Teaching data-based continuous-time model identification with the CONTSID toolbox

Hugues Garnier

Teaching Data-based Continuous-time Model Identification with the CONTSID Toolbox

Hugues GARNIER[1]

[1]Centre de Recherche en Automatique de Nancy (CRAN), Nancy-University, CNRS,
BP 70239, 54506 Vandoeuvre-les-Nancy Cedex, France
hugues.garnier@cran.uhp-nancy.fr

Résumé— **This paper discusses experience to introduce data-based continuous-time model identification to engineering students. Specifically, the paper describes how the CONtinuous-Time System IDentification (CONTSID) toolbox and its graphical user interface to be run with Matlab are used to teach time-domain identification methods for estimating continuous-time models directly from sampled data. The educational focus is to mix theoretical aspects with hands-on experience at numerous computer sessions dealing with simulated and real data examples.**

Mots-clés— **continuous-time model, discrete-time data, education, Matlab toolbox, software tools, system identification**

I. Introduction

The identification of continuous-time (CT) models is a problem of considerable importance that has applications in virtually all disciplines of science. Early research in system identification focussed on identification of CT models from CT data. Subsequently, however, rapid developments in digital data acquisition and computers have resulted in attention being shifted to the identification of discrete-time (DT) models from sampled data, leading to a complete dominance of these methods over continuous-time model identification techniques. Much less attention has been therefore devoted to CT modelling from DT data and many practitioners appear unaware that such alternative methods not only exist but may be better suited to their modelling problems [1].

The relevance of direct continuous-time model identification methods has been recently illustrated with extensive numerical simulation [2], [3]. A toolbox for Matlab is now available : the CONtinuous-Time System IDentification (CONTSID) toolbox which supports continuous-time transfer function or state-space model identification directly from time-domain sampled data, without requiring the determination of a discrete-time model [4], [1].

While undergraduate and graduate courses on DT model model identification (mainly based on Ljung's book [5] are taught in many universities, there is a lack of courses in data-based CT modelling. For some years now, we have introduced a course at Nancy-University which aims at introducing CT model identification to engineering students. CT models are very appealing to engineers since they can be directly and easily interpreted in a physical manner.

Some of the topics that we have been able to introduce in the curriculum include:

- a presentation of the main time-domain identification methods for estimating continuous-time models directly from sampled data;
- an introduction to computer-aided tools, notably graphical tools for CT model identification relying of the CONTSID toolbox for Matlab;
- the mix of theoretical aspects with hands-on experience at numerous computer sessions dealing with simulated and real data examples.

This paper discusses this experience to introduce data-based continuous-time model identification to engineering students.
It is organized as follows. Section 2 outlines the main steps of the procedure for direct continuous-time model identification. An overview of the course organisation is given in Section 3. The main features of the CONTSID toolbox and its graphical user interface (GUI) are then presented in Section 4. Finally, typical results obtained by the students on a complex flexible robot arm are described in Section 5.

II. General procedure for data-based modelling

The procedure to determine a continuous-time model of a dynamical system directly from observed time-domain input-output data is close to the one used for discrete-time model identification and involves three basic ingredients:

- the time-domain sampled input-output data;
- a set of candidate models (the model structure);
- a criterion to select a particular model in the set, based on the information in the data (the identification method).

The identification procedure consists then in repeatedly selecting a model structure, computing the best model in the chosen structure, and evaluating the identified model. The iterative procedure involves the following steps:
1. Design an experiment and collect input-output data from the process to be identified.
2. Examine the data. Remove trends and outliers, and select useful portions of the original data.
3. Select and define a model structure (a set of candidate system descriptions) within which a model is to be estimated.
4. Compute the best model in the model structure according to the input-output data and a given criterion of fit.
5. Examine the obtained model properties.

If the model is good enough, then stop; otherwise go back to step 3 to try another model set. Possibly also try other estimation methods (step 4) or work further on the input-output data (steps 1 and 2).

III. The course for engineering students

The course on continuous-time model identification given at Nancy-University is intended for engineering students who have already a good background in system identification. The course organisation is as follows :

1. Introduction to data-based CT modelling
2. Parameter estimation of linear CT models
3. Model structure estimation and validation
4. Relevance of CT model identification
5. Software aspects - The CONTSID toolbox
6. Simulated and real data application examples

The 18-hour course is typically laid out over 9 weeks, with one two-hour lecture per week for the first five weeks, followed by four two-hour computer sessions. The textbook used is [1] and the course outline uses Chapters 1, 4 and 9 of this textbook.

In the course, we consider two specific parametric estimation methods that exemplify the historical development of direct CT identification. Initially, most methods were largely deterministic, in the sense that they did not explicitly model the additive noise process nor attempt to quantify the statistical properties of the parameter estimates. Instead, consistent estimates were obtained by using basic Instrumental Variable (IV) methods. One deterministic approach of this type, known as the state-variable filter (SVF) method is reviewed first, with the aim of highlighting some of the peculiarities that occur in comparison with DT model identification. Then, a more sophisticated IV (SRIVC) method for direct CT stochastic model identification is introduced in order to demonstrate the advantages of the stochastic model formulation.

