

CONTSID: a Matlab toolbox for standard and advanced identification of black-box continuous-time models

H. Garnier*, M. Gilson*

* *University of Lorraine, CNRS, CRAN, F-54000 Nancy, France*
{*hugues.garnier, marion.gilson*}@univ-lorraine.fr

Abstract: The CONTinuous-Time System Identification (CONTSID) toolbox provides MATLAB functions for estimating continuous-time black-box models of dynamical systems from measured data without having to fully characterize the mathematics governing the system behavior. The toolbox includes tools for standard identification of linear continuous-time models such as simple process, transfer functions and state-space models. The toolbox also provides algorithms for more advanced identification such as errors-in-variable (EIV) and closed-loop model estimation or to capture nonlinear system dynamics. This paper presents an overview of the main features of the latest release of the CONTSID toolbox and outlines some recent developments for on-line parameter and time-delay system estimation.

Keywords: black-box model, continuous-time model, discrete-time data, irregularly sampled data, MATLAB toolbox, parametric models, software tools, system identification.

1. INTRODUCTION

The CONTSID toolbox to be run with MATLAB was the first toolbox entirely dedicated to continuous-time (CT) model identification from sampled data. It was first released in 1999 (Garnier and Mensler, 1999) at a time where discrete-time model identification was the classical approach. Fortunately, things have recently changed and continuous-time model identification has now taken over discrete-time model identification as exemplified by the more pronounced role of continuous-time model in the System identification toolbox (Ljung and Singh, 2012). One of the clear reasons is coming from the fact that control scientists and engineers have a better understanding and every-day practice of continuous time models, while they are less familiar with input/output polynomial black-box models such as discrete-time ARX, ARMAX or Box-Jenkins models.

The CONTSID toolbox has been updated on a regular basis since its first release, mainly at IFAC SYSID Symposia (see *e.g.* Padilla et al. (2015) and references herein). To facilitate its use, it has been designed to be an add-on to the MATLAB System Identification toolbox (Ljung et al., 2018). It exploits the same syntax, data and model objects.

The toolbox provides standard parametric estimation techniques such as subspace and prediction-error minimization (Garnier and Wang, 2008). However, in the CONTSID toolbox, there is a special focus towards instrumental variable (IV)-based estimation methods

(Garnier, 2015). The identification process is preferably executed in the command-line even though there is also an app that lets the user analyse the experimental data, identify and evaluate linear models in an easy way. The toolbox includes many demonstration program to illustrate its use and, last but not least, it can be freely downloaded from:

www.cran.univ-lorraine.fr/contsid/

The CONTSID toolbox includes tools for basic identification of linear black-box continuous-time models such as:

- Identification of simple (low-order) process models;
- Identification of transfer function models;
- Identification of input/output black-box polynomial models such as autoregressive (CARX), output-error (COE) and Box-Jenkins (CBJ) models;
- Identification of state-space models with free or canonical parametrizations;
- Identification from time-domain response data;
- Identification from frequency-domain response data.

The CONTSID toolbox also includes tools for more advanced identification such as:

- Identification from irregularly sampled data;
- Identification of errors-in-variables (EIV) models;
- Closed-loop model identification;
- Identification of nonlinear block-oriented (Hammerstein and Hammerstein-Wiener) models;
- Identification of linear parameter varying (LPV) input/output models;
- On-line identification for tracking time-varying system dynamics.

The latest release of the CONTSID toolbox includes the following new developments:

- It supports new routines for estimating transfer function plus time-delay models;
- It supports new on-line estimation methods for tracking time-varying parameter (TVP) models;
- For educational purposes, the demonstration programs have been entirely revisited. The toolbox includes several new built-in-tutorials that illustrate typical identification sessions with the CONTSID toolbox.

The paper is organized in the following way. An overview of the different forms of CT models along with standard and advanced methods available in the toolbox is first presented in Section 2. The recommended sequence for identifying linear models with the CONTSID toolbox is then presented and illustrated on experimental data coming from a flexible robot arm in Section 3. The latest developments for the demonstration programs along with for on-line and time-delay model estimation are described in Sections 4 to 6.

