

Identification of continuous-time models with slowly time-varying parameters

A. Padilla^a, H. Garnier^b, P. C. Young^c, F. Chen^d, J. I. Yuz^e

^a*Universidad de La Frontera, Department of Mechanical Engineering, Francisco Salazar 01145, Temuco, Chile*

^b*Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France*

^c*Systems and Control Group, Lancaster Environment Centre, Lancaster University, Lancaster, UK*

^d*Wuhan University, Department of Automation, Wuhan, China*

^e*Universidad Técnica Federico Santa María, Department of Electronic Engineering, Valparaiso, Chile*

Abstract

The off-line estimation of the parameters of continuous-time, linear, time-invariant transfer function models can be achieved straightforwardly using linear prefilters on the measured input and output of the system. The on-line estimation of continuous-time models with time-varying parameters is less straightforward because it requires the updating of the continuous-time prefilter parameters. This paper shows how such on-line estimation is possible by using recursive instrumental variable approaches. The proposed methods are presented in detail and also evaluated on a numerical example using both single experiment and Monte Carlo simulation analysis. In addition, the proposed recursive algorithms are tested using data from two real-life systems.

Keywords: Recursive methods; continuous-time model identification; instrumental variable method; linear time-varying system; linear filter methods.

*Corresponding author

Email address: arturo.padilla@ufrontera.cl (A. Padilla)

1. Introduction

A time-varying model can be defined as a model whose properties vary in time, as reflected by changes in the model parameters. In real-life, processes are usually time-varying, for example, due to ageing effects or changes in the environmental conditions such as the temperature. The estimated parameters of a time-varying model can vary in different ways such as slow persistent changes (sometimes called parameter drift), infrequent abrupt changes, mixed-mode variations (slow and abrupt changes), and all other changes (also called fast parameter variations) [1, p. 60]. In this paper, we consider continuous-time (CT) linear time-varying (LTV) models for which the parameter variation is slow compared to the system dynamics.

It is well known that batch estimation, also called non-recursive or off-line identification, can be used for the estimation of constant parameter discrete-time (DT) or CT LTV models. In the case of systems whose dynamic behaviour is changing very slowly, their parameters can be updated from time-to-time, for example, estimating ‘local’ linear time-invariant (LTI) models through weighted least squares in a sliding window. In [1], two approaches are presented: sliding window least squares, which considers a rectangular window, meaning that all the weights are equal, and also an exponentially weighted least squares, where old data are discarded by means of exponential weights. On the other hand, if the system is changing more rapidly, however, improved approaches are necessary. Sometimes, prior knowledge is such that deterministic or near deterministic models for the parameter variations can be used, and then estimation with a batch sliding window can be performed. Such a technique is explored for instance in [2] by means of local polynomial modeling. The current paper is concerned with the general situation where the nature of the parameter variation is slow but unknown.

The problem of LTV estimation for DT models has been extensively studied in the literature. In this context, recursive estimation algorithms for LTI models are usually modified to allow for parameter variation. In this approach, also called on-line or real-time identification, the model is sequentially updated every time new measurements become available.

In comparison with batch estimation, recursive algorithms are advantageous in terms of having less computational cost for each recursive update. Nevertheless, the obtained estimates can be less statistically efficient due to fixed interval smoothing procedures, that are advantageous in the LTI situation [3], but cannot be utilised in on-line estimation. While a large number

of studies have been conducted on the real-time recursive estimation of DT models [see, e.g. 4, 5, 6, 7, 3, 8, and the references therein], comparatively less attention has been given to CT models. A simple and widely used algorithm is recursive least squares (RLS). In order to track time-varying parameters with RLS, two adaptation mechanisms can be used: introducing a *forgetting* factor; or using a recursive algorithm that is analogous to the Kalman Filter (KF) [see e.g. 7, 9, 3], where the latter KF version is more convenient when tracking parameters that are varying at different rates. Moreover, the noise-variance ratio (NVR) hyper-parameters that control the rate of variation of the estimated parameters in the recursive algorithm can be optimized by maximum likelihood [3].

Different methods are available to estimate CT LTV models. Recent studies present a frequency domain approach [10] or use a subspace method to identify MIMO systems [11]. There are also integral methods [see 12, 13], such as block-pulse functions [14], or the CT version of the recursive prediction error method [15]. Another type of CT identification method is the modulating function approach proposed in [16] and [17] for on-line estimation. Additionally, a linear filter method was first suggested by [18] and considered further in [9, 3, 19, 20], which provides the main stimulus for the present paper. In batch estimation of CT LTI models, different approaches based on the linear filter method have been used to tackle the identification of real-life systems [21, 22]. In the linear filter approach, suitably chosen linear prefilters are used to generate the filtered derivatives of the measured input and output signals that are required for CT model estimation. These prefilters can be chosen based on the system bandwidth or selected to have optimal properties. A simple linear prefilter approach is the recursive least squares state-variable filter (RLSSVF) method which is discussed later in Section 4.1.

On the other hand, identification methods based on recursive instrumental variable (RIV) can avoid the asymptotic bias that may appear in RLS estimates. To circumvent this problem in CT model estimation, the refined instrumental variable method for continuous-time systems (RIVC) or its simplified alternative (SRIVC) can be used [23, 3]. In [3], only some comments about the recursive on-line version of SRIVC (RSRIVC) are given. More recently, the RSRIVC approach, which exploits the KF version of the algorithm as an adaptation mechanism, has been developed for the estimation of CT LTV models in [24].

The identification methods for CT models can be classified between indi-

rect and direct approaches. In the former case, first a DT model is estimated and then converted to a CT model. In the latter case, the CT model is directly estimated from the available data. RLSSVF and RSRIVC correspond to direct estimation approaches. The indirect estimation of CT LTV models is considered, for example, in [25].

The current paper proposes a variant of RSRIVC and discusses a robust implementation, which includes a stability test and a filter that smooths and delays the parameter estimates before updating the prefilters and the *auxiliary* model. In addition, practical aspects of the algorithms are addressed, such as the guidelines to choose the hyper-parameters of the IV based approaches of the proposed algorithms. In order to assess the proposed methods, measures of estimation performance are also discussed and used. The proposed direct CT identification methods are also compared with indirect approaches.

The remainder of the paper is organized as follows: the CT model estimation problem is firstly presented in Section 2. In Section 3, indirect estimation approaches for CT models are briefly reviewed. Then, the recursive version of the LSSVF and SRIVC algorithms are developed in Sections 4.1 and 4.2, respectively. In Section 4.2, a variant of the recursive SRIVC is discussed. Two implementation issues are addressed in Section 5, namely, the stability of the estimates, and the correlation between the prefiltered estimates and the measurement noise. In addition, practical aspects of the proposed algorithms are discussed. In Section 6, measures of the estimator performance are presented. In Section 7, the application of the IV based approaches are illustrated through numerical examples, including Monte Carlo simulations and two examples using real data. Finally, conclusions are presented in Section 9.

2. Problem formulation

Let us consider a CT LTV OE system with input $u(t)$ and output $y(t)$. The data-generating system is given by

$$\mathcal{S} \begin{cases} A_o(p, t)x(t) = B_o(p, t)u(t) \\ y(t_k) = x(t_k) + e_o(t_k) \end{cases} \quad (1)$$

where p is the differentiation operator; t_k is the time index sample number for sampled data; and $e_o(t_k)$ is a zero-mean DT white noise sequence. It

is assumed that the system (1) can be represented by the model set

$$\mathcal{M} \begin{cases} A(p, t, \boldsymbol{\theta})x(t) = B(p, t, \boldsymbol{\theta})u(t) \\ y(t_k) = x(t_k) + e(t_k) \end{cases} \quad (2)$$

Here, $A(p, t, \boldsymbol{\theta})$ and $B(p, t, \boldsymbol{\theta})$ are the following polynomials with time-varying parameters:

$$B(p, t, \boldsymbol{\theta}) = b_0(t)p^{n_b} + b_1(t)p^{n_b-1} + \dots + b_{n_b}(t) \quad (3)$$

$$A(p, t, \boldsymbol{\theta}) = p^{n_a} + a_1(t)p^{n_a-1} + \dots + a_{n_a}(t) \quad (4)$$

where $n_a \geq n_b$ and $e(t_k)$ is a zero-mean DT white noise. The time-varying parameters can be gathered in the parameter vector $\boldsymbol{\theta}(t)$,

$$\boldsymbol{\theta}(t) = [a_1(t) \quad \dots \quad a_{n_a}(t) \quad b_0(t) \quad \dots \quad b_{n_b}(t)]^T \quad (5)$$

with $\boldsymbol{\theta}(t) \in \mathbb{R}^{n_\theta}$, and $n_\theta = n_a + n_b + 1$.

The following assumptions are made:

A1. The system and the model set have a CT LTV OE representation, with polynomial degrees n_a and n_b identified or known *a priori*.

A2. The true parameter vector $\boldsymbol{\theta}_o(t)$ is slowly time-varying in the sense that, in a local window with time interval $s \in [t_i, t_f]$, it can be locally approximated by a constant parameter vector $\boldsymbol{\theta}_c$, *i.e.*

$$\|\dot{\boldsymbol{\theta}}_o(t)\| \leq \epsilon_\theta \quad \text{such that} \quad \|\boldsymbol{\theta}_o(s) - \boldsymbol{\theta}_c\| \leq \epsilon_{\Delta\theta} \quad (6)$$

where ϵ_θ and $\epsilon_{\Delta\theta}$ are small numbers.

A3. The input $u(t)$ is persistently exciting.

The identification problem is then to recursively estimate the time-varying parameters that characterize the model structure given by (2), based on sequential samples of input and output data $Z_k = \{u(t_i); y(t_i)\}_{i=1}^k$, where k is the number of samples which increases by one with every recursion.

3. Indirect estimation of CT models

From the frozen systems of (1) and considering the appropriate intersample behaviour for the input, a DT equivalent system can be obtained. Let us consider that the equivalent DT time-varying OE system is given by

$$\mathcal{S} \begin{cases} x(t_k) + \alpha_1^o(t_k)x(t_{k-1}) + \dots + \alpha_{n_\alpha}^o(t_k)x(t_{k-n_\alpha}) = \\ \beta_1^o(t_k)u(t_{k-1}) + \dots + \beta_{n_\beta}^o(t_k)u(t_{k-n_\beta}) \\ y(t_k) = x(t_k) + e_o(t_k) \end{cases} \quad (7)$$

For the indirect identification approaches, first a DT model representing (7) is estimated. Then, the DT parameter estimates are converted to the equivalent CT parameters; in this case, the conversion is done with the MATLAB routine `d2c`. Next, two indirect identification methods are presented.

