

Shrinking Complexity of Scheduling Dependencies in LS-SVM Based LPV System Identification

René Duijkers, Roland Tóth, Dario Piga and Vincent Laurain

Abstract— In the past years, Linear Parameter-Varying (LPV) identification has rapidly evolved from parametric identification methods to nonparametric methods allowing the relaxation of restrictive assumptions. For example, Least-Square Support Vector Machines (LS-SVMs) offer an attractive way of estimating LPV models directly from data without requiring from the user to specify the functional dependencies of the model coefficients on the scheduling variable. These methods have also been recently extended in order to automatically determine the model order directly from data by the help of regularization. Nonetheless, despite all these recent improvements, LPV identification methods still require some strong *a priori* such as *i*) the dependencies are static or dynamic, *ii*) it is known which variables are considered to be the scheduling or *iii*) all coefficient functions of the underlying system depend on all scheduling variables. This prevents the complexity of the scheduling dependency of the model to be shrunk gradually and independently until an optimal bias-variance trade off is found. In this paper, a novel reformulation of the LPV LS-SVM approach is proposed which, besides of the non-parametric estimation of the coefficient functions, achieves data-driven coefficient complexity selection via convex optimization. The properties of the introduced approach are illustrated by a simulation study.

I. INTRODUCTION

The need to efficiently develop high-performance control solutions for industrial applications with position dependency, varying operating conditions or nonlinear/time-varying system behavior, has resulted in an increasing interest for model structures that can represent the behavior of such systems with a linear representation, allowing the use of powerful control synthesis tools. *Linear parameter-varying* (LPV) models, introduced in [1], have become a promising way of answering to this need in control engineering [2].

An efficient way to estimate LPV models in general is using a *least-square support vector machine* (LS-SVM) approach [3]. This method allows to model nonlinear systems with an LPV *input-output* (IO) model without requiring from the user to specify the functional dependencies of the model coefficients on the scheduling variables. This is achieved by

R. Duijkers and R. Tóth are with the Department of Electrical Engineering, Eindhoven University of Technology, P.O. Box 513 5600 MB Eindhoven, The Netherlands. E-mail: r.duijkers@student.tue.nl; r.toth@tue.nl.

Dario Piga is with the Dalle Molle Institute for Artificial Intelligence Research, Scuola Universitaria Professionale della Svizzera Italiana, Galleria 1, 6928 Manno, Switzerland. E-mail: dario.piga@supsi.ch

V. Laurain is with the Centre de Recherche en Automatique de Nancy, Université de Lorraine, CNRS, 2 rue Jean Lamour, 54519 Vandoeuvre-lès-Nancy Cedex, France. E-mail: vincent.laurain@univ-lorraine.fr.

This work was supported by the Netherlands Organization for Scientific Research (NWO, grant. no.: 639.021.127).

using the dual solution of a regularized ℓ_2 loss function. The resulting approach provides a direct nonparametric estimation of the dependencies from data under general noise conditions, where the estimate is obtained by solving a set of linear equations [4], [5].

A particular handicap of the of LS-SVM based LPV identification approaches is that the non-parametric estimation is governed by the so-called Kernel functions, which are assumed to depend on all scheduling variables to avoid prior structural specification of the dependencies. However, this corresponds to a prejudice that all coefficient functions of the underlying system depend on all of these variables which is not the case in most practical applications. Commonly, it is not known beforehand which variables play a significant role in the dependence and which variables can be excluded. Taking all of them into account leads to an over-parametrization problem which results in an increased variance and complexity of the estimated models.

In this paper, the aim is to tackle this over-parameterization by estimating the underlying structure of the process. To achieve this, we propose a regularization based LS-SVM method which penalizes coefficient functions that are insensitive w.r.t. variations of the scheduling variable. The introduced penalty on the coefficient functions gradually shrinks the non-dependent functions to constants (including zero), thereby revealing the underlying structure and reducing complexity in terms of model order and dependency. In contrast with a previous work [6], the proposed method does not directly penalize the amplitude of the coefficient functions, but it shrinks the sensitivity (i.e., the derivative) of the functions with respect to the scheduling signals to zero. This means that the dependency can decrease in terms of complexity rather than just completely vanishing as in the sparse estimator approach proposed in [6]. This is highly beneficial when using the estimated models for control synthesis, commonly requiring low coefficient complexity.

The paper is organized as follows: first, in Section II-A, the LPV context in which the estimation problem is addressed is presented. Then, Sections II-B to II-D describe the identification problem and the solution using the LS-SVM estimator and the proposed regularization structure. Finally, the performance of the approach is demonstrated in a Monte-Carlo study in Section III and conclusions are drawn in Section IV.

II. LPV MODEL SELECTION

In this section, the proposed approach to determine the structure and the underlying dependencies of an LPV-IO

model is detailed. Once the LPV model to be identified and its variables are introduced, the proposed regularized identification problem is presented: the objective of coefficient dependency shrinkage is expressed by introducing an additional term into the regularized ℓ_2 prediction-error cost function. Finally an LS-SVM estimator is formulated and it is shown that the introduced regularization term allows a gradual shrinking of the complexity of the estimated functional dependencies.