The educational focus is to mix theoretical aspects introduced during the lecture with hands-on experience during the computer sessions where simulated and real data are used to apply the identification procedure. Identification from simulated data is not as challenging as from real-life data but it allows the students to gain some familiarity with the parametric model estimation and validation techniques.

The students can easily get confused by the iterative identification procedure which includes different methods in every steps. It is therefore important to package the identification tools in a user-friendly way. An attempt to do that was carried out with the CONTinuous-time System IDentification (CONTSID) toolbox for MATLAB®. It is described in the next section with a focus on the graphical user interface (GUI) which allows the students to perform easily data analysis, model parameter estimation and validation by mouse-click operations.

IV. Software aspects - The CONTSID toolbox

A. The CONTSID toolbox

The key features of the CONTSID toolbox are [4], [6]:

- it supports most of the time-domain methods developed over the last thirty years [7] for identifying linear dynamic continuous-time parametric models from measured input/output DT data;
- it provides transfer function and state-space model identification methods for single-input single-output and multiple-input multiple-output systems, including both traditional and more recent approaches;
- it can handle mild irregularly sampled data in a straightforward way;
- it may be seen as an add-on to the system identification (SID) toolbox. To facilitate its use, it has been indeed given a similar setup to the SID toolbox;
- it provides a flexible graphical user interface (GUI) that lets the user analyse the experimental data, identify and evaluate models in an easy way;
- It can be freely downloaded from http://www.cran.uhp-nancy.fr/contsid/

B. The GUI for the CONTSID toolbox

The graphical user interface for the CONTSID toolbox provides a main window, as shown in Figure 1, which is divided into three basic parts:

- a data panel on the left part where data sets can be imported, plotted, pre-treated and selected;
- a model estimation panel in the middle where different model structures and identification methods to directly estimate a CT transfer function model can be tested;
- a model validation panel in the right part where basic properties of the identified model can be examined.

The CONTSID GUI can be started by typing `contsidgui` in the Matlab command window.

C. The data panel

As shown in Figure 1, the GUI lets the user to define two data sets: one for identifying the model and one for cross-validation purpose.

C.1 Importing measured data.

By clicking on the *Load data* button, you can import time-domain sampled data from a *.mat* file for systems with multiple input and output channels. From this window, you can select the input and output variables, specify the type of sampling scheme (regular or irregular), the sampling time (T_s) and the assumption on the input intersample behavior (piecewise constant (ZOH) or continuous).

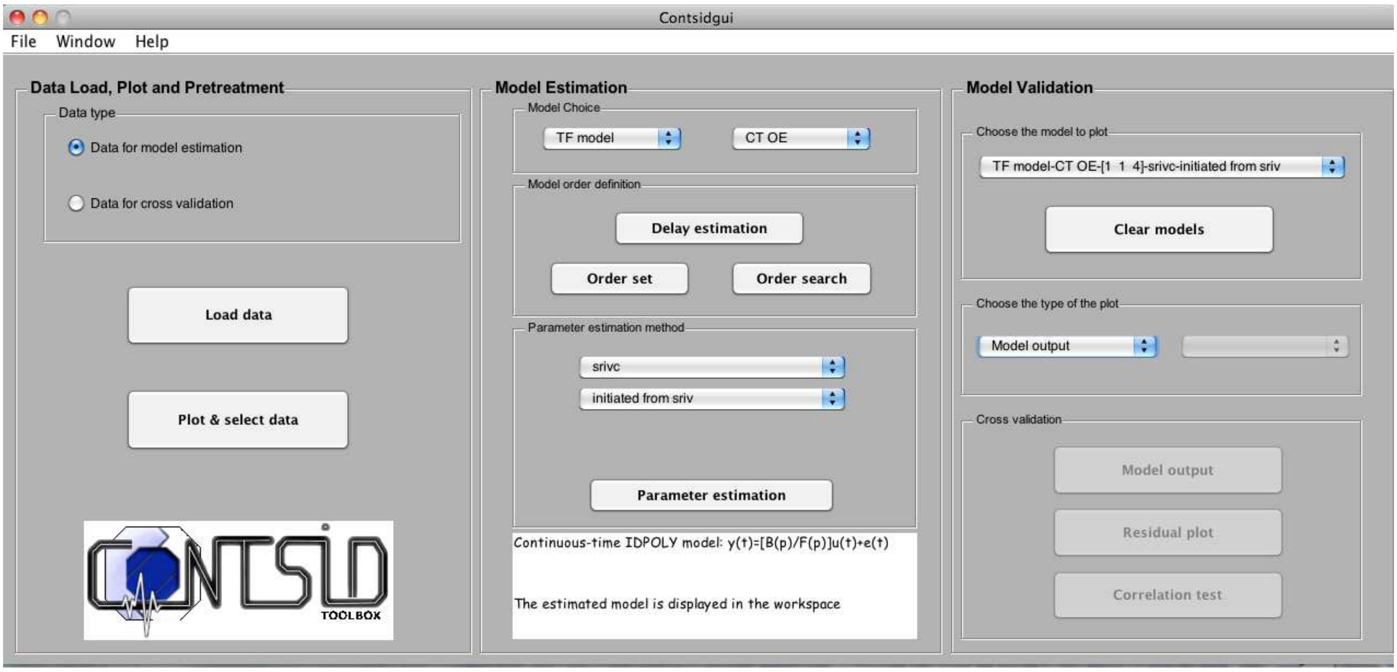
C.2 Preprocessing and selecting observed data.