2. OVERVIEW OF THE MAIN CONTSID TOOLBOX COMMANDS

2.1 The system identification workflow

In practice, the common system identification workflow is iterative as shown in Figure 1 (Ljung, 1999). It includes several tasks. Starting from measured input/output data, a set of candidate models is estimated by using suitable identification algorithms. The identified model which produces the best results according to the chosen validation criterion is finally selected. The system identification work-

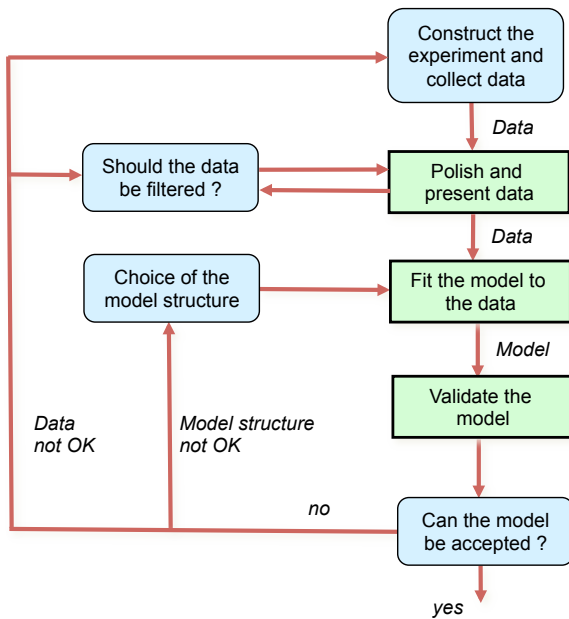


Fig. 1. The system identification procedure

flow is general and pragmatic. It is independent of the chosen discrete-time or continuous-time model parametriza-

tion used, although the latter can present many advantages (see *e.g.* Garnier and Young (2014); Garnier (2015)).

2.2 Main CONTSID toolbox commands

The latest version 7.3 of the CONTSID toolbox offers a variety of parametric model estimation methods for common linear and nonlinear model structures. Tables 1 and 2 summarize the main CONTSID toolbox commands for standard linear model identification and for more advanced identification respectively.

Table 1. Main CONTSID toolbox commands for standard linear model identification

Model type	Estimation commands
Transfer function models	tfstrivc
Process models	procsrivc
Input/output polynomial models	lssvf (CARX models) ivsvf (CARX models) coe (COE models) srivc (COE models) rivc (CBJ models)
State-space models	sidgpmf ssivgpmf

Table 2. Main CONTSID toolbox commands for more advanced identification

Model Type	Estimation commands
Polynomial models	clsrivc
in closed loop	cl2srivc
EIV models	focils
Hammerstein models	hsrivc
Hammerstein-Wiener models	hwsrivc
LPV input/output models	lpvsrivc

3. TUTORIAL INTRODUCTION TO THE CONTSID

As seen before, system identification is an iterative process, where you identify models with different structures from data and compare model performance. You start by estimating the parameters of simple model structures. If the model performance is poor, you gradually increase the complexity of the model structure. Ultimately, you choose the simplest model that best describes the dynamics of your system. Another reason to start with simple model structures is that higher-order models are not always more accurate. Increasing model complexity increases the uncertainties in parameter estimates and typically requires more data (which is common in the case of nonlinear models).

3.1 Recommended model identification sequence

The recommended sequence for identifying parametric models with the CONTSID toolbox commands is the following:

- (1) Start by trying the simplest linear model structures such as transfer function, simple process and COE polynomial, models. Use the **tfstrivc**, **procsrivc** and **srivc** commands, respectively. Use the **srivcstruc** command to select the appropriate model order.