A. Model structure

For the sake of brevity, the simplest form of LPV models used in the discrete-time IO identification framework is considered, which is the *autoregressive with exogenous input* (ARX) structure, defined, in the *single-input single-output* (SISO) case, as

$$y(k) = \sum_{i=1}^{n_a} a_i(p(k))y(k-i) + \sum_{j=0}^{n_b} b_j(p(k))u(k-j) + e(k). \quad (1)$$

Here, $k \in \mathbb{Z}$ denotes the discrete time, $e(k) \in \mathbb{R}$ is assumed to be a zero-mean white noise, $u(k) \in \mathbb{R}$ and $y(k) \in \mathbb{R}$ represent the measured input and output signals of the system to be identified, respectively, and $p(k) \in \mathbb{R}^{n_p}$ is the so-called *scheduling variable*, which ranges in a compact set $\mathbb{P} \subseteq \mathbb{R}^{n_p}$ and, according to the literature (see, e.g., [2]), it is assumed to be measurable. According to the LPV nature of the model, the coefficients a_i and b_j are seen as functions of the scheduling signal p and they are intended to describe the varying linear dynamical relation between u and y . For clarity of exposition, in this paper, the model coefficients a_i and b_j are considered to be differentiable and to have only static dependence on p , i.e., a_i and b_j only depend on the value of p at time k . Note that in parametric identification of LPV systems, the dependence of the coefficient functions a_i and b_j on the scheduling parameter p is commonly parameterized in terms of a linear combination of a finite number of *a priori* chosen basis functions (linear basis, monomials, etc.) in the variable $p(k)$, see e.g., [7], [8]. On the other hand, in this contribution, the dependence of a_i and b_j on the scheduling variable p is not *a priori* parameterized.

Rewrite the model structure (1) as

$$y(k) = \sum_{i=1}^{n_g} a_i(p(k))x_i(k) + e(k), \quad (2)$$

where $x_i(k)$ denotes the i -th component of the $n_g = n_a + n_b + 1$ -dimensional vector $x(k)$ defined as

$x(k) = [y(k-1) \dots y(k-n_a) \ u(k) \dots u(k-n_b)]^\top$ and, with an abuse of notation, $\{a_i(\cdot)\}_{i=1}^{n_g}$ stands for all the associated coefficient functions, according to (1). Similar to the parametric case, the coefficient functions a_i are considered to have a hypothetical parameterization

$$a_i(\cdot) = \omega_i^\top \phi_i(\cdot), \quad (3)$$

where each ϕ_i is a nonlinear map from the scheduling parameter space \mathbb{P} to a high-dimensional space, the so-called feature space \mathbb{R}^{n_H} . ω_i is an n_H -dimensional vector

of constant parameters. In contrast with the parametric case, the structural dependence of the coefficients ϕ_i on p is assumed to be *a priori* unknown and the dimension n_H of the parameter vector ω_i is potentially infinite. One of the originality of this work with respect to usual assumptions is to consider that $a_i(\cdot)$ might depend only on a dimensional subset of $p(k)$. Should the elements of $p(k)$ correspond to delayed version of a one dimensional scheduling variable, the presented approach can be seen as a way to estimate parsimoniously dynamical or delayed dependencies. Should the elements of $p(k)$ be different scheduling variables, the aim would be to determine which dependencies best suit for each nonlinear dependency or whether the coefficients are scheduling dependent at all.

B. Identification problem

The goal is to estimate the dependence of the coefficient functions $a_i(\cdot)$ on the scheduling variable p accurately, but, at the same time, minimize the “complexity” of these estimated functions till an appropriate trade off between the prediction error (fit w.r.t. the data) and variation of the functions over \mathbb{P} is found. The latter corresponds to a particular bias/variance trade off. This is done by solving an optimization problem using a set of N data points $\mathcal{D}_N = \{(u(k), p(k), y(k))\}_{k=1}^N$, generated by the system to be identified. This system is assumed to lay inside of the model set corresponding to (1).

Like in non-regularized LS-SVM identification approaches, e.g., [3], [9], [10], the optimization problem uses a cost function penalizing the 2-norm of the parameter vectors ω_i and the residual $e(k) = y(k) - \sum_{i=1}^{n_g} \omega_i^\top \phi_i(p(k))x_i(k)$ corresponding to the one-step-ahead prediction error by (1). Because we aim at estimating the dependence of ϕ_i on the scheduling variable p with a minimal complexity, which translates to the expectation of a smooth variation of each a_i over \mathbb{P} , an additional term to penalize the norm of the sensitivity of each ϕ_i towards each dimension of p is introduced. This results in the following identification criterion, i.e., cost function:

$$\mathcal{W}(\omega, e) = \frac{1}{2} \|\omega\|_{\ell_2}^2 + \frac{\gamma_1}{2} \|e(k)\|_{\ell_2}^2 + \gamma_2 \sum_{i=1}^{n_g} \sum_{j=1}^{n_p} \left\| \frac{\partial \omega_i^\top \phi_i(p)}{\partial p_j} \right\|_{\ell_\infty} \quad (4)$$

where $\gamma_1 > 0$ and $\gamma_2 > 0$ represent the tuning hyperparameters specified by the user, commonly required in regularized optimization.