3.1. Indirect approach considering a DT ARX model

It can be assumed that (7) can be represented by the DT ARX model

$$\mathcal{M} \begin{cases} y(t_k) + \alpha_1(t_k)y(t_{k-1}) + \dots + \alpha_{n_\alpha}(t_k)y(t_{k-n_\alpha}) = \\ \beta_1(t_k)u(t_{k-1}) + \dots + \beta_{n_\beta}(t_k)u(t_{k-n_\beta}) + e(t_k) \end{cases} \quad (8)$$

Equation (8) can be written as a linear regression

$$y(t_k) = \boldsymbol{\phi}^T(t_k)\boldsymbol{\rho}(t_k) + e(t_k) \quad (9)$$

where $\boldsymbol{\phi}(t_k)$ is the regressor given by

$$\boldsymbol{\phi}^T(t_k) = [-y(t_{k-1}) \quad \dots \quad -y(t_{k-n_\alpha}) \quad u(t_{k-1}) \quad \dots \quad u(t_{k-n_\beta})] \quad (10)$$

and $\boldsymbol{\rho}(t_k)$ is the DT time-varying parameter vector given by

$$\boldsymbol{\rho}(t_k) = [\alpha_1(t_k) \quad \dots \quad \alpha_{n_\alpha}(t_k) \quad \beta_1(t_k) \quad \dots \quad \beta_{n_\beta}(t_k)]^T \quad (11)$$

To estimate (11) it is assumed that the parameter vector can be modeled as a random walk. Then, the whole model is written as follows

$$\mathcal{M} \begin{cases} \boldsymbol{\rho}(t_{k+1}) = \boldsymbol{\rho}(t_k) + \boldsymbol{w}(t_k) \\ y(t_k) = \boldsymbol{\phi}^T(t_k)\boldsymbol{\rho}(t_k) + e(t_k) \end{cases} \quad (12)$$

where $\mathbf{w}(t_k)$ is assumed to be an independent, zero-mean, white Gaussian noise model. (12) corresponds to a state-space representation and then the Kalman filter can be used to estimate $\boldsymbol{\rho}(t_k)$. The KF algorithm is given by [see e.g. 3],

Prediction step:

$$\hat{\boldsymbol{\rho}}(t_k|t_{k-1}) = \hat{\boldsymbol{\rho}}(t_{k-1}) \quad (13a)$$

$$\mathbf{P}(t_k|t_{k-1}) = \mathbf{P}(t_{k-1}) + \mathbf{Q}_n \quad (13b)$$

Correction step:

$$\hat{\boldsymbol{\rho}}(t_k) = \hat{\boldsymbol{\rho}}(t_k|t_{k-1}) + \mathbf{l}(t_k)\varepsilon(t_k) \quad (13c)$$

$$\varepsilon(t_k) = y(t_k) - \boldsymbol{\phi}^T(t_k)\hat{\boldsymbol{\rho}}(t_k|t_{k-1}) \quad (13d)$$

$$\mathbf{l}(t_k) = \frac{\mathbf{P}(t_k|t_{k-1})\boldsymbol{\phi}(t_k)}{1 + \boldsymbol{\phi}^T(t_k)\mathbf{P}(t_k|t_{k-1})\boldsymbol{\phi}(t_k)} \quad (13e)$$

$$\mathbf{P}(t_k) = \mathbf{P}(t_k|t_{k-1}) - \mathbf{l}(t_k)\boldsymbol{\phi}^T(t_k)\mathbf{P}(t_k|t_{k-1}) \quad (13f)$$

The performance of this method, in terms of tracking ability and noise sensitivity, depends on the noise-variance ratio (NVR) matrix (sometimes called ‘normalized covariance matrix’) \mathbf{Q}_n , which is defined by

$$\mathbf{Q}_n = \frac{\mathbf{Q}_w}{\sigma_e^2} \quad (14)$$

where \mathbf{Q}_w and σ_e^2 are the covariance matrix of $\mathbf{w}(t_k)$ and the variance of $e(t_k)$, respectively. The KF (13) is implemented in the MATLAB routine `recursiveARX`. Note that the Kalman filter is usually presented in terms of a matrix $\mathbf{P}^*(t_k)$, where $\mathbf{P}^*(t_k)/\sigma_e^2 = \mathbf{P}(t_k)$. However, in this paper, the normalized version of the KF is used, which is defined in terms of $\mathbf{P}(t_k)$ [3].

Remark 1. *Let us consider that the true system is defined by a random walk and a linear regression, i.e.*

$$\mathcal{S} \begin{cases} \boldsymbol{\rho}^o(t_{k+1}) = \boldsymbol{\rho}^o(t_k) + w_o(t_k) \\ y(t_k) = \boldsymbol{\phi}^T(t_k)\boldsymbol{\rho}^o(t_k) + e_o(t_k) \end{cases} \quad (15)$$

where $w_o(t_k)$ and $e_o(t_k)$ are zero-mean Gaussian noises with covariance matrix \mathbf{Q}_{w_o} and variance $\sigma_{e_o}^2$, respectively. Then, if the system belongs to the

model set ($\mathcal{S} \in \mathcal{M}$), and the KF is run with \mathbf{Q}_{w_0} and $\sigma_{e_0}^2$, the KF estimate $\hat{\boldsymbol{\rho}}(t_k)$ is optimal, in the sense that it minimizes the a posteriori parameter error covariance matrix. Note that in this case, $\mathbf{P}^*(t_k)$ is exactly the covariance matrix of the posterior distribution of $\boldsymbol{\theta}(t_k)$. If the model is not a linear regression, the KF is an ad hoc algorithm (see [6, p. 379] and [7]).

The random walk model is a way of representing that the parameters are not constant, without specifying any ‘adequate’ global model for them [1, p. 232]. As pointed out by Parkum [26, p. 63], if the formal assumptions connected with the parameters varying as a random walk model cannot be justified, the update can still be used for *ad hoc* estimation. Note that the random walk model can be used for the case where the parameters vary slowly and/or abruptly. However, the ability of an algorithm, which considers the random walk model, to track the parameters, will depend on a suitable setting of the normalized covariance matrix \mathbf{Q}_n . Actually, in a scenario where there are both slowly and abrupt parameter changes, the covariance matrix should be time-varying. Then, the problem is to detect when the jumps occur, in order to change the covariance matrix [6].

Let us recall that in an LTI identification problem, for an ARX model

$$A(q^{-1})y(t_k) = B(q^{-1})u(t_k) + e(t_k) \quad (16)$$

the prediction error is given by [7]

$$\varepsilon(t_k) = L(q^{-1})A(q^{-1}) \left(y(t_k) - \frac{B(q^{-1})}{A(q^{-1})}u(t_k) \right) \quad (17)$$

The effect of the prefilter $L(q^{-1})$ is then to cancel out the effect of an incorrect noise model. In fact, in the LTV case, by representing (7) with a time-varying ARX model means that $e(t_k)$ in (8) is not actually a white noise process. From Remark 1, it is known that KF can be used anyway as an *ad hoc* algorithm. In order to reduce the effect of a misspecified noise model in the estimation of (8), a fixed prefilter is used. Notice that such a prefilter will only partially cancel the effect of an incorrect noise model since the true system has time-varying parameters.

Once the DT parameters from (8) are estimated, they are converted to CT parameters. This approach will be called IndirectARX.

3.2. Indirect approach considering a DT OE model

To represent (7) an OE model can also be considered,

$$\mathcal{M} \begin{cases} x(t_k) + \alpha_1(t_k)x(t_{k-1}) + \dots + \alpha_{n_\alpha}(t_k)x(t_{k-n_\alpha}) = \\ \beta_1(t_k)u(t_{k-1}) + \dots + \beta_{n_\beta}(t_k)u(t_{k-n_\beta}) \\ y(t_k) = x(t_k) + e(t_k) \end{cases} \quad (18)$$

Equation (18) can be written as a pseudo-linear regression [7],

$$y(t_k) = \boldsymbol{\phi}^T(t_k, \boldsymbol{\rho})\boldsymbol{\rho}(t_k) + e(t_k) \quad (19)$$

where the regressor is

$$\boldsymbol{\phi}^T(t_k, \boldsymbol{\rho}) = [-x(t_{k-1}) \quad \dots \quad -x(t_{k-n_\alpha}) \quad u(t_{k-1}) \quad \dots \quad u(t_{k-n_\beta})] \quad (20)$$

Notice that $x(t_k)$ depends on $\boldsymbol{\rho}$, therefore the regressor depends on $\boldsymbol{\rho}$. Again it can be assumed that the parameter vector is represented by a random walk model. Then, the whole model is written as follows

$$\mathcal{M} \begin{cases} \boldsymbol{\rho}(t_{k+1}) = \boldsymbol{\rho}(t_k) + \boldsymbol{w}(t_k) \\ y(t_k) = \boldsymbol{\phi}^T(t_k, \boldsymbol{\rho})\boldsymbol{\rho}(t_k) + e(t_k) \end{cases} \quad (21)$$

Analogously, $\boldsymbol{\rho}(t_k)$ is then estimated using the Kalman filter. Such an approach is implemented in the MATLAB routine `recursiveOE`.

The algorithm that estimates (18) is initialized with the estimation of an ARX model (without any prefilter). Therefore, a new hyper-parameter $N_{s,oe}$ is introduced; $N_{s,oe}$ is the sample time at which there is a switch between the ARX and OE models. Once the DT parameters are estimated, they are converted to CT parameters. This approach will be called IndirectOE.

4. Direct estimation of CT models

4.1. Recursive least squares based algorithm

Basically, in the direct estimation of CT models through linear prefilter methods, the following two steps are required: In Step (a) a prefilter is applied to the model and a filtered linear regression is obtained; in Step (b) the DT methodology is used to estimate the parameters by applying a recursive algorithm (see Section 3).

In CT LTI system identification, time-derivatives can be handled using linear prefilter methods, *i.e.* applying a prefilter $F(p)$ to the model. However, in the LTV case, multiplication with the differentiation operator p is not commutative with the time-dependent variables, *i.e.*

$$F(p)B(p, t, \boldsymbol{\theta})u(t) \neq B(p, t, \boldsymbol{\theta})F(p)u(t)$$

To circumvent this problem, Assumption (A2) in Section 2 states that the true parameters are slowly varying in the sense that the system can be locally approximated by a constant parameter vector, *i.e.* an LTI model. Recursive estimation algorithms capable of tracking time-varying parameters have a finite memory, meaning that the data used for identification is a sliding window in which the approximation of the LTI system is valid. Then, a prefilter can be applied to obtain filtered time derivatives as follows,

$$y_f^{(i)}(t_k) = F(p)y^{(i)}(t_k) \quad i = 0, \dots, n_a \quad (22a)$$

$$u_f^{(i)}(t_k) = F(p)u^{(i)}(t_k) \quad i = 0, \dots, n_b \quad (22b)$$

or equivalently as

$$y_f^{(i)}(t_k) = p^i F(p)y(t_k) \quad i = 0, \dots, n_a \quad (23a)$$

$$u_f^{(i)}(t_k) = p^i F(p)u(t_k) \quad i = 0, \dots, n_b \quad (23b)$$

$F(p)$ is a low pass filter which can be specified in a statistically optimal form or, more simply, chosen to be in the special, suboptimal, state-variable filter (SVF) form:

$$F(p) = \frac{1}{(p + \lambda_{\text{svf}})^{n_a}} \quad (24)$$

where λ_{svf} is the cut-off frequency. Applying the SVF to (2) yields

$$\begin{aligned} y_f^{(n_a)}(t_k) + a_1(t_k)y_f^{(n_a-1)}(t_k) + \dots + a_{n_a}(t_k)y_f(t_k) = \\ b_0(t_k)u_f^{(n_b)}(t_k) + \dots + b_{n_b}(t_k)u_f(t_k) + v_f(t_k) \end{aligned} \quad (25)$$

where

$$v_f(t_k) = F(p)A(p, t, \boldsymbol{\theta})e(t_k) \quad (26)$$

Equation (25) can be rewritten as a linear regression,

$$y_f^{(n_a)}(t_k) = \boldsymbol{\varphi}_f^T(t_k)\boldsymbol{\theta} + v_f(t_k) \quad (27)$$

where

$$y_f^{(n_a)}(t_k) = F(p)p^{n_a}y(t_k) \quad (28)$$

and $\varphi_f(t_k)$ is the filtered version of the regressor vector

$$\varphi^T(t_k) = \begin{bmatrix} -y^{(n_a-1)}(t_k) & \cdots & -y(t_k) \\ u^{(n_b)}(t_k) & \cdots & u(t_k) \end{bmatrix} \quad (29)$$

i.e.