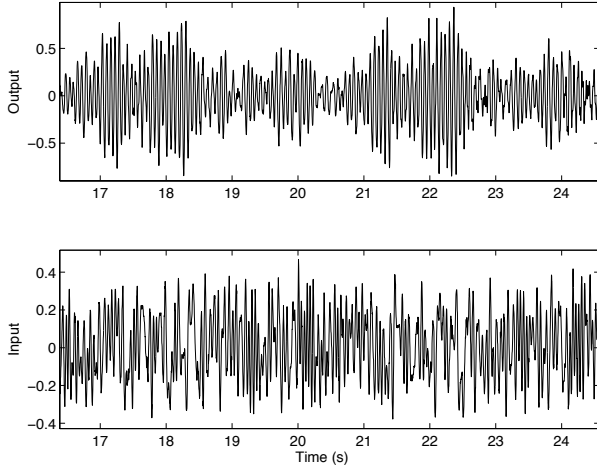


Fig. 2. Input/output data over the 3rd-period for the robot arm

- (2) Try continuous-time BJ polynomial models which provide additional flexibility to model the noise contribution. Try several model orders and keep the model orders as low as possible. Use the `srivc` command.
- (3) Try LPV, Hammerstein or Hammerstein-Wiener models which provide additional flexibility to model nonlinear effects. It is more complicated to select the polynomial orders used to capture the nonlinearities. Use the `lpvsrivc`, `hsrivc` and `hwsrivc` commands.

3.2 Identification of a flexible robot arm

A short example will illustrate the typical steps involved in identifying a linear model in the case of a flexible robot arm. The full demonstration program can be run from the demonstration program presented in Section 4.

The behavior of a flexible robot arm was measured by applying controlled torque to the vertical axis at one end of the arm, and measuring the tangential acceleration of the other end. The robot arm is described in more detail in Kollar (1994).

The excitation signal is a multi-sine. The sampling period is set to 2 ms. Measurements are made with anti-aliasing filters. $K = 10$ periods each of length $M = 4096$ are exactly measured and a record of $N = KM = 40,960$ data points is collected. The data set over the 3rd period is displayed in Figure 2. The identification problem is to use part of the data to identify an appropriate model structure from the robot arm data set; to then estimate the parameters that characterize this identified model structure; and, finally to validate the identified model on a section of the data not used for the estimation stage.

The empirical transfer function estimate (ETFE) obtained from the 3rd period data set is displayed in Figure 3. From this figure, one can have a good indication about the model orders of the system. Indeed, one can see from the ETFE that the system has at least 3 resonant modes and 4 zeros in the frequency band $\omega \in [0; 350]$ rad/s.

Different model structures in the range $[\mathbf{n}_b \ \mathbf{n}_f \ \mathbf{n}_k] = [4 \ 4 \ 0]$ to $[7 \ 6 \ 0]$ have been computed by using `srivcstruc`