Due to the differentiability assumption, the norm penalizing the sensitivity can be written as

$$\left\| \frac{\partial \omega_i^\top \phi_i(p)}{\partial p_j} \right\|_{\ell_\infty} = \sup_{p \in \mathbb{P}} \left| \frac{\partial \omega_i^\top \phi_i(p)}{\partial p_j} \right|. \quad (5)$$

Let us define the index set $\mathbb{I}_s^\tau = \{z \in \mathbb{Z} \mid s \leq z \leq \tau\}$ and the set $\mathcal{P} = \{[\bar{p}_1^{(s_1)} \dots \bar{p}_{n_p}^{(s_{n_p})}]\} \in \mathbb{P} \mid \text{with } s_1 \in \mathbb{I}_1^{n_s}, \dots, s_{n_p} \in \mathbb{I}_1^{n_s}\}$ which corresponds to an (equidistant) gridding of \mathbb{P} with n_s points along each dimension. Using these grid points, we can calculate a lower-bound approxi-

mation of the sensitivity norm:

$$\sup_{p \in \mathbb{P}} \left| \frac{\partial \omega_i^\top \phi_i(p)}{\partial p_j} \right| \geq \max_{\bar{p} \in \mathcal{P}} \left| \frac{\partial \omega_i^\top \phi_i(p)}{\partial p_j} \right|_{p=\bar{p}}. \quad (6)$$

By introducing the slack variables $\{r_{i,j}\}_{i=1,j=1}^{n_g, n_p}$, we can formulate the identification problem in terms of a quadratic program:

$$\min_{\omega, r, e} \frac{1}{2} \sum_{i=1}^{n_g} \omega_i^\top \omega_i + \frac{\gamma_1}{2} \sum_{k=1}^N e^2(k) + \gamma_2 \sum_{i=1}^{n_g} \sum_{j=1}^{n_p} r_{i,j} \quad (7a)$$

$$\text{s.t. } e(k) = y(k) - \sum_{i=1}^{n_g} \omega_i^\top \phi_i(p(k)) x_i(k), \quad k \in \mathbb{I}_1^N \quad (7b)$$

$$-r_{i,j} \leq \left. \frac{\partial \omega_i^\top \phi_i(p)}{\partial p_j} \right|_{p=\bar{p}} \leq r_{i,j}, \quad (7c)$$

$$i \in \mathbb{I}_1^{n_g}, j \in \mathbb{I}_1^{n_p}, \bar{p} \in \mathcal{P}$$

Note that the parameters ω_i in (7) cannot be computed since it would require an explicit representation of the feature maps ϕ_i .

C. Dual formulation

Since we cannot estimate both the parameters and the feature maps together, the dual formulation of (7) is considered. For this purpose, let us define the *Lagrangian* as

$$\begin{aligned} \mathcal{L}(\omega, e, r, \alpha, \beta^+, \beta^-) = & \frac{1}{2} \sum_{i=1}^{n_g} \omega_i^\top \omega_i + \frac{\gamma_1}{2} \sum_{k=1}^N e^2(k) + \gamma_2 \sum_{i=1}^{n_g} \sum_{j=1}^{n_p} r_{i,j} \\ & - \sum_{k=1}^N \alpha_k \left(e(k) - y(k) + \sum_{i=1}^{n_g} \omega_i^\top \phi_i(p(k)) x_i(k) \right) \\ & - \sum_{i=1}^{n_g} \sum_{j=1}^{n_p} \sum_{m=1}^M \beta_{i,j,m}^+ \left(r_{i,j} - \frac{\partial \omega_i^\top \phi_i(\bar{p}_m)}{\partial p_j} \right) \\ & - \sum_{i=1}^{n_g} \sum_{j=1}^{n_p} \sum_{m=1}^M \beta_{i,j,m}^- \left(r_{i,j} + \frac{\partial \omega_i^\top \phi_i(\bar{p}_m)}{\partial p_j} \right). \end{aligned} \quad (8)$$

Here $\alpha = \{\alpha_k\}_{k=1}^N$, $\beta^+ = \{\beta_{i,j,m}^+\}_{i=1,j=1,m=1}^{n_g, n_p, M}$ and $\beta^- = \{\beta_{i,j,m}^-\}_{i=1,j=1,m=1}^{n_g, n_p, M}$ are the Lagrangian multipliers and $\mathcal{P} = \{\bar{p}_m\}_{m=1}^M$ is a set containing all $(M = n_p \times n_s)$ grid points.

The term $\frac{\partial \omega_i^\top \phi_i(\bar{p}_m)}{\partial p_j}$ has been used as a short notation for $\left. \frac{\partial \omega_i^\top \phi_i(p)}{\partial p_j} \right|_{p=\bar{p}_m}$.