$$\begin{aligned} \varphi_f^T(t_k) &= F(p)\varphi^T(t_k) \\ &= \begin{bmatrix} -y_f^{(n_a-1)}(t_k) & \cdots & -y_f(t_k) \\ u_f^{(n_b)}(t_k) & \cdots & u_f(t_k) \end{bmatrix} \end{aligned} \quad (30)$$

As pointed out before, once the prefiltered linear regression (27) is defined, Step (b) of the linear prefilter approach can be applied. The parameter variation can be represented by a random walk model, so that the full model becomes:

$$\mathcal{M} \begin{cases} \boldsymbol{\theta}(t_{k+1}) = \boldsymbol{\theta}(t_k) + \boldsymbol{w}(t_k) \\ y_f^{(n_a)}(t_k) = \varphi_f^T(t_k)\boldsymbol{\theta}(t_k) + v_f(t_k) \end{cases} \quad (31)$$

where $\boldsymbol{w}(t_k)$ is assumed to be an independent, zero-mean, white Gaussian noise model. (31) corresponds to a state-space representation and then the Kalman filter can be used to estimate $\boldsymbol{\theta}(t_k)$. Thus, the time-varying parameters can be tracked by the following algorithm:

Prediction step:

$$\hat{\boldsymbol{\theta}}(t_k|t_{k-1}) = \hat{\boldsymbol{\theta}}(t_{k-1}) \quad (32a)$$

$$\boldsymbol{P}(t_k|t_{k-1}) = \boldsymbol{P}(t_{k-1}) + \boldsymbol{Q}_n \quad (32b)$$

Correction step:

$$\hat{\boldsymbol{\theta}}(t_k) = \hat{\boldsymbol{\theta}}(t_k|t_{k-1}) + \boldsymbol{l}(t_k)\varepsilon(t_k) \quad (32c)$$

$$\varepsilon(t_k) = y_f^{(n_a)}(t_k) - \varphi_f^T(t_k)\hat{\boldsymbol{\theta}}(t_k|t_{k-1}) \quad (32d)$$

$$\boldsymbol{l}(t_k) = \frac{\boldsymbol{P}(t_k|t_{k-1})\varphi_f(t_k)}{1 + \varphi_f^T(t_k)\boldsymbol{P}(t_k|t_{k-1})\varphi_f(t_k)} \quad (32e)$$

$$\boldsymbol{P}(t_k) = \boldsymbol{P}(t_k|t_{k-1}) - \boldsymbol{l}(t_k)\varphi_f^T(t_k)\boldsymbol{P}(t_k|t_{k-1}) \quad (32f)$$

where $y_f^{(n_a)}(t_k)$ and $\varphi_f(t_k)$ are defined in (28) and (30), respectively. The algorithm (32) will be called the recursive least squares state-variable filter (RLSSVF) method. At each recursion the output error

$$\varepsilon_y(t_k) = y(t_k) - \hat{y}(t_k) \quad (33)$$

can be also computed, where $\hat{y}(t_k)$ is the simulated model output. Here, the KF approach is considered, but it is also possible to use the forgetting factor as the adaptation mechanism. Note that $v_f(t_k)$ in (31) is a filtered noise and the conditions specified in Remark 1 are not met. Thus, RLSSVF is an *ad hoc* approach.

4.2. Recursive instrumental variable based algorithms

In the batch estimation of LTI models, it is well known that LSSVF yields biased estimates because of the correlation between $\varphi_f(t_k)$ and $v_f(t_k)$. Instrumental variable methods, such as the simplified refined instrumental variable method for continuous-time models (see Appendix A.2), can be used to obviate this problem. The recursive version of SRIVC (RSRIVC) for the estimation of CT models with slowly time-varying parameters can be developed analogously to RLSSVF by considering the following model which is in state-space form

$$\mathcal{M} \begin{cases} \boldsymbol{\theta}(t_{k+1}) = \boldsymbol{\theta}(t_k) + \boldsymbol{w}(t_k) \\ y_f^{(n_a)}(t_k) = \boldsymbol{\varphi}_f^T(t_k) \boldsymbol{\theta}(t_k) + v_f(t_k) \end{cases} \quad (34)$$

where

$$y_f^{(n_a)}(t_k) = p^{n_a} F(p, \hat{\boldsymbol{\theta}}(t_{k-1})) y(t_k) \quad (35a)$$

$$\boldsymbol{\varphi}_f^T(t_k) = F(p, \hat{\boldsymbol{\theta}}(t_{k-1})) \boldsymbol{\varphi}^T(t_k) \quad (35b)$$

$$v_f(t_k) = F(p, \hat{\boldsymbol{\theta}}(t_{k-1})) A(p, t, \boldsymbol{\theta}) e(t_k) \quad (36)$$

with the adaptive prefilter

$$F(p, \hat{\boldsymbol{\theta}}(t_{k-1})) = \frac{1}{\hat{A}(p, \hat{\boldsymbol{\theta}}(t_{k-1}))} \quad (37)$$

For the sake of simplicity in the notation, in $y_f^{(n_a)}$, φ_f and v_f the dependence on $\boldsymbol{\theta}(t_{k-1})$ is omitted. Considering the DT counterpart (see [3]), model (34)

can be estimated using the KF-based recursive instrumental variable approach. The algorithm is then given by:

Prediction step:

$$\hat{\boldsymbol{\theta}}(t_k|t_{k-1}) = \hat{\boldsymbol{\theta}}(t_{k-1}) \quad (38a)$$

$$\mathbf{P}(t_k|t_{k-1}) = \mathbf{P}(t_{k-1}) + \mathbf{Q}_n \quad (38b)$$

Correction step:

$$\hat{\boldsymbol{\theta}}(t_k) = \hat{\boldsymbol{\theta}}(t_k|t_{k-1}) + \mathbf{l}(t_k)\varepsilon(t_k) \quad (38c)$$

$$\varepsilon(t_k) = y_f^{(n_a)}(t_k) - \boldsymbol{\varphi}_f^T(t_k)\hat{\boldsymbol{\theta}}(t_k|t_{k-1}) \quad (38d)$$

$$\mathbf{l}(t_k) = \frac{\mathbf{P}(t_k|t_{k-1})\boldsymbol{\zeta}_f(t_k)}{1 + \boldsymbol{\varphi}_f^T(t_k)\mathbf{P}(t_k|t_{k-1})\boldsymbol{\zeta}_f(t_k)} \quad (38e)$$

$$\mathbf{P}(t_k) = \mathbf{P}(t_k|t_{k-1}) - \mathbf{l}(t_k)\boldsymbol{\varphi}_f^T(t_k)\mathbf{P}(t_k|t_{k-1}) \quad (38f)$$

where filtered instrument $\boldsymbol{\zeta}_f(t_k)$ is defined through the instrument

$$\boldsymbol{\zeta}(t_k, \hat{\boldsymbol{\theta}}(t_{k-1})) = \begin{bmatrix} -\hat{x}^{(n_a-1)}(t_k) & \dots & -\hat{x}(t_k) \\ u^{(n_b)}(t_k) & \dots & u(t_k) \end{bmatrix}^T \quad (39)$$

with $\hat{x}(t_k)$ given by the ‘auxiliary model’

$$\hat{A}(p, \hat{\boldsymbol{\theta}}(t_{k-1}))\hat{x}(t_k) = \hat{B}(p, \hat{\boldsymbol{\theta}}(t_{k-1}))u(t_k) \quad (40)$$

Then,

$$\begin{aligned} \boldsymbol{\zeta}_f(t_k) &= F(p, \hat{\boldsymbol{\theta}}(t_{k-1}))\boldsymbol{\zeta}(t_k, \hat{\boldsymbol{\theta}}(t_{k-1})) \\ &= \begin{bmatrix} -\hat{x}_f^{(n_a-1)}(t_k) & \dots & -\hat{x}_f(t_k) \\ u_f^{(n_b)}(t_k) & \dots & u_f(t_k) \end{bmatrix}^T \end{aligned} \quad (41)$$

The method will be abbreviated by RSRIVC and it is initialized with RLSSVF. The transition from the LS to the IV based method can be set manually by the user at a time instant denoted $t_{s,iv}$.

Variants of RSRIVC can be considered by fixing the auxiliary model and/or the prefilter. Such an approach has been used in [27], where both the auxiliary model and the prefilter were fixed (in a hybrid, analog-digital

implementation). Experience shows that a variant of RSRIVC that works well in practice is the one which uses an adaptive auxiliary model but a fixed prefilter. Different fixed prefilters can be used: one option is to consider (37) with $\hat{\boldsymbol{\theta}}(t_{k-1})$ being the result of an estimation performed using some training data. Alternatively the prefilter can be fixed with the last RLSSVF estimates obtained just before the switching time $t_{s,iv}$. This alternative is evaluated in an example in Section 7.3. A third option is to use the SVF (24), yielding the recursive instrumental variable state variable filter (RIVSVF) algorithm. RIVSVF is then given by (38), but with $y_f^{(na)}(t_k)$ and $\boldsymbol{\varphi}_f(t_k)$ defined in (28) and (30), respectively. In these algorithms, the filtered instrument $\boldsymbol{\zeta}_f(t_k)$, is defined by

$$\boldsymbol{\zeta}_f(t_k) = F(p)\boldsymbol{\zeta}(t_k, \hat{\boldsymbol{\theta}}(t_{k-1})) \quad (42)$$

with $F(p)$ given by (24), $\boldsymbol{\zeta}(t_k)$ by (39) and the auxiliary model by (40).

5. Implementation issues and practical aspects

This section considers implementation issues and practical aspects of the recursive IV based approaches.

5.1. Stability of the estimates

Low SNR or mismatch model order may lead to unstable estimated LTV models, similarly as in the LTI system case. Thus, in off-line SRIVC estimation, a stability test is required when updating the estimates so that the prefilter and auxiliary model remain stable. In the case of RSRIVC, the prefilter (37) and the auxiliary model (40) depend on the estimates at the previous recursion $\hat{\boldsymbol{\theta}}(t_{k-1})$ and so a stability check is also required *at each recursion*. However, in this recursive situation it is more complicated, since an LTV system is considered. Both the prefilter and the auxiliary model can be written as an LTV state-space model

$$\mathcal{M}_{ss} \begin{cases} \dot{\boldsymbol{x}}(t) = \mathbf{F}(t)\boldsymbol{x}(t) + \mathbf{G}(t)u(t) \\ y(t) = \mathbf{H}(t)\boldsymbol{x}(t) + \mathbf{J}(t)u(t) \end{cases} \quad (43)$$

A check on the stability of these equations can be achieved using the following lemma [see 28, p. 206].

Lemma 5.1. *Suppose the system (43) is uniform exponentially stable, and there exist finite constant τ_1 , τ_2 and τ_3 such that, for all t ,*

$$\|\mathbf{G}(t)\| \leq \tau_1, \quad \|\mathbf{H}(t)\| \leq \tau_2, \quad \|\mathbf{J}(t)\| \leq \tau_3 \quad (44)$$

Then the state equation also is uniformly bounded-input, bounded-output stable.

Assuming that $\mathbf{G}(t)$, $\mathbf{H}(t)$ and $\mathbf{J}(t)$ are bounded, *i.e.* (44) is fulfilled, it remains to check uniform exponential stability, which can be carried out in practice by solving a linear matrix inequality problem.

