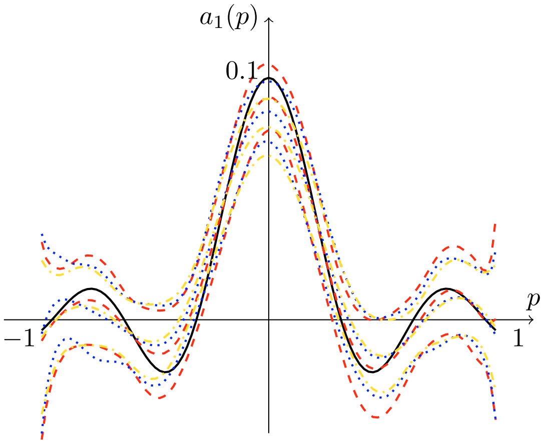

Fig. 3. Parameter $a_1(p)$, SNR=10dB.

7. CONCLUSIONS AND FUTURE WORK

In this paper, a KPCR algorithm for the identification of ARX-LPV systems is proposed. The LPV parameters are linear combinations of unknown basis functions. A LS problem is formulated in a primal space, but it is solved in a dual space, using a Kernel function to avoid the lack of knowledge of the regressors. Only the principal components of the Gram matrix are retained to circumvent its singularity and ill conditioning. The number of principal components also defines the dimension of the primal space and may be determined by the AIC and FPE Akaike criteria yielding a high computationally efficient way of determining the hyper-parameters. The new algorithm is closely related to the LS-SVM approach, recently introduced to solve the same problem. Its main difference is the regularisation of the Gram matrix. In a case study two versions of the KPCR algorithm (**KPCR1** and **KPCR2**) were compared with the **LS-SVM** through MC experiments with SNR=20dB and SNR=10dB. The different versions are obtained by using different criteria to select the hyper-parameters. The precision shown by both versions is comparable to the one of the **LS-SVM** for both noise levels. The **KPCR1** produced the less biased estimates although with the higher variance.

In future work, the KPCR approach will be used to identify more general LPV model structures, including state-space LPV with the MOLI parameterisation (Romano et al. (2016b)). It was referred above that the number of principal components defines the dimension of the primal space where the underlying LS problem is defined. Based on this, it will be investigated if a solution can be found in this primal space. Finally, the predictor in (41) assumes that the regressor $\boldsymbol{\varphi}_k \otimes \boldsymbol{\psi}(p_k)$ is in the row-space of the regressors' matrix $\boldsymbol{\Upsilon}$. It will be studied whether it is possible to check this condition prior to the simulation in a unsupervised data-driven learning framework.

REFERENCES

Abbasi, F., Mohammadpour, J., Tóth, R., and Meskin, N. (2014). A support vector machine-based method for lpv-arx identification with noisy scheduling parameters. In 2014 European Control Conference (ECC), 370–375. IEEE.

Hoegaerts, L., Suykens, J.A., Vandewalle, J., and De Moor, B. (2005). Subset based least squares subspace regression in rkhs. Neurocomputing, 63, 293–323.

Hsu, K., Vincent, T.L., and Poolla, K. (2008). Nonparametric methods for the identification of linear parameter varying systems. In 2008 IEEE International Conference on Computer-Aided Control Systems, 846–851. IEEE.

Laurain, V., Toth, R., Zheng, W.X., and Gilson, M. (2012). Nonparametric identification of lpv models under general noise conditions: an ls-svm based approach. IFAC Proceedings Volumes, 45(16), 1761–1766.

Lima, M.M., Romano, R.A., dos Santos, P.L., and Pait, F. (2018). An extended instrument variable approach for nonparametric lpv model identification. IFAC-PapersOnLine, 51(26), 81–86.

Lopes dos Santos, P., Azevedo-Perdicoulis, T., Ramos, J.A., Deshpande, S., Rivera, D.E., and De Carvalho, J.M. (2014). Lpv system identification using a separable least squares support vector machines approach. In 53rd IEEE Conference on Decision and Control, 2548–2554. IEEE.

Mejari, M., Piga, D., and Bemporad, A. (2016). Regularized least square support vector machines for order and structure selection of lpv-arx models. In 2016 European Control Conference (ECC), 1649–1654. IEEE.

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. IEEE.

Rizvi, S.Z., Mohammadpour, J., Tóth, R., and Meskin, N. (2015). A kernel-based approach to mimo lpv state-space identification and application to a nonlinear process system. IFAC-PapersOnLine, 48(26), 85–90.

Rizvi, S.Z., Velni, J.M., Abbasi, F., Tóth, R., and Meskin, N. (2018). State-space lpv model identification using kernelized machine learning. Automatica, 88, 38–47.

Romano, R.A., dos Santos, P.L., Pait, F., and Perdicoúlis, T. (2016a). State space lpv model identification using ls-svm: A case-study with dynamic dependence. In 2016 IEEE Conference on Control Applications (CCA), 995–1000. IEEE.

Romano, R.A., dos Santos, P.L., Pait, F., Perdicoúlis, T., and Ramos, J.A. (2016b). Machine learning barycenter approach to identifying lpv state-space models. In 2016 American Control Conference (ACC), 6351–6356. IEEE.

Rosipal, R., Girolami, M., Trejo, L.J., and Cichocki, A. (2001). Kernel pca for feature extraction and denoising in nonlinear regression. Neural Computing & Applications, 10(3), 231–243.

Rosipal, R. and Trejo, L.J. (2001). Kernel partial least squares regression in reproducing kernel hilbert space. Journal of machine learning research, 2(Dec), 97–123.

Rosipal, R., Trejo, L.J., and Cichocki, A. (2000). Kernel principal component regression with em approach to nonlinear principal components extraction. University of Paisley.

Schölkopf, B., Smola, A., and Müller, K.R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. Neural computation, 10(5), 1299–1319.

Suykens, J.A.K., Van Gesten, T., De Brabanter, J., De Moor, B., and Vandewalle, J. (2002). Least squares support vector machines. World Scientific.

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. IEEE.

Wibowo, A. and Yamamoto, Y. (2012). A note on kernel principal component regression. Computational Mathematics and Modeling, 23(3), 350–367.