Next, the Lagrange dual function $g(\alpha, \beta^+, \beta^-)$ is defined as the infimum of $\mathcal{L}(\omega, e, r, \alpha, \beta^+, \beta^-)$ over (ω, e, r) , i.e.,

$$g(\alpha, \beta^+, \beta^-) = \inf_{\omega, e, r} \mathcal{L}(\omega, e, r, \alpha, \beta^+, \beta^-). \quad (9)$$

Since $\mathcal{L}(\omega, e, r, \alpha, \beta^+, \beta^-)$ is a convex quadratic function of (ω, e, r) , the infimum of the Lagrange dual function $g(\alpha, \beta^+, \beta^-)$ can be computed from the (necessary and

sufficient) optimality conditions:

$$\frac{\partial \mathcal{L}}{\partial \omega_i} = 0 \rightarrow \omega_i = \sum_{k=1}^N \alpha_k \phi_i(p(k)) x_i(k) \quad (10a)$$

$$+ \sum_{j=1}^{n_p} \sum_{m=1}^M \frac{\partial \phi_i(\bar{p}_m)}{\partial p_j} (\beta_{i,j,m}^- - \beta_{i,j,m}^+),$$

$$\frac{\partial \mathcal{L}}{\partial e} = 0 \rightarrow \alpha_k = \gamma_1 e(k), \quad (10b)$$

$$\frac{\partial \mathcal{L}}{\partial r_{i,j}} = 0 \rightarrow \gamma_2 = \sum_{m=1}^M (\beta_{i,j,m}^- + \beta_{i,j,m}^+). \quad (10c)$$

Here, condition (10a) needs to be satisfied for all $i \in \mathbb{I}_1^{n_g}$, (10b) for all $k \in \mathbb{I}_1^N$ and condition (10c) for all $i \in \mathbb{I}_1^{n_g}$ and $j \in \mathbb{I}_1^{n_p}$. Substitution of the optimality conditions (10a)-(10c) into the Lagrangian (8) gives the Lagrange dual function as given in (11).

In (11), X_i is a diagonal matrix whose diagonal entries are $x_i(1), \dots, x_i(N)$. $\Phi_i^{(N)}, \Delta_i^{(M \times n_p)}, \beta^+, \beta^-$ are defined as

$$\Delta_i^{(M \times n_p)} = \begin{bmatrix} \frac{\partial \phi_i(\bar{p}_1)}{\partial p_1} & \dots & \frac{\partial \phi_i(\bar{p}_1)}{\partial p_{n_p}} & \dots \\ \dots & \frac{\partial \phi_i(\bar{p}_M)}{\partial p_1} & \dots & \frac{\partial \phi_i(\bar{p}_M)}{\partial p_{n_p}} \end{bmatrix} \quad (12a)$$

and

$$\Phi_i^{(N)} = [\phi_i(1) \quad \phi_i(2) \quad \dots \quad \phi_i(N)], \quad (12b)$$

$$\beta^+ = [\beta_{i,1,1}^+ \quad \dots \quad \beta_{i,n_p,1}^+ \quad \beta_{i,1,2}^+ \quad \dots \quad \beta_{i,n_p,M}^+]^\top, \quad (12c)$$

$$\beta^- = [\beta_{i,1,1}^- \quad \dots \quad \beta_{i,n_p,1}^- \quad \beta_{i,1,2}^- \quad \dots \quad \beta_{i,n_p,M}^-]^\top. \quad (12d)$$

The Lagrange dual problem associated with (7) is now given by the optimization problem:

$$\min_{\alpha, \beta^+, \beta^-} -g(\alpha, \beta^+, \beta^-) \quad (13a)$$

$$\text{s.t. } \gamma_2 = \sum_{m=1}^M (\beta_{i,j,m}^- + \beta_{i,j,m}^+) \quad (13b)$$

$$\beta_{i,j,s,1,m}^+, \beta_{i,j,m}^- \geq 0, \quad i \in \mathbb{I}_1^{n_g}, j \in \mathbb{I}_1^{n_p}, m \in \mathbb{I}_1^M \quad (13c)$$

Note that (13) is a convex optimization problem since the Lagrange dual function $g(\alpha, \beta^+, \beta^-)$ is concave. In fact, the matrix

$$\Omega_i = \begin{bmatrix} X_i \left(\Phi_i^{(N)} \right)^\top \Phi_i^{(N)} X_i & X_i \left(\Phi_i^{(N)} \right)^\top \Delta_i^{(M \times n_p)} \\ \left(\Delta_i^{(M \times n_p)} \right)^\top \Phi_i^{(N)} X_i & \left(\Delta_i^{(M \times n_p)} \right)^\top \Delta_i^{(M \times n_p)} \end{bmatrix} \quad (14)$$

appearing in (11) is positive semidefinite since it is defined by the product

$$\Omega_i = \begin{bmatrix} X_i \left(\Phi_i^{(N)} \right)^\top \\ \left(\Delta_i^{(M \times n_p)} \right)^\top \end{bmatrix} \begin{bmatrix} \Phi_i^{(N)} X_i & \Delta_i^{(M \times n_p)} \end{bmatrix}. \quad (15)$$

$$g(\alpha, \beta^+, \beta^-) = -\frac{1}{2\gamma_1} \alpha^\top \alpha + \alpha^\top Y - \frac{1}{2} \sum_{i=1}^{n_g} \left[\beta^- - \beta^+ \right]^\top \begin{bmatrix} X_i \left(\Phi_i^{(N)} \right)^\top \Phi_i^{(N)} X_i & X_i \left(\Phi_i^{(N)} \right)^\top \Delta_i^{(M \times n_p)} \\ \left(\Delta_i^{(M \times n_p)} \right)^\top \Phi_i^{(N)} X_i & \left(\Delta_i^{(M \times n_p)} \right)^\top \Delta_i^{(M \times n_p)} \end{bmatrix} \begin{bmatrix} \alpha \\ \beta^- - \beta^+ \end{bmatrix} \quad (11)$$