In general, uniform exponential stability of time-varying parameter models cannot be specified by the location of the eigenvalues of the matrix $\mathbf{F}(t)$. However, if it is assumed that the parameters vary slowly, uniform exponential stability is guaranteed provided the eigenvalues of $\mathbf{F}(t)$ are in the left half plane [29]. This simpler approach is usually considered in recursive estimation and it is used in the present context.

In RSRIVC, to guarantee the stability of the prefilter and auxiliary model, the projection algorithm presented in [4, 5] is used. The previous discussion about stability was done for the state-space representation (43). In practice however, the poles related to the prefilter and auxiliary model, are considered in the stability test and projection algorithm.

Let us present the projection algorithm by defining first a set \mathbf{D}_s containing stable estimates $\hat{\boldsymbol{\theta}}(t_k)$, *i.e.*

$$\begin{aligned} \mathbf{D}_s = \{ & \boldsymbol{\theta} \mid \text{the poles of the characteristic equation } \dots \\ & A(p, \boldsymbol{\theta}) = 0 \text{ are in the left half plane} \} \end{aligned} \quad (45)$$

If the estimate before the stability check is denoted by $\hat{\boldsymbol{\theta}}^*(t_k)$, and after it by $\hat{\boldsymbol{\theta}}(t_k)$, then

$$\hat{\boldsymbol{\theta}}(t_k) = \begin{cases} \hat{\boldsymbol{\theta}}^*(t_k) & \text{if } \hat{\boldsymbol{\theta}}^*(t_k) \in \mathbf{D}_s \\ \hat{\boldsymbol{\theta}}^\diamond(t_k) & \text{if } \hat{\boldsymbol{\theta}}^*(t_k) \notin \mathbf{D}_s \end{cases} \quad (46)$$

where $\hat{\boldsymbol{\theta}}^\diamond(t_k)$ is a stable estimate obtained by reflecting the poles of $A(p, \boldsymbol{\theta}) = 0$ into the left half plane. This means that the sign of the real part of the unstable poles is changed. Note that in this way, the magnitude characteristic of the prefilter and auxiliary model is not changed. As mentioned in [7, p. 373], in recursive identification algorithms, stability issues may arise at the beginning of the data record.

5.2. Correlation between filtered instruments and filtered noise

In an LTI identification problem, the consistency condition C2 from Appendix A.1 is not fulfilled when there is correlation between the filtered instrument and the filtered version of $v_o(t_k)$. SRIVC is an iterative method where both the prefilter and the instrument depend on the estimates of a previous iteration. Then, for SRIVC, condition C2 holds asymptotically as the number of iterations increases and the estimates get close to the true parameter vector. Analogously, in RSRIVC both the filter F and the instrument ζ depend on the estimates of a previous recursion. Then, in an LTI identification problem, we can expect that condition C2 holds asymptotically as the recursions increase and the estimates get close to the true parameter vector.

If the true parameters are slowly varying, there will be always some correlation between filtered instruments and the filtered version of $v_o(t_k)$, and condition C2 will be slightly violated. In order to reduce this correlation, the estimates can be filtered, following the approach first used in the DT case (see [30] and [Chapter 7 in 9]). The filter, denoted here by $M_\theta(p)$, has the additional effect of smoothing the estimates, which improves the stability of the algorithm [4, p. 315]. A scheme of the RSRIVC approach, including the filter $M_\theta(p)$ and the stability test described previously, is shown in Figure 1.

5.3. Choice of hyper-parameters

It is important to carefully set the values for the different hyper-parameters introduced in Sections 4 and 5.2 because they ensure that the algorithms are robust and work well in practice. Some guidelines for choosing them are presented below.

Choice of $\hat{\boldsymbol{\theta}}(t_0)$ and $\mathbf{P}(t_0)$

To start RLSSVF, the initial conditions $\hat{\boldsymbol{\theta}}(t_0)$ and $\mathbf{P}(t_0)$ have to be specified. If there is no knowledge about $\hat{\boldsymbol{\theta}}(t_0)$, the ‘diffuse prior’ assumption can be used: i.e. $\hat{\boldsymbol{\theta}}(t_0) = 0$ together with a diagonal $\mathbf{P}(t_0)$ matrix with large elements reflecting the lack of prior knowledge. Normally $\hat{\boldsymbol{\theta}}(t_0) = 0$ and $\mathbf{P}(t_0) = \gamma \cdot \mathbf{I}$, with γ a scalar and \mathbf{I} the identity matrix. The value of γ depends on the order of magnitude of the system parameters and also on the variance of the output noise σ_e^2 , because $\mathbf{P}(t_k)$ is normalized by σ_e^2 . Nonetheless, as default value we consider $\gamma = 10^4$. For the IV based methods, another option is to choose $\hat{\boldsymbol{\theta}}(t_0)$ and $\mathbf{P}(t_0)$ as the last estimates computed

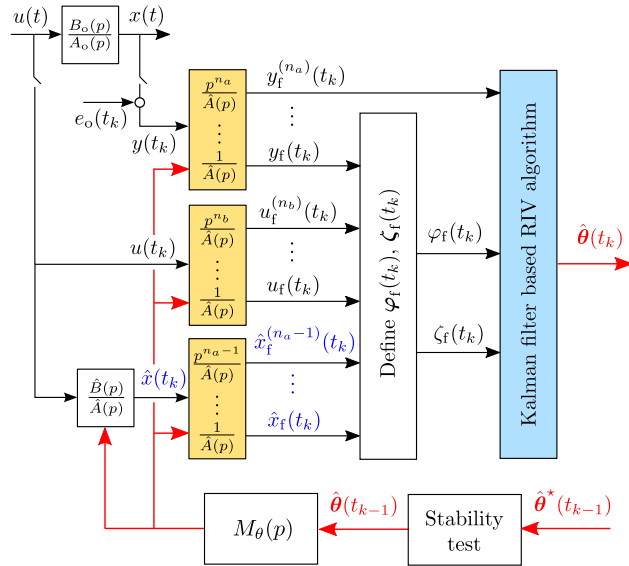


Figure 1: Diagram of RSRIVC including a stability test and a filter $M_\theta(p)$ to reduce the correlation between filtered instruments and filtered noise.

with RLSSVF. That approach is considered in the numerical example in Section 7.2.

Choice of λ_{svf}

The SVF prefilter is a simple alternative to the optimal prefilter used in full RIVC estimation. For the choice of λ_{svf} of the SVF prefilter, the following two aspects have to be considered: (i) Filtering the model to be identified yields a filtered noise $v_f(t_k)$ (see (27)). In general, v_f is a coloured noise that deteriorates the accuracy of the estimates. In order to reduce the effect of high frequency noise, λ_{svf} should be small. (ii) The larger the value of λ_{svf} , the more informative are the filtered signals, which leads to a faster convergence rate¹. Thus, there is a trade-off in the choice of λ_{svf} . The experience shows that a suitable choice is a value in the range

$$[0.5\omega_{b,\max}, 1.1\omega_{b,\max}] \quad (47)$$

¹In [31] it is also shown that the smaller the sampling time, the faster the convergence rate.

with $\omega_{b,\max}$ the largest bandwidth of the LTV system. Assuming that the sampling frequency $1/T_s$ is such that,

$$1/T_s \approx 10\omega_b \quad (48)$$

a first guess for λ_{svf} can be obtained using (47) and $\omega_{b,\max} = \omega_b$. For LTV systems with relatively large bandwidth variations, the specification of λ_{svf} can be critical since the system bandwidth is time-varying (see the numerical example in [24]). In practice, some prior guidance on the selection of the cut-off frequency for the SVF, λ_{svf} , can be obtained by estimating local LTI models on segments of the training data using RIVC, SRIVC or LSSVF. Additionally, the Nyquist frequency can be considered as an upper bound for λ_{svf} .

Choice of \mathbf{Q}_n

In order to use the Kalman filter type of recursive updating, the user has to specify the NVR matrix \mathbf{Q}_n by defining the matrix \mathbf{Q}_w and the scalar σ_e^2 (see (14)). In general, \mathbf{Q}_w is a symmetric positive semidefinite matrix. Usually it is assumed that $\sigma_e^2 = 1$ and \mathbf{Q}_w is set as $\mathbf{Q}_w = \text{diag}\{\sigma_{w_1}^2 \dots \sigma_{w_{n_\theta}}^2\}$, where $\sigma_{w_i}^2$ is the variance of the parameter $\theta_i(t_k)$. If it can be assumed that all the parameters vary at the same speed, then $\mathbf{Q}_w = \sigma_w^2 \mathbf{I}$, with \mathbf{I} the identity matrix. Since the parameters are assumed to be slowly varying, it is possible to detect which parameters are varying by identifying local LTI models on segments of the training data, as in the selection of λ_{svf} above. In case the parameter $\theta_i(t_k)$ is known to be constant, then $\sigma_{w_i}^2$ can be set to zero or a value close to zero. If the parameter $\theta_i(t_k)$ is time-varying, then it is suggested to start setting $\sigma_{w_i}^2 = 0.1$ and then continue tuning by trial and error.

A more complex approach, that has worked well in the off-line case (see Section 4.5.3 in [3]) and has been used in recursive algorithms available in the CAPTAIN Toolbox for MATLAB², is to optimize the NVR hyper-parameters using either maximum likelihood optimization based on prediction error decomposition, or optimization of the single- or multiple-step-ahead prediction errors. The simplest approach of this kind can be utilized in low noise situations (experience suggests with noise levels up to 10% by standard deviation).

²The CAPTAIN Toolbox is freely available via the website http://captaintoolbox.co.uk/Captain_Toolbox.html/Captain_Toolbox.html

It involves first using the `irwsmooth` routine in CAPTAIN Toolbox to differentiate the data (as used successfully in, *e.g.*, [32]). Then the `dlropt` routine is employed to optimize the diagonal elements of \mathbf{Q}_w so that these can be used in the `dlr` routine, which provides on-line estimates of the changing parameters in a time-varying parameter linear regression model of the form (9). \mathbf{Q}_w optimized in this manner is then utilized in the proposed on-line algorithms. This optimization approach is employed later in the example described in Section 7.2, where the noise level is low.

Choice of $t_{s,iv}$

The recursive IV based methods are initialized using RLSSVF. The switch between these algorithms takes place at some time instant $t_{s,iv}$, when the IV based approaches are then initialized. The switch at $t_{s,iv}$ should be carried out once convergence of the RLSSVF estimates is achieved. This can be assessed by checks on one (or more) of the following: (i) the variability or steady state condition of $\Delta\hat{\boldsymbol{\theta}}(t_k) = \hat{\boldsymbol{\theta}}(t_{k-1}) - \hat{\boldsymbol{\theta}}(t_k)$, (ii) the size of the parameter covariance matrix $\mathbf{P}(t_k)$, which can be measured in terms of trace or determinant, (iii) the condition number of the parameter covariance matrix $\mathbf{P}(t_k)$, (iv) the prediction error $\varepsilon(t_k)$, and (v) the output error $\varepsilon_y(t_k)$ defined in (33).

For instance, the variability of $\Delta\hat{\boldsymbol{\theta}}(t_k)$ can be measured through its variance, as computed recursively with, for example, a simple exponential moving average of the kind used in an analogous manner by [33]. The steady state condition of the output error can be assessed using the approach proposed in [34]. Note that the automatic detection of the convergence of RLSSVF requires setting threshold values and these will depend on the system being studied. A simple option that is considered in the examples in Section 7 is to use RLSSVF and check its convergence based on the output error.