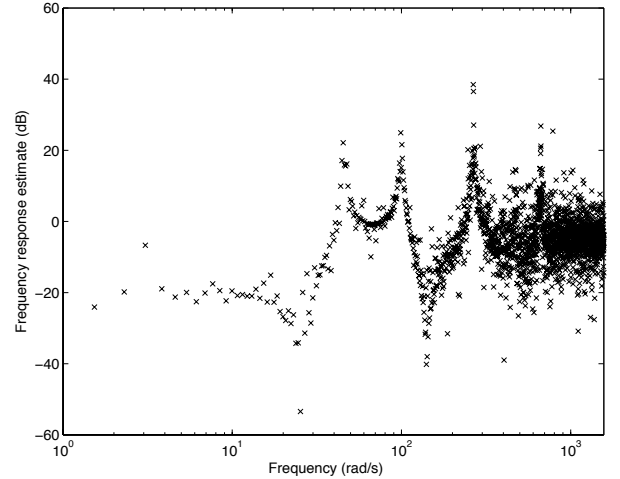


Fig. 3. Empirical transfer function estimate for the robot arm

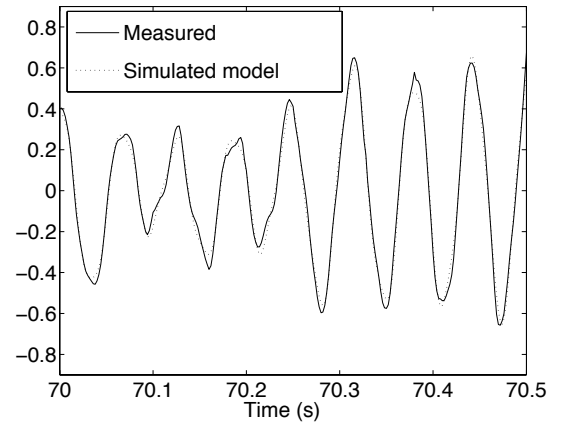


Fig. 4. Cross-validation results on a short section of the 8th-period data set

command for the 3rd period data set. The other data set periods were kept for model validation purposes.

The 7 best models sorted according to YIC (see *e.g.* Garnier (2015)) are given in Table 3. From this table, the first model with $[\mathbf{n}_b \ \mathbf{n}_f \ \mathbf{n}_k] = [6 \ 6 \ 0]$ seems to be quite clear cut (it has the most negative YIC = -9.19, with the highest associated $R_T^2 = 0.907$).

Table 3. Best SRIVC model orders for the robot arm data set

\mathbf{n}_b	\mathbf{n}_f	\mathbf{n}_k	YIC	R_T^2
6	6	0	-9.19	0.907**
4	4	0	-8.56	0.940
7	6	0	-8.03	0.907
5	6	0	-7.41	0.906
5	4	0	-7.01	0.940

The process identification is performed with the `srivc` algorithm on the third-period data set. The identification result is given as the following Laplace transfer function model

$$\hat{G}(s) = \frac{20.87(s - 618.5)(s^2 - 1.698s + 710.6)(s^2 + 8.435s + 2.012e4)}{(s^2 + 1.033s + 2094)(s^2 + 0.9808s + 9905)(s^2 + 2.693s + 7.042e4)}$$

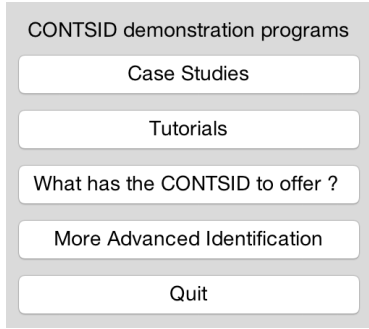


Fig. 5. Main window of the CONTSID demonstration program.

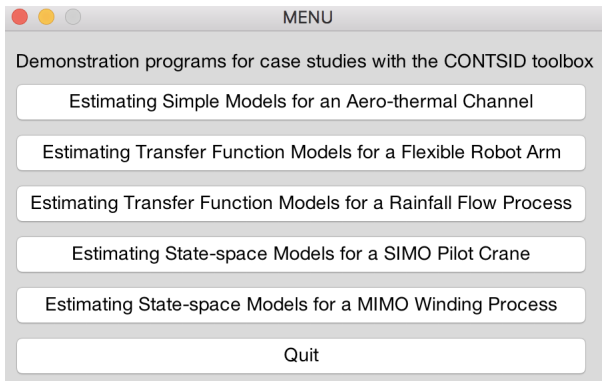


Fig. 6. Sub-menu window to run demos showing some case studies with the CONTSID toolbox.

Figure 4 compares the simulated SRIVC model output with the measured output series, over a short section of 0.4 s for the 8th-period data set. It can be noticed that the simulated output matches the measured data quite well, with $R_T^2 > 0.95$. The estimated linear model is therefore accurate enough here and there is no need to go through steps 2 and 3 of the recommended identification sequence described in Subsection 3.1.

4. NEW DEMONSTRATION PROGRAMS

Demonstration programs are crucial to illustrate the use of new aspects of a toolbox. The demonstration programs for the CONTSID toolbox have been entirely revisited for educational purposes. The new demos can now be executed by typing the following command: `contsid_demo`. This opens a main menu window from which the user is invited to select between four options as shown in Figure 5. The four sub-menu windows from which built-in-tutorials can be run, are displayed in Figure 6 to Figure 9.