D. Estimation of the coefficient functions

Since the feature maps ϕ are assumed to be *a priori* unknown, the parameter vector ω_i can not be explicitly computed via (10a). On the other hand, thanks to the structure of the Lagrange dual function (11), the dual problem can be solved without a prior specification of the feature maps defining the matrices $\Phi_i^{(N)}$ and $\Delta_i^{(M \times n_p)}$. Let us first rewrite the matrix Ω_i in the block form

$$\begin{aligned} \Omega_i &= \begin{bmatrix} X_i \left(\Phi_i^{(N)} \right)^\top \Phi_i^{(N)} X_i & X_i \left(\Phi_i^{(N)} \right)^\top \Delta_i^{(M \times n_p)} \\ \left(\Delta_i^{(M \times n_p)} \right)^\top \Phi_i^{(N)} X_i & \left(\Delta_i^{(M \times n_p)} \right)^\top \Delta_i^{(M \times n_p)} \end{bmatrix} \\ &= \begin{bmatrix} \Omega_i^{(N, N)} & \Omega_i^{(N, M \times n_p)} \\ \left(\Omega_i^{(N, M \times n_p)} \right)^\top & \Omega_i^{(M \times n_p, M \times n_p)} \end{bmatrix}. \end{aligned} \quad (16)$$

The entries of $\Omega_i^{(N, N)}$ are given by

$$\begin{aligned} \left[\Omega_i^{(N, N)} \right]_{j, k} &= x(j) \phi_i^\top(p(j)) \phi_i(p(k)) x(k) \\ &= x(j) K_i(p(j), p(k)) x(k), \end{aligned} \quad (17)$$

where K_i is a positive definite kernel function defining the inner products of $\phi_i^\top(j) \phi_i(k)$. Consequently, K_i defines Ω_i and hence characterizes the feature maps $\{\phi_i\}_{i=1}^{n_g}$ in an efficient fashion. This allows to characterize a wide range of nonlinear dependencies as a linear combination of infinitely many functions ($n_H = \infty$) defined through the choice of the particular inner product and a relatively low dimensional parameter α . This approach is called the *kernel trick* [11], [12] and in fact it corresponds to function estimator based on the *reproducing kernel Hilbert space* theory [13].

The matrix $\Omega_i^{(N, M \times n_p)}$ is defined as

$$\Omega_i^{(N, M \times n_p)} = [J_i(\bar{p}_1) \quad J_i(\bar{p}_2) \quad \dots \quad J_i(\bar{p}_M)], \quad (18)$$

where the entries of $J_i(\bar{p})$ are given by

$$\left[J_i^{(N, n_p)}(\bar{p}) \right]_{j, k} = x(j) \phi_i^\top(p(j)) \frac{\partial \phi(\bar{p})}{\partial p_k} = x(j) \frac{\partial K_i(p(j), \bar{p})}{\partial p_k}. \quad (19)$$

Finally, the matrix $\Omega_i^{(M \times n_p, M \times n_p)}$ is defined as

$$\begin{aligned} \Omega_i^{(M \times n_p, M \times n_p)} &= \\ &= \begin{bmatrix} H_i(\bar{p}_1, \bar{p}_1) & H_i(\bar{p}_2, \bar{p}_1) & \dots & H_i(\bar{p}_M, \bar{p}_1) \\ H_i(\bar{p}_1, \bar{p}_2) & H_i(\bar{p}_2, \bar{p}_2) & \dots & H_i(\bar{p}_M, \bar{p}_2) \\ \vdots & \vdots & \ddots & \vdots \\ H_i(\bar{p}_1, \bar{p}_M) & H_i(\bar{p}_2, \bar{p}_M) & \dots & H_i(\bar{p}_M, \bar{p}_M) \end{bmatrix}, \end{aligned} \quad (20)$$

where entries of $H_i(\bar{p}_1, \bar{p}_2)$ are given by

$$\left[H_i^{(n_p, n_p)}(\bar{p}_1, \bar{p}_2) \right]_{j, k} = \frac{\partial \phi^\top(\bar{p}_1)}{\partial p_j} \frac{\partial \phi(\bar{p}_2)}{\partial p_k} = \frac{\partial K_i(\bar{p}_1, \bar{p}_2)}{\partial p_j \partial p_k}. \quad (21)$$

It can be noticed that the Kernel not only characterizes the feature maps, but is also defines their partial derivatives w.r.t. to the scheduling variables.

For example, a typical choice of kernel is the *radial basis function* (RBF) kernel:

$$K_i(p(j), p(k)) = \exp\left(-\frac{\|p(j) - p(k)\|_{\ell_2}^2}{\sigma_i^2}\right). \quad (22)$$

Using RBFs as the kernel in the simplest case where $n_p = 1$, the derivatives in (19) and (21) become:

$$\frac{\partial K_i(s_1, s_2)}{\partial s_2} = \frac{2(s_1 - s_2)}{\sigma_i^2} K_i(s_1, s_2), \quad (23a)$$

$$\frac{\partial^2 K_i(s_1, s_2)}{\partial s_1 \partial s_2} = \left[\frac{2}{\sigma_i^2} - \frac{4(s_1 - s_2)^2}{\sigma_i^4} \right] K_i(s_1, s_2). \quad (23b)$$

Note that other choice of kernel functions supply other partial derivatives, while in the RBF case, these partial derivatives corresponds to the first and second moments of a multidimensional Gaussian distribution.