Choice of $M_\theta(p)$

Noisy data will lead to estimates with high variability, which at the same time can cause numerical issues in the identification algorithm. Then, in such cases, a filter $M_\theta(p)$ (see Figure 1) should be used. Slowly time-varying parameters are considered, and thus the signals have low frequency content. Thus, $M_\theta(p)$ should be a lowpass filter such that the frequencies of interest of the time-varying parameters are not affected. A simple choice is a first order filter,

$$M_\theta(p) = \frac{1}{\tau_\theta p + 1} \quad (49)$$

with τ_θ being a hyper-parameter to be chosen. Note that choosing a filter with a bandwidth close to the largest frequency of interest of the time-varying parameters can delay the estimates such that the fit between measured and simulated output will be affected. Thus, the filter bandwidth should be much larger than the largest frequency of interest. In order to implement a filter such as (49), it has to be discretized, for instance using the Tustin method. Alternatively, a low-pass finite impulse response (FIR) filter can be considered. Similarly to the approach in [30], a simple multi-sample delay $M_\theta(q^{-m})$, with $m \geq 1$ can also be used³. As default option, $M_\theta(q^{-1}) = q^{-1}$, is considered.

Remark 2. *The algorithms that have been presented, are available in the CON- tinuous-Time System IDentification (CONTSID) toolbox, which can be freely downloaded*⁴.

6. Measures of estimator performance

In practical applications, the tracking ability of the proposed approaches can be measured only indirectly in terms of the prediction error ε , or the fit between measured output and simulated output. The fit is defined as follows

$$\mathcal{F} = 100 \times \left[1 - \frac{\|y(t_k) - \hat{y}(t_k, \hat{\boldsymbol{\theta}}, u)\|}{\|y(t_k) - \text{mean}(y(t_k))\|} \right] \quad (50)$$

In numerical simulations, however, the true parameters are available, which allows us to assess the performance by means of the tracking error

$$\tilde{\boldsymbol{\theta}}(t_k) = \boldsymbol{\theta}_o(t_k) - \hat{\boldsymbol{\theta}}(t_k) \quad (51)$$

The i th element of $\boldsymbol{\theta}_o(t_k)$ is denoted by $\theta_i^o(t_k)$, *i.e.*

$$\boldsymbol{\theta}_o^T(t_k) = [\theta_1^o(t_k) \quad \theta_2^o(t_k) \quad \dots \quad \theta_{n_\theta}^o(t_k)] \quad (52)$$

Analogously, the i th element of $\hat{\boldsymbol{\theta}}(t_k)$ is denoted by $\hat{\theta}_i(t_k)$. Then, the parameter relative error can be defined as follows

$$\tilde{\theta}_{i,r}(t_k) = \frac{\theta_i^o(t_k) - \hat{\theta}_i(t_k)}{\theta_i^o(t_k)} \quad (53)$$

³The approach from [30] can be understood as a combination of two different variants to define the instruments (see [35, p. 25, variants 1 and 3]).

⁴<http://www.contsid.cran.univ-lorraine.fr/>

where $\tilde{\theta}_{i,r}(t_k)$ is the i th element of the parameter relative error vector,

$$\tilde{\boldsymbol{\theta}}_r^T(t_k) = [\tilde{\theta}_{1,r}(t_k) \quad \tilde{\theta}_{2,r}(t_k) \quad \dots \quad \tilde{\theta}_{n_\theta,r}(t_k)] \quad (54)$$

Finally, the average value of the squared norm of the relative error vector for the k th time instant can be computed as follows.

$$\text{MMSE} = \frac{1}{N \cdot n_\theta} \sum_{k=1}^N \|\tilde{\boldsymbol{\theta}}_r(t_k)\|^2 \quad (55)$$

By adding and subtracting $\mathbb{E}[\hat{\boldsymbol{\theta}}(t_k)]$ in (51), the tracking error can be split into two parts [8, p. 524], the lag error $\tilde{\boldsymbol{\theta}}_l(t_k)$ and the noise error $\tilde{\boldsymbol{\theta}}_n(t_k)$, *i.e.*

$$\tilde{\boldsymbol{\theta}}(t_k) = \underbrace{\boldsymbol{\theta}_o(t_k) - \mathbb{E}[\hat{\boldsymbol{\theta}}(t_k)]}_{\tilde{\boldsymbol{\theta}}_l(t_k)} + \underbrace{\mathbb{E}[\hat{\boldsymbol{\theta}}(t_k)] - \hat{\boldsymbol{\theta}}(t_k)}_{\tilde{\boldsymbol{\theta}}_n(t_k)} \quad (56)$$

These errors cannot be minimized simultaneously due to the well-known trade-off of recursive algorithms between tracking capability and noise sensitivity [6]. For tracking problems, the lag error can be compared to the standard concept of bias in LTI systems [36, p. 15]. The presence of lag error is evidence of the non-stationary nature of the environment [8], meaning that, in the case of time-varying parameters, the lag error is larger than in the time-invariant case. Indeed, in the latter, the lag error could be zero.

The lag error and noise error can be computed by Monte Carlo simulations using a given input and different measurement noise realizations. Then, the expectation $\mathbb{E}[\hat{\boldsymbol{\theta}}(t_k)]$ is replaced by the sample mean $\bar{\boldsymbol{\theta}}(t_k)$ over the total number of simulations. Such analysis is applied in Section 7.1. It is then convenient to define two additional measures, the average mean squared value of the lag error defined by

$$\text{MMSE}_l[\hat{\boldsymbol{\theta}}(t_k)] = \frac{1}{N \cdot n_\theta} \sum_{k=1}^N \|\boldsymbol{\theta}_o(t_k) - \bar{\boldsymbol{\theta}}(t_k)\|^2 \quad (57)$$

and the average mean squared value of the noise error given by

$$\text{MMSE}_n^i[\hat{\boldsymbol{\theta}}(t_k)] = \frac{1}{N \cdot n_\theta} \sum_{k=1}^N \|\bar{\boldsymbol{\theta}}(t_k) - \hat{\boldsymbol{\theta}}(t_k)\|^2 \quad (58)$$

where the superscript i means that the computation is carried out w.r.t. the estimate $\hat{\boldsymbol{\theta}}(t_k)$ from the i th simulation, which can be chosen arbitrarily.

Remark 3. *In the identification of LTI models, IV methods are applied to remove (or reduce) bias. Analogously, in the LTV case, the aim is to reduce the lag error which comes at the expense of slightly increasing the noise error. This is the well-known trade-off of recursive algorithms between tracking ability and noise sensitivity.*

It is important to mention that two different estimations of time-varying parameters, computed with different methods, can have a similar fit; however that does not mean that the estimates are similar, that is, they can have different values for MMSE (see the Monte Carlo simulation in the example from Section 7.1). This happens because, for a certain estimation, the errors among the parameters compensate each other; thus the error in the parameter estimates do not have a significant impact in the fit.

7. Examples

7.1. Numerical example

Let us consider a mass-spring-damper system given by:

$$(mp^2 + c(t)p + k)x(t) = k_u u(t)$$

with time-varying damping coefficient $c(t)$,

$$c(t) = 1 + 0.4 \cos(2\pi t/4500 + \pi) \quad (59)$$

The other coefficients are constant: $m = 0.6$, $k = 1$, $k_u = 1.2$. Then, the data generating system is given by

$$\mathcal{S} \begin{cases} (p^2 + a_1^o(t)p + a_2^o(t))x(t) = b_0^o(t)u(t) \\ y(t_k) = x(t_k) + e(t_k) \end{cases} \quad (60)$$

where $a_1^o(t) = c(t)/m$, $a_2^o(t) = k/m$, $b_0^o(t) = k_u/m$. In this example, the DT measurement noise is a zero-mean Gaussian noise with variance $\sigma_e^2 = 0.03$. Input-output data is generated using a sampling time $T_s = 0.3$ s; part of the data is shown in Figure 2.

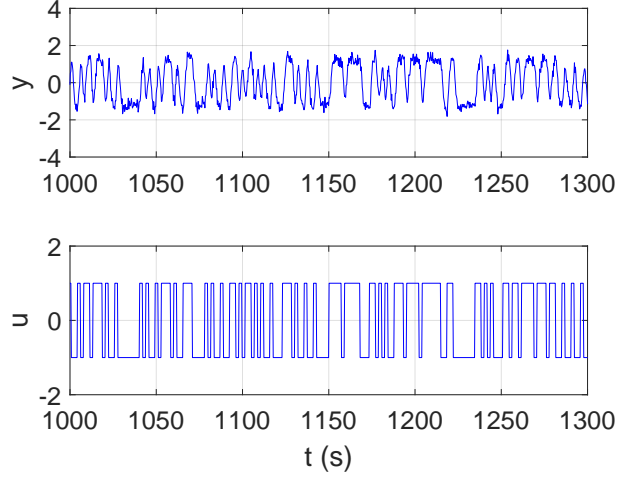


Figure 2: Part of the input-output data.

Single experiment analysis

In this section, the performance of the indirect and direct approaches are analysed using a single experiment. Considering the usual zero-order hold (ZOH) assumption for the input, the DT equivalent system is given by

$$x(t_k) + \alpha_1^o(t_k)x(t_{k-1}) + \alpha_2^o(t_k)x(t_{k-2}) = \beta_1^o(t_k)u(t_{k-1}) + \beta_2^o(t_k)u(t_{k-2}) \quad (61a)$$

$$y(t_k) = x(t_k) + e(t_k) \quad (61b)$$

Considering the discussion in Section 5.3, the hyper-parameters of the approaches that are being tested are given next.

- IndirectARX: $\mathbf{Q}_n = \text{diag}([10^{-3} \ 10^{-3} \ 10^{-5} \ 10^{-5}])$. As a prefilter a SVF of order 2 with $\lambda_{\text{svf}} = 2 \text{ rad/s}$ is considered. The SVF is discretized with ZOH.
- IndirectOE: $N_{\text{s,oe}} = 1501$ ($t = 450 \text{ s}$), $\mathbf{Q}_n = \text{diag}([10^{-5} \ 10^{-5} \ 10^{-3} \ 10^{-3}])$.
- RLSSVF: $\lambda_{\text{svf}} = 2 \text{ rad/s}$, $\mathbf{Q}_n = \text{diag}([0 \ 10^{-4} \ 0])$.
- RSRIVC: $t_{\text{s,iv}} = 450 \text{ s}$, $\lambda_{\text{svf}} = 2 \text{ rad/s}$, $\mathbf{Q}_n = \text{diag}([0 \ 10^{-4} \ 0])$.
- RIVSVF: $t_{\text{s,iv}} = 450 \text{ s}$, $\lambda_{\text{svf}} = 2 \text{ rad/s}$, $\mathbf{Q}_n = \text{diag}([0 \ 10^{-4} \ 0])$.

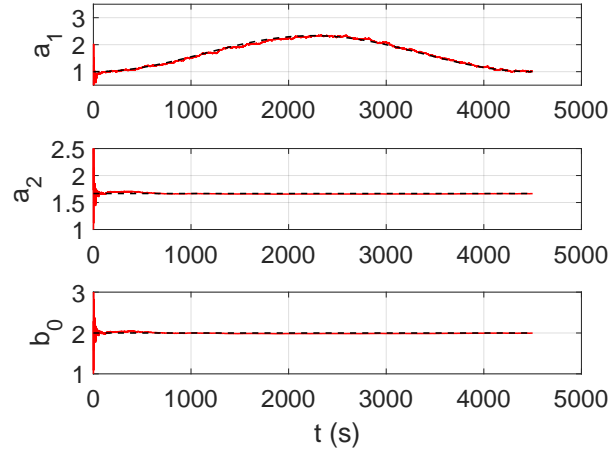


Figure 3: True CT parameters (dashed lines) and RSRIVC estimates (continuous line).