5. LATEST DEVELOPMENTS FOR ON-LINE IDENTIFICATION

5.1 Tracking time-varying parameters with the CONTSID new algorithms

In the recent years, there has been a renewed interest in recursive identification, mainly due to an increased demand for fault detection, monitoring and predictive maintenance. On-line recursive estimation of linear time-varying systems usually involves discrete-time models. Continuous-time models have received less attention,

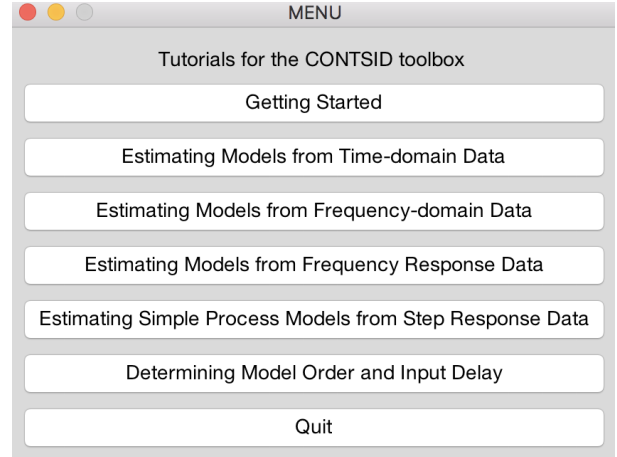


Fig. 7. Sub-menu window to run demos illustrating standard linear model identification with the CONTSID toolbox.

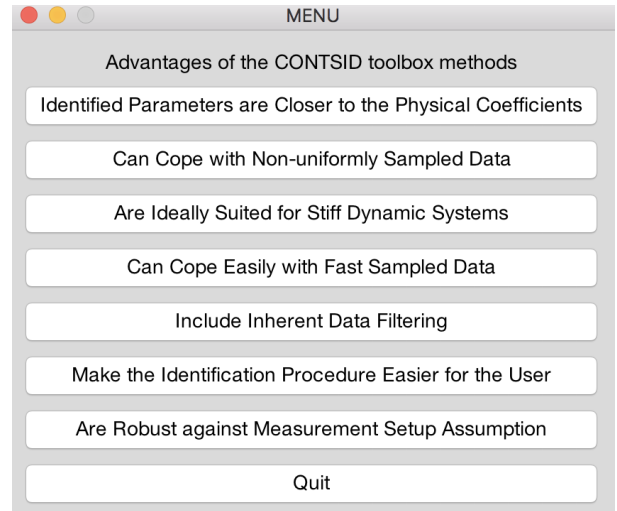


Fig. 8. Sub-menu window to run demos showing the advantages of the CONTSID methods.

partly due to the increased complexity associated with the need to handle the time-derivatives of the input and output variables. Recent research have led to the development of new recursive algorithms for on-line estimation of slowly linear time-varying continuous-time models (Padilla et al., 2016). Table 4 summarizes the recently developed algorithms now available in the CONTSID toolbox.

Two adaptation mechanisms are available for each of the algorithm (Padilla et al., 2016):

- Kalman filter where it is assumed that the variations of the parameters are described by a stochastic random walk model;
- Standard forgetting factor.

Table 4. New CONTSID toolbox commands for on-line estimation

Model Type	Estimation commands
Linear polynomial models	<code>rlssvf</code> (CARX models)
	<code>rivsvf</code> (CARX models)
	<code>rsrivc</code> (COE models)