With the use of this kernel trick, the Lagrangian multipliers α, β^+, β^- can be computed through the dual problem (13) without specifying the feature maps ϕ_i . When we combine the optimality condition (10a) and the definition of the kernel function, we can use these Lagrangian multipliers to estimate the coefficient functions:

$$\begin{aligned} \alpha_i(\cdot) &= \phi_i(\cdot)^\top \omega_i = \sum_{k=1}^N \alpha_k K_i(\cdot, p(k)) x_i(k) \\ &+ \sum_{j=1}^{n_p} \sum_{m=1}^M \frac{\partial K_i(\cdot, \bar{p}_m)}{\partial p_j} (\beta_{i, j, m}^- - \beta_{i, j, m}^+). \end{aligned} \quad (24)$$

With these estimated coefficient functions, it becomes possible to find the underlying dependency structure of the data generating process. Because of the regulation term, all coefficient functions without a dependency on a particular dimension of p will be forced to zero along that dimension. By using a threshold value on these relationships, these dependency structures give valuable knowledge that can be further used in identification steps. Furthermore, due to the shrinkage on the variation of the estimated coefficient functions, increasing the γ_2 hyper-parameter will directly result in enforcing a smoother estimate of the dependencies wherever it is possible without significantly increasing the ℓ_2 loss on the prediction. This results in a gradual decrease of the complexity of the estimated functions.

III. SIMULATION EXAMPLE

The performance of the approach w.r.t. recovering the structural dependence of an LPV system is shown in this section by means of a Monte-Carlo study based simulation example. In this study, we assume that $p(k)$ is scalar (i.e., $n_p = 1$).

The considered asymptotically stable LPV data-generating system is described by the difference equation

$$y(k) = a_1(p(k))y(k-1) + a_2(p(k))y(k-2) + a_5(p(k))y(k-5) + a_8(p(k))y(k-8) + b_2(p(k))u(k-2) + b_7(p(k))u(k-7) + e(k), \quad (25)$$

where $e(k)$ is a zero-mean white Gaussian noise and $u(k)$ is the input signal. The coefficients functions are described by the (nonlinear) functions:

$$a_1(p(k)) = -0.2, \quad (26a)$$

$$a_2(p(k)) = 0.2 + 0.22 \sin\left(\frac{2}{\pi}p(k)\right), \quad (26b)$$

$$a_5(p(k)) = 0.15, \quad (26c)$$

$$a_8(p(k)) = \begin{cases} -0.4 & \text{if } p(k) > 0.4; \\ -p(k) & \text{if } -0.4 \leq p(k) \leq 0.4; \\ 0.5 & \text{if } p(k) < -0.4; \end{cases} \quad (26d)$$

$$b_2(p(k)) = 0.5, \quad (26e)$$

$$b_7(p(k)) = \exp(-p(k)). \quad (26f)$$

Note that a_1, a_5 and b_2 do not have a dependency on the scheduling variable p .

The system described in (25) is estimated from a data set $\mathcal{D}_N = \{(u(k), p(k), y(k))\}_{k=1}^N$ with $N = 500$ input, output, and scheduling-variable measurements. To gather data, the input and the scheduling trajectories have been chosen to be white-noise sequences, independent of each other, and with a uniform distribution $\mathcal{U}(-1, 1)$. In order to provide representative results, a Monte-Carlo simulation of 100 runs is performed. At each run, new realizations of the noise, of the input and of the scheduling variable are considered. The average of the *signal-to-noise ratio* (SNR) over the Monte-Carlo simulation is equal to 23 dB, where the SNR is defined, at each run, as

$$10 \log_{10} \left(\frac{\sum_{k=1}^N w(k)^2}{\sum_{k=1}^N e_0(k)^2} \right), \quad (27)$$

with $w(k)$ denoting the noise-free system output.

The following LPV model structure is used to estimate the data-generating system:

$$y(k) = \sum_{i=1}^{n_a} a_i(p(k))y(k-i) + \sum_{j=1}^{n_b} b_j(p(k))u(k-j), \quad (28)$$

with $n_a = n_b = 10$. In the model structure, the dependencies of $a_i(p(k))$ and $b_j(p(k))$ on the scheduling variable p are not specified.

The coefficient functions $a_i(p(k))$ are estimated in two ways: by means of a non-regularized LS-SVM approach,

corresponding to the minimization of the objective function (4) for $\gamma_2 = 0$ and by using the *regularized* LS-SVM (RLS-SVM) approach described in this paper. In both approaches, radial basis functions are used as kernels to define the inner product among the feature maps. The values of the parameters γ_1 and γ_2 are chosen via cross-validation based grid search, independently for each method. The interval $\mathbb{P} = [-1 \ 1]$ is gridded into 13 equidistant nodes $\mathcal{P} = \{\bar{p}_m\}_{m=1}^{13}$.