λ_{svf} has been chosen slightly larger than the maximum system bandwidth which is approximately 1.8 rad/s. $t_{s,iv}$ has been defined by using RLSSVF and checking its convergence based on the output error; $t_{s,iv}$ is then used to set the corresponding value for $N_{s,oe}$. In this example, the noise level by standard deviation is approximately 17%; therefore, the approach presented in Section 5.3 to estimate \mathbf{Q}_n cannot be applied, and it is estimated by trial and error. Keep in mind that the choice of \mathbf{Q}_n is a trade-off between tracking ability and noise sensitivity, or in other words a trade-off between noise error and lag error (see Remark 3). Therefore, there is no unique value of \mathbf{Q}_n which yield suitable estimates.

The best result is obtained with RSRIVC (see Figure 3), while from the indirect approaches the best result is obtained with IndirectARX (see Figure 4). Note that, the conversion from DT to CT parameters yields two parameters b_i ; according to (3), b_1 in Figure 4 corresponds to b_0 in Figure 3, and b_0 in Figure 4 should be equal to zero.

The correct DT model is OE and not ARX. Therefore, for IndirectOE, once the switch from ARX to OE takes place, more accurate estimates are obtained. On the other hand, with the indirect approaches, there is one additional DT parameter to be estimated and all the DT parameters are time-varying. That makes more complex the estimation problem and the choice of the NVR matrix \mathbf{Q}_n .

During the initial recursions, the DT to CT conversion can fail because

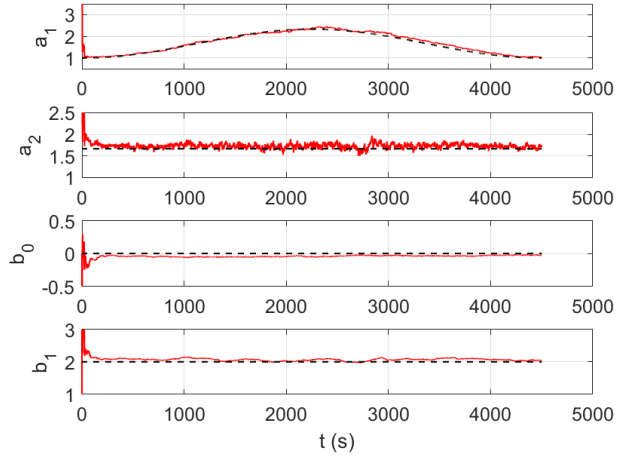


Figure 4: True CT parameters (dashed lines) and CT estimates obtained from IndirectARX (continuous lines).

the poles are close to the origin; or the model order has increased to handle real negative poles. In these situations, at recursion t_k , the estimates are set to those obtained at the previous sampling instant t_{k-1} . All these issues probably make the direct estimation of CT LTV models a better approach.

Monte Carlo simulation

In order to illustrate the benefits of RSRIVC and its variant RIVSVF over RLSSVF, a Monte Carlo simulation analysis is performed with 100 realizations. Since this is a numerical example, the noise-free output data is available. Then, the mean and standard deviation of the fits (see (50)) computed with the noise-free output (\mathcal{F}_0) are shown in Table 1; as it can be observed, in terms of $\text{mean}(\mathcal{F}_0)$, the IV based approaches outperform RLSSVF. For each method, also the mean and standard deviation of the fits computed with the noisy output data (\mathcal{F}) are shown in Table 1. It can be seen that the mean of these fits are very similar; however, the estimates are different, as it is shown next.

The MMSE is computed for each estimation method and for each simulation. Also, for each estimation method, the average of MMSE, denoted by $\overline{\text{MMSE}}$, is computed. The values of $\overline{\text{MMSE}}$ for the different approaches are shown in Table 2; notice that the value of $\overline{\text{MMSE}}$ for RIVSVF is much smaller than the value for RLSSVF, however, the smallest $\overline{\text{MMSE}}$ is obtained for RSRIVC, as expected. RIVSVF also exhibits good performance, partly

Table 1: Mean and standard deviation for the fits obtained with the proposed approaches. \mathcal{F} : fits computed with measured output; \mathcal{F}_0 : fits computed with noisy-free output.

Method	mean(\mathcal{F}_0)	std(\mathcal{F}_0)	mean(\mathcal{F})	std(\mathcal{F})
RLSSVF	96.3	0.1348	82.6	0.1143
RSRIVC	98.3	0.1173	82.9	0.1048
RIVSVF	98.1	0.1449	82.8	0.1037

Table 2: Performance indexes for the different approaches. $\overline{\text{MMSE}}$: mean of MMSE (see (55) for definition of MMSE); $\text{MMSE}_l[\hat{\boldsymbol{\theta}}(t_k)]$: average mean squared value of the lag error; $\text{MMSE}_n^1[\hat{\boldsymbol{\theta}}(t_k)]$: average mean squared value of the noise error considering the run 1 as a reference.

Method	$\overline{\text{MMSE}}$	$\text{MMSE}_l[\hat{\boldsymbol{\theta}}(t_k)]$	$\text{MMSE}_n^1[\hat{\boldsymbol{\theta}}(t_k)]$
RLSSVF	37.8	$1.41 \cdot 10^{-2}$	$3.35 \cdot 10^{-4}$
RSRIVC	1.6	$3.86 \cdot 10^{-4}$	$3.04 \cdot 10^{-4}$
RIVSVF	3.2	$8.71 \cdot 10^{-4}$	$4.53 \cdot 10^{-4}$

because of the fortuitous selection of λ_{svf} .

In the LTI case, it is known that the RLSSVF estimates are always biased due to the measurement noise. Even if the bias cannot be removed, it can be reduced by a proper choice of the cut off frequency λ_{svf} . In the LTV case, the issue is more difficult since the system bandwidth is varying. However, instrumental variable methods, like RSRIVC, can cope with the measurement noise and reduce the lag error, as pointed out previously in Remark 3. Table 2 shows the average mean squared value of the lag error $\text{MMSE}_l[\hat{\boldsymbol{\theta}}(t_k)]$, and average mean squared value of the noise error $\text{MMSE}_n^1[\hat{\boldsymbol{\theta}}(t_k)]$. Note that the smallest values are obtained for RSRIVC. On the other hand, the noise error is smaller for RLSSVF than for RIVSVF. Nonetheless, if these two approaches are compared in relative terms, it can be observed that the reduction of the lag error with RIVSVF is greater than the increase in the noise error.

7.2. An electrical circuit example

In [37], real data collected from a second order bandpass filter has been proposed as a benchmark for both LPV model and LTV model identification⁵. The electronic circuit has a parallel connection of an n -type J-FET transistor and a 470 k Ω resistor. The scheduling variable $r(t)$, the input $u(t)$ and output $y(t)$ are voltages. The noise level in the measurements is very low, with an SNR of more than 60 dB. Note that some small non-linear effects are also present.

Several scenarios are considered in [37], consisting on different excitations and scheduling signals are provided. In order to evaluate the algorithms proposed in the present paper, one of the scenarios with smooth parameter variations is considered. The chosen scenario corresponds to the mat-file:

$$\text{MS_Harm_h3_N15640_RMS70_P2P700.mat} \quad (62)$$

According to the qualitative description of the circuit given in [37], the system has two complex poles and one zero at the origin. Therefore, the following CT LTV OE model is considered

$$(p^2 + a_1(t)p + a_2(t))x(t) = b_0(t)pu(t) \quad (63a)$$

$$y(t_k) = x(t_k) + e(t_k) \quad (63b)$$

where the input $u(t)$ and output $y(t)$ are voltages.

The data which is contained in (62) consist of 6 experiments, denoted here by Ei , with $i = 1, \dots, 6$. Since only LTV models are identified, the measured scheduling signal is not used. For the present purposes, only the following experiments, with a sampling frequency of 156250 Hz, are used: $E1$ as estimation data and $E3$ as validation data. As part of the preprocessing, the mean levels are removed from the data. A section of the input-output data from experiment $E1$ is presented in Figure 5.

The goal of the analysis is to assess the performance of RLSSVF, RIVSVF and RSRIVC. Experiment $E1$ is used to guide the choice of the hyper-parameters. λ_{svf} is set equal to the Nyquist frequency, *i.e.* $\lambda_{svf} = \pi/T_s \approx 4.91 \cdot 10^5$ rad/s; this is possible to do since the data is not noisy. From [37], it is known that the parameter $b_0(t)$ is constant. In addition, tests have shown

⁵see http://homepages.ulb.ac.be/~jlataire/BenchmarkSYSID2015_ELEC_LTV_LP/

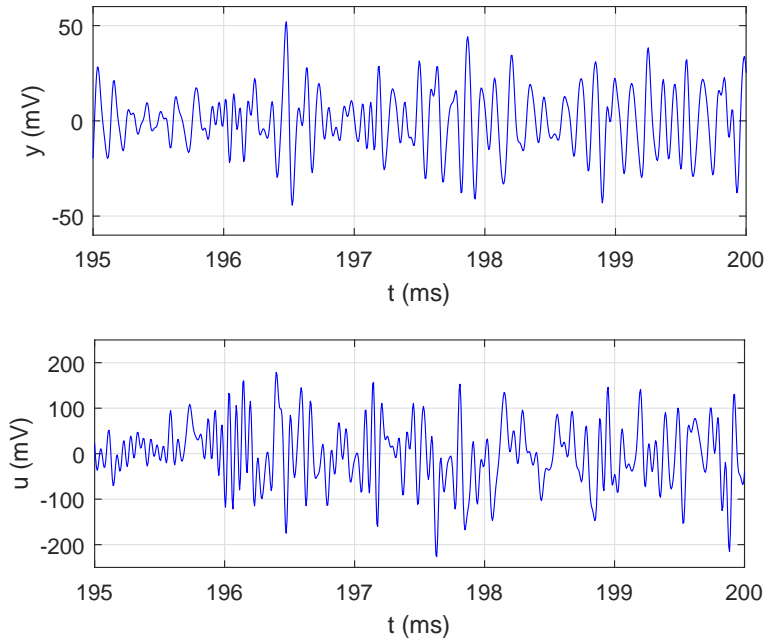


Figure 5: Part of the input-output data for the identification of the circuit model.

that also $a_1(t)$ is nearly constant. This means that, in \mathbf{Q}_n , the corresponding NVR values for both $a_1(t)$ and $b_0(t)$ can be set to zero. As it is confirmed by the results that are shown next, the noise on the data is low in this example; thus, it is possible to use the hyper-parameter optimization approach outlined in Section 5.3, which yields $\mathbf{Q}_n = \text{diag}([0 \ 0.67 \ 0])$. Regarding $t_{s,iv}$, noting the convergence of RLSSVF, $t_{s,iv}$ is set to 20 ms. Finally, in order to choose $\hat{\boldsymbol{\theta}}(t_0)$ and $\mathbf{P}(t_0)$, the order of magnitude of the system parameters has to be considered. In this example, the values of the system parameter are large, so the default value for γ (see Section 5.3) might not be suitable. However, in this case $\hat{\boldsymbol{\theta}}(t_0)$ and $\mathbf{P}(t_0)$ are obtained as the last RLSSVF values.