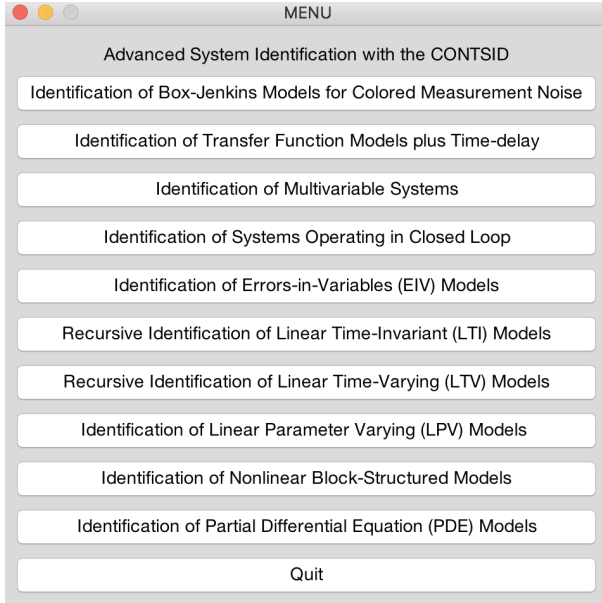


Fig. 9. Sub-menu window to run demos illustrating advanced system identification with the CONTSID toolbox.

5.2 Numerical example for on-line estimation

The use of the new recursive RLSSVF and RSRIVC routines with the Kalman filter adaptation mechanism is illustrated here by a numerical example. The assumed random-walk model presents the advantage of easily handle the case of model parameters with different types of variations. The data generating system is given by

$$\mathcal{S} \begin{cases} (p^2 + a_1^o(t)p + a_2^o(t))x(t) = b_0^o(t)u(t) \\ y(t_k) = x(t_k) + e(t_k) \end{cases} \quad (1)$$

where p denotes the differentiation operator, $a_1^o(t)$ varies slowly between 5 and 45 in a linear fashion, $b_0^o(t)$ remains constant at 200 and $a_2^o(t)$ varies slowly as follows,

$$a_2^o(t) = 160 - 90 \cos(2\pi t/1000)$$

Note that the parameter slow variations are different. $a_1(t)$ varies slowly between 5 and 45 in a linear fashion, $b_0(t)$ remains constant while $a_2(t)$ varies slowly in a sine fashion.

The sampling time is set to 0.01 s and the total simulation time is 1000 s. The input is a PRBS and the DT measurement noise is zero-mean, Gaussian noise with constant variance 0.1. In this example, as a consequence of the time-varying parameters, the DC gain is decreasing towards half of the simulation time; and since the noise variance is kept constant, the signal-to-noise ratio (SNR) is decreasing around half of the simulation time. For some frozen LTI systems obtained every 100 s, the magnitude Bode diagram is plotted in Figure 10. From this figure, it can be noted that not only the steady-state gain of the system is varying but also the system bandwidth. The bandwidths of these frozen systems are shown in Figure 11 where it can be noted also that the ratio between the maximum and minimum bandwidths is nearly 10, *i.e.* the bandwidth variation is relatively large over the total simulation time.

The simple RLSSVF algorithm is first tested. This routine assumes a CARX model structure and so the triad which defines the number of parameters to be estimated for each

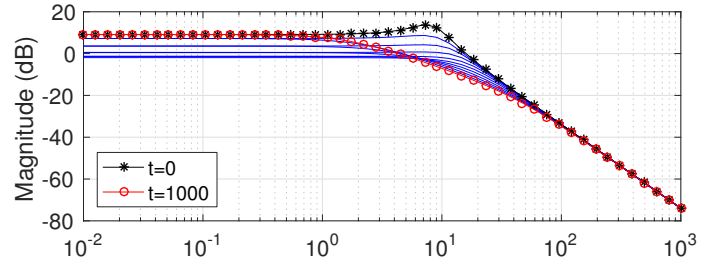


Fig. 10. Magnitude Bode diagram for some frozen systems obtained every 100 s. The system responses for $t = 0$ and $t = 1000$ s are highlighted.