To compare the regularized LS-SVM to the non-regularized method, the average (over the Monte Carlo simulation) of the estimated peak to peak values of the coefficient functions is computed. These values are reported in Table I. Here we see that the RLS-SVM method is able to estimate the coefficient functions with a reduced bias and variance in comparison with the LS-SVM method. The small mean values of the non-dependent coefficient functions show that it is possible to extract information about the dependency structure of the underlying process. This can be further exploited via thresholding based re-estimation.

In order to visualize the difference between the LS-SVM and RLS-SVM methods, the difference between the true coefficient functions and the coefficient functions estimated with the two methods is plotted in Fig. 1, together with the standard deviation boundaries. The estimated values are the mean values of the 100 Monte-Carlo runs. The error reduction by using the RLS-SVM is clearly visible and it can also be seen that the recovered coefficient functions a_1, a_5 and b_2 do not have any dependency on p . This knowledge could be further used in order to better parameterize the model and hence reduce the variance.

IV. CONCLUSIONS

In this paper, we have addressed the problem of identifying LPV-ARX models in the LS-SVM framework, where the underlying dependence of the coefficient functions on the scheduling variable is not *a priori* known. By penalizing the sensitivity of the coefficient functions w.r.t. the scheduling variables, the method provides a way to estimate the dependency structure of the plant without exploiting any *a priori* information on the underlying behavior of the true LPV system. The reported simulation has shown the effectiveness of the proposed approach in providing a lower-variance estimate (w.r.t. to standard LS-SVM methods) of the coefficient functions describing the LPV model.

REFERENCES

- [1] J. S. Shamma and M. Athans, "Analysis of gain scheduled control for nonlinear plants," *IEEE Trans. on Automatic Control*, vol. 35, no. 8, pp. 898–907, 1990.
- [2] R. Tóth, *Modeling and Identification of Linear Parameter-Varying Systems*, ser. Lecture Notes in Control and Information Sciences, Vol. 403. Heidelberg: Springer, 2010.
- [3] J. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. World Scientific, 2002.
- [4] R. Tóth, V. Laurain, W. Zheng, and K. Poolla, "Model structure learning: A support vector machine approach for LPV linear-regression models," in *Proc. of the 50th IEEE Conf. on Decision and Control*, Orlando, Florida, USA, Dec. 2011, pp. 3192–3197.

TABLE I

AVERAGE AND VARIANCE (OVER 100 MONTE-CARLO RUNS) OF THE MAXIMUM PEAK TO PEAK VALUE OF THE COEFFICIENT FUNCTIONS $a_i(p(k))$ AND $b_j(p(k))$ ESTIMATED BY MEANS OF THE LS-SVM AND RLS-SVM APPROACH.

True value	\bar{a}_1 0	\bar{a}_2 0.22	\bar{a}_3 0	\bar{a}_4 0	\bar{a}_5 0	\bar{a}_6 0	\bar{a}_7 0	\bar{a}_8 0.9	\bar{a}_9 0	\bar{a}_{10} 0
Mean (RLS-SVM)	$2.1 \cdot 10^{-5}$	$2.0 \cdot 10^{-1}$	$2.2 \cdot 10^{-3}$	$3.8 \cdot 10^{-8}$	$8.6 \cdot 10^{-4}$	$2.2 \cdot 10^{-8}$	$3.1 \cdot 10^{-4}$	$8.70 \cdot 10^{-1}$	$3.7 \cdot 10^{-4}$	$2.7 \cdot 10^{-8}$
Mean (LS-SVM)	$8.8 \cdot 10^{-2}$	$1.7 \cdot 10^{-1}$	$9.4 \cdot 10^{-2}$	$4.8 \cdot 10^{-2}$	$1.4 \cdot 10^{-1}$	$6.6 \cdot 10^{-2}$	$7.3 \cdot 10^{-2}$	$5.4 \cdot 10^{-1}$	$5.8 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$
Var (RLS-SVM)	$5.8 \cdot 10^{-10}$	$8.4 \cdot 10^{-4}$	$4.2 \cdot 10^{-5}$	$2.7 \cdot 10^{-15}$	$1.2 \cdot 10^{-5}$	$2.9 \cdot 10^{-16}$	$2.0 \cdot 10^{-6}$	$8.4 \cdot 10^{-4}$	$7.1 \cdot 10^{-6}$	$7.1 \cdot 10^{-16}$
Var (LS-SVM)	$1.1 \cdot 10^{-3}$	$8.7 \cdot 10^{-4}$	$1.3 \cdot 10^{-3}$	$8.5 \cdot 10^{-4}$	$1.5 \cdot 10^{-3}$	$8.9 \cdot 10^{-4}$	$8.6 \cdot 10^{-4}$	$2.4 \cdot 10^{-3}$	$9.3 \cdot 10^{-4}$	$5.8 \cdot 10^{-4}$