The RSRIVC estimates are shown in Figure 6 but, because the noise on the measurements is so low, the standard deviations are not visible at the given scale. For instance, in the last estimation the standard deviations are $7.2 \cdot 10^{-12} \%$, $8.3 \cdot 10^{-11} \%$ and $4.2 \cdot 10^{-12} \%$, respectively, for \hat{a}_1 , \hat{a}_2 and \hat{b}_0 .

To assess the quality of the estimates, the benchmark criterion proposed in [37] is used, which corresponds to the RMS error computed on the second

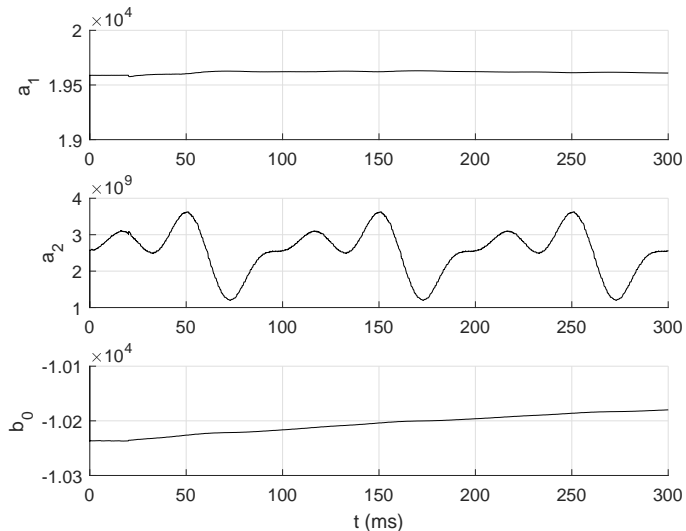


Figure 6: RSRIVC estimates and their corresponding standard deviation: because the noise level is so small, the standard deviations are not visible at this scale.

period of the simulated output signal, *i.e.*

$$\text{RMSE}_y = \sqrt{\frac{1}{N - N_{\text{tr}}} \sum_{k=N+N_{\text{tr}}}^{2N-1} (y(t_k) - \hat{y}(t_k, \hat{\theta}, u))^2} \quad (64)$$

where $y(t_k)$ is the measured output from the validation data. The simulated output $\hat{y}(t_k, \hat{\theta}, u)$ is computed from the estimate $\hat{\theta}$ and the measured input $u(t_k)$ from the validation data $E3$. $N_{\text{tr}} = 500$ is used to eliminate the transient effects. In addition, the fit between the measured and simulated outputs is computed. In [37], the proposed model structure for the data set used here, which is different than ours (see (63)), has 3 poles and one zero, with none of the parameters in the numerator being set to zero.

The results obtained, using the data set $E3$ for validation, are shown in Table 3. The fits are very good, which again indicates that there is a low noise level in the data. From the RMSE_y , it can be seen that RSRIVC is doing better than RLSSVF and RIVSVF, but this result is still not as good as that given in [37]. Notice that in our case, however, apart from having a simpler model structure, only the data set $E1$ is considered as estimation data. On the other hand, assuming that the parameters are slowly varying, they are estimated using a random walk model. The advantage of it is that

Table 3: Performance indexes using validation data.

Method	RMSE _y (mV)	\mathcal{F} (%)
[37]	0.1156	not reported
RLSSVF	0.6487	95.9
RIVSVF	0.6445	96.0
RSRIVC	0.4137	97.4

no fixed global model for the parameter variations is needed. If the system properties are time-varying due to ageing effects or environmental conditions, the random walk model can be specially useful in an adaptive control scheme that includes a recursive identification approach as the one proposed in this paper.

7.3. A hydrological system example

This example concerns the modelling of the measured daily rainfall-flow data from the ephemeral River Canning in Western Australia [38], as shown in Figure 7, for 701 days between 1985 and 1987. The relationship between rainfall and flow is clearly nonlinear, with rainfall leading to run-off and flow in the river during the wet periods of the year but no flow occurring at all during the dry periods. This is because transient rainfall during this dry period is absorbed by the dry soil. Indeed, this example is used as an demonstration example for nonlinear DT model estimation in the CAPTAIN Toolbox for MATLAB. Nevertheless, it can be seen that it is also possible to represent this system by the following simple first order CT LTV model between the input rainfall $u(t)$ and output flow $x(t)$:

$$(p + a_1(t))x(t) = b_0(t)u(t) \quad (65a)$$

$$y(t_k) = x(t_k) + e(t_k) \quad (65b)$$

based on the sampled rainfall and flow measurements $u(t_k)$ and output $y(t_k)$ shown in Figure 7. This approach is useful because it provides an adaptive model that can be used for on-line flow forecasting.

In this example, it is known that the rainfall is varying between samples so the Tustin approach is used for discretization. The hyper-parameter values were selected partly using the guidelines from Section 5.3 and partly by

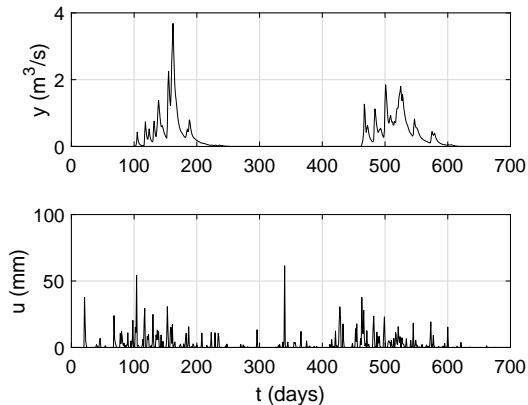


Figure 7: Input-output data for the rainfall-flow system.

trial and error; the latter is simple in this example because there are only two parameters, one of which appears almost constant. Note that trial and error is applied seeking that, by using the estimation data, the fit between measured output and simulated output is maximized. Let us discuss the choices of the hyperparameters. Choice of λ_{svf} : from the sampling frequency, an order of magnitude for the bandwidth can be obtained considering (47) and (48), which yields $\lambda_{\text{svf}} \in [0.32 \ 0.69]$ rad/day. That give us an order of magnitude for λ_{svf} . By trial and error, $\lambda_{\text{svf}} = 0.25$ rad/day, is finally chosen. Choice of $\boldsymbol{\theta}(t_0)$ and $\mathbf{P}(t_0)$: if the parameter a_1 is constant, as it happens to be, $a_1(t_0) = \lambda_{\text{svf}} = 0.25$ rad/day. Therefore, $\boldsymbol{\theta}(t_0) = [0.25 \ 0]$, and since there is some knowledge about $a_1(t_0)$, it is assumed $\mathbf{P} = \text{diag}([10 \ 10^4])$. Choice of \mathbf{Q}_n : the estimation of \mathbf{Q}_n is done by trial and error using RLSSVF. First it is considered that $\mathbf{Q}_n = \sigma_\omega^2 \mathbf{I}$, and then the more involved case where $\mathbf{Q}_n = \text{diag}([\sigma_{\omega_1}^2 \ \sigma_{\omega_2}^2])$; finally, $\mathbf{Q}_n = \text{diag}([0.0001 \ 0.006])$, is chosen. Choice of $t_{\text{s,iv}} = 125$ days: considering the convergence of RLSSVF, $t_{\text{s,iv}} = 125$ days is chosen. The filter $M_\theta(p)$ is not used for the IV based approaches because better estimates are obtained without it. The reason for this is probably the relatively slow sampling rate in this example.

In Table 4 the fits for the different approaches are compared. The result for RSRIVC with a fixed prefilter is also included; in this case the prefilter is fixed with the last RLSSVF estimates, that is, the estimates obtained just before the switching time $t_{\text{s,iv}}$. The best fit corresponds to RSRIVC with a fixed prefilter; the estimates obtained with this approach are plotted in Figure 8, together with the standard deviation bounds, shown in grey. The

Table 4: Fits for the different methods.

Method	\mathcal{F} (%)
RLSSVF	72.3
RIVSVF	72.2
RSRIVC	72.2
RSRIVC with fixed prefilter	72.5

expansion of these bounds at various stages in the estimation is caused by the absence of any rainfall input over these sections of the data, so that the recursive estimate of b_0 is zero or nearly zero and the uncertainty in the estimates grows continually until rainfall occurs. This is to be expected, of course, and it demonstrates that the algorithm is working well in this regard.

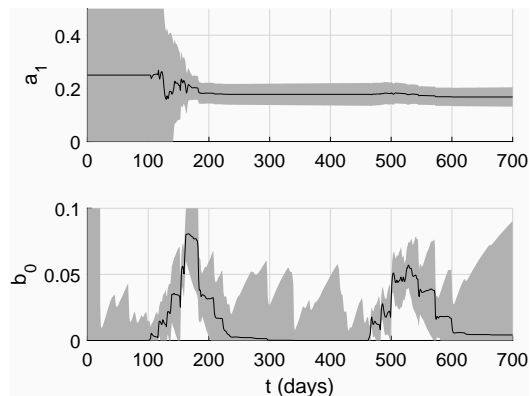


Figure 8: Estimates obtained with RSRIVC with fixed prefilter and their corresponding standard deviation.

8. Acknowledgements

This work was supported by the Advanced Center for Electrical and Electronic Engineering, AC3E, Basal Project FB0008, CONICYT.

9. Conclusions

This paper addresses the problem of estimating slowly variable parameters of continuous-time models in real time, from sampled data. It proposes and evaluates recursive least squares and instrumental variable algorithms using both simulated and real data. This includes implementation issues and the practical aspects of the proposed instrumental variable algorithms. The numerical example uses both single experiment and Monte Carlo based simulations; and the results of this analysis demonstrate, through different measures of estimator performance, the benefit of using the instrumental variable approach. Finally, the recursive estimation algorithms are tested on two practical identification problems, using real data, and the results obtained confirm the practical utility of the algorithms and provide important practical guideline to the user.

References

- [1] M. Niedźwiecki, Identification of Time-varying Processes, Wiley, 2000.
- [2] S. Chan, Z. Zhang, Local polynomial modeling and variable bandwidth selection for time-varying linear systems, *IEEE Transactions on Instrumentation and Measurement* 60 (3) (2011) 1102–1117.
- [3] P. C. Young, Recursive Estimation and Time-series Analysis. An Introduction for the Student and Practitioner, Springer-Verlag, Berlin, 2nd edition, 2011.
- [4] L. Ljung, T. Söderström, Theory and Practice of Recursive Identification, The MIT Press, Cambridge, MA, 1983.
- [5] G. C. Goodwin, K. S. Sin, Adaptive Filtering Prediction and Control, Dover Publications Inc., 1984.
- [6] L. Ljung, S. Gunnarsson, Adaptation and tracking in system identification - a survey, *Automatica* 26 (1) (1990) 7–21.
- [7] L. Ljung, System Identification. Theory for the User, 2nd Edition, Prentice Hall, Upper Saddle River, 1999.
- [8] S. Haykin, Adaptive Filter Theory, 5th Edition, Pearson, 2014.