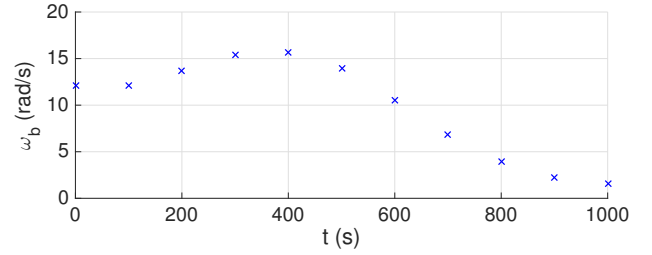


Fig. 11. Bandwidth of frozen systems obtained every 100 s.

polynomial needs to be set in the appropriate way. Q_{nvr} is set as a diagonal matrix with diagonal elements equal to 10^{-5} , 10^{-4} and 10^{-10} corresponding to a_1^o , a_2^o and b_0^o , respectively.

```
nn=[2 1 0]; % [na nb nk]
Qn=diag([1e-5 1e-4 1e-10]);
```

The value corresponding to b_0^o is 10^{-10} , because it is assumed known that this parameter is constant.

For the SVF, $\lambda_{svf} = 16$ rad/s is chosen, *i.e.* a value larger than the maximum bandwidth.

```
th1 = rlssvf(data,nn,'lambda_svf',16,'...
adm','kf','adg',Qn);
```

With these settings, the RLSSVF estimates are displayed in Figure 12 for noise-free and noisy output measurement. In the former case, since there is no noise, RLSSVF performs very well and it is difficult to distinguish between the true and estimated parameters. However, in the noisy case, RLSSVF does not provide reasonable parameter estimates. Note that the relative errors of the three estimates are similar but this is difficult to see because the scales for the y -axes are different. It can be observed from Figure 12 that the estimation error variance increases at the central portion of the plot. This is because the SNR is decreasing along with the time while the system bandwidth reduces when the time increases (which would also require λ_{svf} to be lower).

To get a better tracking of the time-varying parameters in the noisy situation, one can use the recursive version of simplified refined instrumental variable method for continuous-time model estimation (RSRIVC) as suggested in Padilla et al. (2016). Notice that the routine assumes an COE model, then nn and Qn must be redefined.

```
nn = [1 2 0]; % [nb nf nk]
Qn = diag([1e-10 1e-5 1e-4]);
```

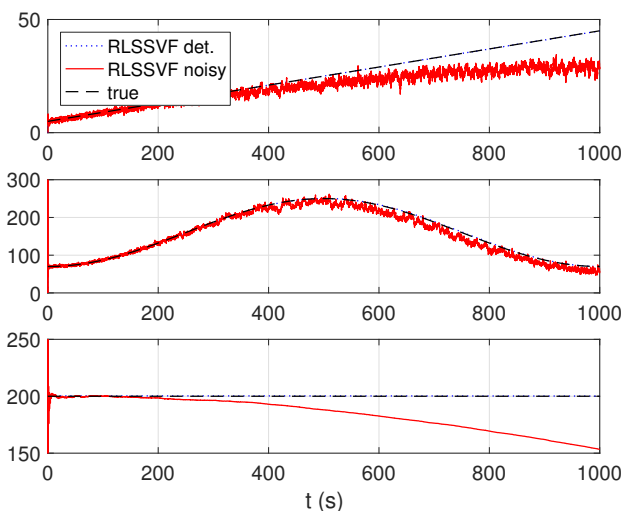


Fig. 12. True and RLSSVF parameter estimates for the deterministic (det.) case and noisy case (the estimates for the deterministic case are matching the true values).

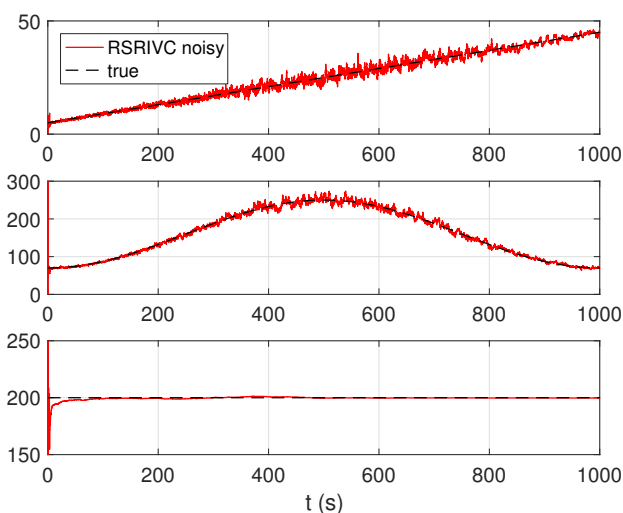


Fig. 13. True and RSRIVC parameter estimates for the noisy case.

```
th2 = rsrivc(data,nn,'lambda_svf',16,...
'adm','kf','adg',Qn)
```