True value	\bar{b}_1 0	\bar{b}_2 0	\bar{b}_3 0	\bar{b}_4 0	\bar{b}_5 0	\bar{b}_6 0	\bar{b}_7 2.35	\bar{b}_8 0	\bar{b}_9 0	\bar{b}_{10} 0
Mean (RLS-SVM)	$2.2 \cdot 10^{-8}$	$3.8 \cdot 10^{-5}$	$2.4 \cdot 10^{-8}$	$3.1 \cdot 10^{-8}$	$2.6 \cdot 10^{-8}$	$2.3 \cdot 10^{-8}$	$1.8 \cdot 10^0$	$4.1 \cdot 10^{-8}$	$6.6 \cdot 10^{-8}$	$7.2 \cdot 10^{-8}$
Mean (LS-SVM)	$4.6 \cdot 10^{-2}$	$2.1 \cdot 10^{-1}$	$4.3 \cdot 10^{-2}$	$4.0 \cdot 10^{-2}$	$4.5 \cdot 10^{-2}$	$4.4 \cdot 10^{-2}$	$8.5 \cdot 10^{-1}$	$4.1 \cdot 10^{-2}$	$5.8 \cdot 10^{-2}$	$4.6 \cdot 10^{-2}$
Var (RLS-SVM)	$4.5 \cdot 10^{-16}$	$3.3 \cdot 10^{-10}$	$6.6 \cdot 10^{-16}$	$7.1 \cdot 10^{-16}$	$6.0 \cdot 10^{-16}$	$3.4 \cdot 10^{-16}$	$5.3 \cdot 10^{-3}$	$2.2 \cdot 10^{-15}$	$3.0 \cdot 10^{-15}$	$4.1 \cdot 10^{-15}$
Var (LS-SVM)	$8.7 \cdot 10^{-4}$	$6.9 \cdot 10^{-4}$	$7.6 \cdot 10^{-4}$	$5.8 \cdot 10^{-4}$	$9.4 \cdot 10^{-4}$	$6.4 \cdot 10^{-4}$	$3.8 \cdot 10^{-3}$	$4.3 \cdot 10^{-4}$	$9.7 \cdot 10^{-4}$	$7.5 \cdot 10^{-4}$

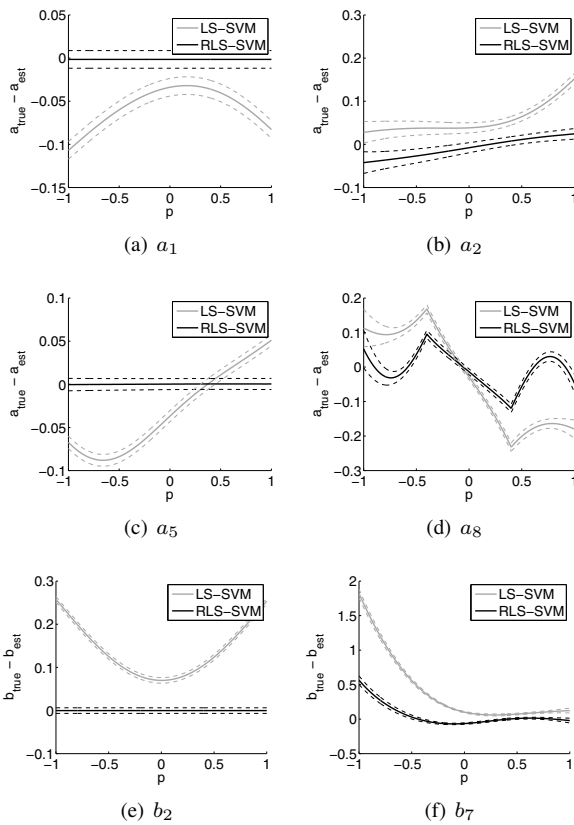


Fig. 1. Difference between the true and the estimated coefficient functions together with the standard deviation boundaries.

- [5] V. Laurain, R. Tóth, W. Zheng, and M. Gilson, "Nonparametric identification of LPV models under general noise conditions. An LS-SVM based approach," in *Proc. of the 16th IFAC Symposium on System Identification*, Brussels, Belgium, July 2012, pp. 1761–1766.
- [6] D. Piga and R. Tóth, "LPV model order selection in an LS-SVM setting," in *Proc. of the 52nd IEEE Conference on Decision and Control*, Florence, Italy, Dec. 2013, pp. 4128–4133.
- [7] L. Giarré, D. Bauso, P. Falugi, and B. Bamieh, "LPV model identification for gain scheduling control: An application to rotating stall and

surge control problem," *Control Engineering Practice*, vol. 14, no. 4, pp. 351–361, 2006.

- [8] J. W. van Wingerden and M. Verhaegen, "Subspace identification of bilinear and LPV systems for open- and closed-loop data," *Automatica*, vol. 45, no. 2, pp. 372–381, 2009.
- [9] T. Falck, K. Pelckmans, J. Suykens, and B. De Moor, "Identification of wiener-hammerstein systems using LS-SVMs," in *15th IFAC symposium on System Identification*, Saint Malo, France, July 2009.
- [10] I. Goethals, K. Pelckmans, J. Suykens, and B. De Moor, "Identification of MIMO Hammerstein models using least squares support vector machines," *Automatica*, vol. 41, no. 7, pp. 1263–1272, 2005.
- [11] B. Schölkopf and A. Smola, *Learning with kernels*. Cambridge MA: MIT Press, 2002.
- [12] V. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.
- [13] F. Cucker and S. Smale, "On the mathematical foundations of learning," *Bulletin of the American Mathematical Society*, vol. 39, no. 1, pp. 1–49, 2001.