- [9] P. C. Young, *Recursive Estimation and Time-Series Analysis: An Introduction*, Springer-Verlag, Berlin, 1984.
- [10] J. Lataire, *Frequency domain measurement and identification of linear, time-varying systems*, Ph.D. thesis, Vrije Universiteit Brussel (2011).
- [11] M. Bergamasco, M. Lovera, *Continuous-time predictor-based subspace identification using Laguerre filters*, *IET Control Theory and Applications* 5 (7) (2011) 856–867.
- [12] H. Garnier, M. Mensler, A. Richard, *Continuous-time Model identification from sampled data: Implementation issues and performance evaluation*, *International Journal of Control* 76 (13) (2003) 1337–1357.
- [13] G. Rao, H. Unbehauen, *Identification of continuous-time systems*, *IEE Proceedings* 153 (2) (2006) 185–220.
- [14] Z. Jiang, W. Schaufelberger, *A recursive identification method for continuous time-varying linear systems*, in: *34th Midwest Symposium on Circuits and Systems*, Vol. 1, Monterey, CA, USA, 1991, pp. 436–439.
- [15] M. Östring, S. Gunnarsson, *Recursive identification of physical parameters in a flexible robot arm*, *Asian Journal of Control* 6 (3) (2004) 407–414.
- [16] T. Co, S. Ungarala, *Batch scheme recursive parameter estimation of continuous-time systems using the modulating function method*, *Automatica* 33 (6) (1997) 1185–1191.
- [17] S. Ungarala, T. Co, *Time-varying system identification using modulating function and spline models with application to bio-processes*, *Computer & Chemical Engineering* 24 (12) (2000) 2739–2753.
- [18] P.C.Young, A.J.Jakeman, *Refined instrumental variable methods of time-series analysis: Part III, extensions*, *International Journal of Control* 31 (1980) 741–764.
- [19] K. Åstrom, B. Wittenmark, *Adaptive Control*, Dover Publications Inc., 2008.
- [20] R. Isermann, M. Münchhof, *Identification of Dynamic Systems. An Introduction with Applications*, Springer-Verlag, 2010.

- [21] M. Mensler, S. Joe, T. Kawabe, Identification of a toroidal continuously variable transmission using continuous-time system identification methods, *Control Engineering Practice* 14 (1) (2006) 45–58.
- [22] H. Garnier, M. Gilson, P. C. Young, E. Huselstein, An optimal IV technique for identifying continuous-time transfer function model of multiple input systems, *Control Engineering Practice* 46 (15) (2007) 471–486.
- [23] P. C. Young, H. Garnier, M. Gilson, Refined instrumental variable identification of continuous-time hybrid Box-Jenkins models, in: *Identification of continuous-time models from sampled data*, Vol. H. Garnier and L. Wang (Eds.), Springer, 2008, pp. 1–30.
- [24] A. Padilla, H. Garnier, P. C. Young, J. I. Yuz, Real-time identification of continuous-time linear time-varying systems, in: *IEEE 55th Conference on Decision and Control*, Las Vegas, US, 2016, pp. 3769–3774.
- [25] L. Gustafsson, M. Olsson, Robust on-line estimation, Master’s thesis, Lund Institute of Technology (1999).
- [26] J. E. Parkum, Recursive identification of time-varying systems, Ph.D. thesis, Technical University of Denmark (1992).
- [27] P. C. Young, An instrumental variable method for real-time identification of a noisy process, *Automatica* 6 (1970) 271–287.
- [28] W. J. Rugh, *Linear System Theory*, Prentice Hall, 1996.
- [29] A. Ilchmann, D. Owens, D. Prätzel-Wolters, Sufficient conditions for stability of linear time-varying systems, *System & Control Letters* 9 (1987) 157–163.
- [30] K. Wong, E. Polak, Identification of linear discrete time systems using the instrumental variable method, *IEEE Transactions on Automatic Control* AC-12 (6) (1967) 707–718.
- [31] C. Canudas de Wit, Recursive estimation of the continuous-time process parameters, in: *25th Conference on Decision and Control*, Athens, Greece, 1986, pp. 2016–2020.

- [32] A. Janot, P. C. Young, M. Gautier, Identification and control of electromechanical systems using state-dependent parameter estimation, *International Journal of Control* 90 (4) (2017) 643–660.
- [33] A. Bittencourt, A. Isakson, D. Peretzki, K. Forsman, An algorithm for finding process identification intervals from normal operating data, *Processes - Open Access Journal* 3 (2015) 357–383.
- [34] S. Cao, R. Rhinehart, An efficient method for on-line identification of steady state, *Journal of Process Control* 5 (6) (1995) 363–374.
- [35] T. Söderström, P. Stoica, *Instrumental Variable Methods for System Identification*, Springer-Verlag, 1983.
- [36] J. Rutström, *Simplified Wiener LMS Tracking with Automatic Tuning of the Step-Size*, Ph.D. thesis, Uppsala University (2005).
- [37] J. Lataire, E. Louarroudi, R. Pintelon, Y. Rolain, Benchmark data on a linear time- and parameter-varying system, in: *IFAC Symposium on System Identification*, Beijing, China, 2015.
- [38] P. C. Young, Real-time flow forecasting, in: H. Wheeler, S. Sarooshian, K. D. Sharma (Eds.), *Hydrological modelling in arid and semi-arid areas*, Cambridge University Press, Cambridge, UK, 2008, pp. 113–138.
- [39] R. Tóth, V. Laurain, M. Gilson, H. Garnier, Instrumental variable scheme for closed-loop LPV model identification, *Automatica* 48 (2012) 2314–2320.
- [40] H. Garnier, L. Wang, P. C. Young, Refined instrumental variable identification of continuous-time hybrid box-jenkins models, in: *Identification of Continuous-time Models from Sampled Data*, Vol. H. Garnier and L. Wang (Eds.), Springer, 2008, pp. 91–131.
- [41] H. Garnier, Direct continuous-time approaches to system identification. Overview and benefits for practical applications, *European Journal of Control* 24 (2015) 50–62.

Appendix A. Identification of a CT LTI system

Appendix A.1. Optimal theoretical estimates

If the parameters in (1) are constant, then the system corresponds to the following CT LTI OE representation

$$\mathcal{S} \begin{cases} A_o(p)x(t) = B_o(p)u(t) \\ y(t_k) = x(t_k) + e_o(t_k) \end{cases} \quad (\text{A.1})$$

where $e_o(t_k)$ is a zero-mean DT white noise sequence. The polynomials $A_o(p)$ and $B_o(p)$ are assumed to be relatively coprime and are given by

$$B_o(p) = b_0^o p^{n_b} + b_1^o p^{n_b-1} + \dots + b_{n_b}^o \quad (\text{A.2})$$

$$A_o(p) = p^{n_a} + a_1^o p^{n_a-1} + \dots + a_{n_a}^o \quad (\text{A.3})$$

with $n_a \geq n_b$. System (A.1) can be written as follows

$$y^{(n_a)}(t_k) = \boldsymbol{\varphi}^T(t_k) \boldsymbol{\theta}_o + v_o(t_k) \quad (\text{A.4})$$

where $v_o(t_k) = A_o(p)e_o(t_k)$ and

$$\boldsymbol{\theta}_o = [a_1^o \quad \dots \quad a_{n_a}^o \quad b_0^o \quad \dots \quad b_{n_b}^o]^T \quad (\text{A.5})$$

Our goal is to estimate a model for (A.1). The optimal IV estimate is defined by

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}} \frac{1}{N} \left\| \left[\sum_{k=1}^N F(p) \boldsymbol{\zeta}(t_k) F(p) \boldsymbol{\varphi}^T(t_k) \right] \boldsymbol{\theta} - \left[\sum_{k=1}^N F(p) \boldsymbol{\zeta}(t_k) F(p) y^{(n_a)}(t_k) \right] \right\|_{\mathbf{W}}^2 \quad (\text{A.6})$$

where $\boldsymbol{\zeta}(t_k)$ is the instrument vector and $F(p)$ is a stable prefilter and $\|x\|_{\mathbf{W}}^2 = x^T \mathbf{W} x$, with \mathbf{W} a positive definite weighting matrix.

If $\mathcal{S} \in \mathcal{M}$, the estimates (A.6) are consistent under the following conditions [35]⁶:

- C1. $\bar{\mathbb{E}}\{F(p)\boldsymbol{\zeta}(t_k)F(p)\boldsymbol{\varphi}^T(t_k)\}$ is full column rank.

⁶The notation $\bar{\mathbb{E}}[\cdot] = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=1}^N \mathbb{E}[\cdot]$ is adopted from the prediction error framework of [7].

$$\text{C2. } \bar{\mathbb{E}}\{F(p)\zeta(t_k)F(p)v_o(t_k)\} = 0.$$

The instrument $\zeta(t_k)$ can be computed in different ways. Basically, it is defined such that the filtered instrument $F(p)\zeta(t_k)$ is uncorrelated with the filtered noise $F(p)v_o(t_k)$, which corresponds to condition C2.

Optimal estimates, *i.e.* unbiased and minimum variance estimates, are obtained if the following additional conditions are satisfied (see [39] and references therein)

$$\text{C3. } \mathbf{W} = \mathbf{I}$$

$$\text{C4. } F(p) = \frac{1}{A_o(p)}$$

C5. The instrument vector is computed using the auxiliary model

$$\zeta(t_k) = [-x^{(n_a-1)}(t_k) \quad \dots \quad -x(t_k) \quad u^{(n_b)}(t_k) \quad \dots \quad u(t_k)]^T \quad (\text{A.7})$$

where

$$x(t) = \frac{B_o(p)}{A_o(p)}u(t) \quad (\text{A.8})$$

Appendix A.2. Optimal estimates in practice: the SRIVC method

The optimal estimate (A.6) cannot be obtained in practice since it requires knowledge of the true system. The practical and optimal solution is the SRIVC method. This is an iterative approach, where both the prefilter and the instrument are updated in each iteration based on the estimates obtained at the previous iteration. The SRIVC estimate at the i th iteration is given by [SRIVC : see e.g. 40, 3, 41]

$$\hat{\theta}^i = \left[\sum_{k=1}^N \zeta_f(t_k, \hat{\theta}^{i-1}) \varphi_f^T(t_k, \hat{\theta}^{i-1}) \right]^{-1} \cdot \left[\sum_{k=1}^N \zeta_f(t_k, \hat{\theta}^{i-1}) y_f^{(n_a)}(t_k, \hat{\theta}^{i-1}) \right] \quad (\text{A.9})$$

where

$$y_f^{(n_a)}(t_k, \hat{\theta}^{i-1}) = p^{n_a} F(p, \hat{\theta}^{i-1}) y(t_k) \quad (\text{A.10a})$$

$$\varphi_f^T(t_k, \hat{\theta}^{i-1}) = F(p, \hat{\theta}^{i-1}) \varphi^T(t_k) \quad (\text{A.10b})$$

with

$$F(p, \hat{\theta}^{i-1}) = \frac{1}{\hat{A}(p, \hat{\theta}^{i-1})} \quad (\text{A.11})$$

The filtered instrument $\zeta_f(t_k, \hat{\theta}^{i-1})$ is defined through the instrument

$$\zeta(t_k, \hat{\theta}^{i-1}) = [-\hat{x}^{(n_a-1)}(t_k) \quad \dots \quad -\hat{x}(t_k) \quad u^{(n_b)}(t_k) \quad \dots \quad u(t_k)]^T \quad (\text{A.12})$$

with $\hat{x}(t_k)$ defined by the auxiliary model

$$\hat{x}(t_k) = \frac{\hat{B}(p, \hat{\theta}^{i-1})}{\hat{A}(p, \hat{\theta}^{i-1})} u(t_k) \quad (\text{A.13})$$

Then,

$$\begin{aligned} \zeta_f(t_k, \hat{\theta}^{i-1}) &= F(p, \hat{\theta}^{i-1}) \zeta(t_k, \hat{\theta}^{i-1}) \\ &= [-\hat{x}_f^{(n_a-1)}(t_k) \quad \dots \quad -\hat{x}_f(t_k) \quad u_f^{(n_b)}(t_k) \quad \dots \quad u_f(t_k)]^T \end{aligned} \quad (\text{A.14})$$