The RSRIVC estimates are displayed in Figure 13. The improvement in the estimation performance is clear from Figure 13, where the proposed adaptive RSRIVC algorithm is able to track the parameter changes very well. Naturally, the estimation error variance changes discussed above for the RLSSVF case also occur here.

6. LATEST DEVELOPMENTS FOR TIME-DELAY SYSTEM IDENTIFICATION

Time-delays are a common feature of many industrial processes. Obtaining an accurate estimate of the time-delay is of crucial importance for controller design, in order to achieve good control system performance.

Recent developments have led to extend the simplified refined instrumental variable method for CT models (SRIVC) to handle arbitrarily time-delayed systems (Chen et al., 2018). The developed method, abbreviated as TFSRIVC (SRIVC for transfer function models) is based on the principle of variable projection, combining an IV method for the rational model parameters and an adaptive search for the time-delay. The new TFSRIVC method has shown to be effective in terms of numerical simulations (Chen et al., 2018). New demos have been added to illustrate the recent developments for transfer function plus time-delay model identification with the CONTSID toolbox.

7. CONCLUSION

The CONTSID toolbox is dedicated to the direct identification of continuous-time models from sampled data. The toolbox is in continual development. The new release includes new demonstration programs ; it also supports new routines for tracking time-varying parameters and for identifying transfer function plus time-delay models. Future developments are aiming at improving the graphical user interface (app) to facilitate the use of the CONTSID toolbox methods for those who are not familiar with the Matlab commands.

REFERENCES

- Chen, F., Zhuang, X., Garnier, H., and Gilson, M. (2018). Issues in separable identification of continuous-time models with time-delay. *Automatica*, 94, 258–273.
- Garnier, H. (2015). Direct continuous-time approaches to system identification. overview and benefits for practical applications. *European Journal of Control*, 24, 50–62.
- Garnier, H. and Mensler, M. (1999). CONTSID: a continuous-time system identification toolbox for Matlab. In *5th European Control Conference (ECC'99)*. Karlsruhe (Germany).
- Garnier, H. and Wang, L. (eds.) (2008). *Identification of continuous-time models from sampled data*. Springer-Verlag, London.
- Garnier, H. and Young, P. (2014). The advantages of directly identifying continuous-time transfer function models. *International Journal of Control*, 87(7-9), 1319–1338.
- Kollar, I. (1994). *Frequency Domain System Identification Toolbox Users's Guide*. The Mathworks, Inc., Mass.
- Ljung, L. (1999). *System identification. Theory for the user*. Prentice Hall, Upper Saddle River, 2nd edition.
- Ljung, L., Ozdemir, A., and Singh, R. (2018). On-line features in the system identification toolbox. In *18th IFAC Symposium on System Identification (SYSID'2018)*. Stockholm (Sweden).
- Ljung, L. and Singh, R. (2012). Version 8 of the system identification toolbox. In *16th IFAC Symposium on System Identification (SYSID'2012)*. Brussels (Belgium).
- Padilla, A., Garnier, H., and Gilson, M. (2015). Version 7.0 of the CONTSID toolbox. In *17th IFAC Symposium on System Identification (SYSID'2015)*. Beijing (China).
- Padilla, A., Garnier, H., Young, P., and Yuz, J. (2016). Real-time identification of linear continuous-time systems with slowly time-varying parameters. In *IEEE Control and Decision Conference (CDC'2016)*. Las Vegas (USA).