After the data has been imported, you can apply basic operations for data analysis and preprocessing. An example of the window obtained after a click on the button *Plot and select data* is displayed in Figure 2. It shows the plot of the input/output data. This window also allows the preprocessing of data including offset and drift removal and the display of the results after the operation.

It is often the case that the whole data record is not suitable for identification. This is mainly for two reasons:

- these vectors include erroneous values which it is essential to eliminate;
- if only one data set is available, it is advisable to divide the data set in two parts, the first for model estimation purpose and the second reserved for cross-validation purpose.

The *Cursor selection* button allows the insertion of two vertical axes on the output plot which can be used to define the portion of the measured data that you want to select.



Model Choice: TF model, CT OE

Parameter estimation method: srivc, initiated from sriv

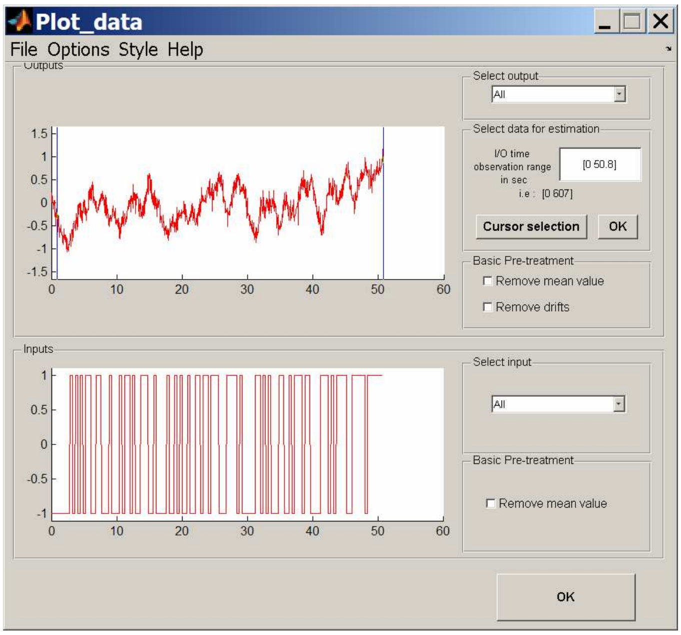
Choose the model to plot: TF model-CT OE-[1 1 4]-srivc-initiated from sriv

Choose the type of the plot: Model output

Continuous-time IDPOLY model: y(t)=[B(p)/F(p)]u(t)+e(t)

The estimated model is displayed in the workspace

Fig. 1. The main window of the CONTSID GUI



Select output: All

I/O time observation range in sec: [0 50.8]
i.e : [0 607]

Select input: All

Fig. 2. Data plot and pretreatment window

D. Model estimation panel

While the CONTSID toolbox supports transfer function and state-space model identification methods, the GUI lets you estimate continuous-time polynomial (transfer function) models only, using the following two predefined model structures:

$$A(p)y(t_k) = B(p)u(t_k) + e(t_k) \qquad (1)$$

$$y(t_k) = \frac{B(p)}{F(p)}u(t_k) + e(t_k) \qquad (2)$$

where p denotes the differential operator, *i.e.* $p := \frac{d}{dt}$; $u(t_k)$ and $y(t_k)$ represent the input and output signals at time-instant $t = kT_s$ respectively; $e(t_k)$ is a DT white Gaussian sequence; $A(p)$, $B(p)$ and $F(p)$ are polynomials in p. Equation (1) defines what is called a CT-ARX model structure while equation (2) defines a so-called CT-OE model structure. The user is thus invited to choose the type and the structure of his model in the two unrolling menus at the top of the *model estimation* panel, as shown in Figure 1.

After selecting the model structure, the user has to specify the polynomial orders and the time-delay of the model to be estimated.

A first option is to deduce an estimate of the number of samples for the time-delay from an estimation of the impulse response by correlation analysis.

Then, if the TF model order is not known *a priori*, the *Order search* button allows the user to automatically search over a whole range of different model orders, as illustrated in Figure 3. The user can choose several available criteria to sort and display the estimation results in the Matlab workspace. From these results, the user can select the best model orders and then set the order of the final model to be estimated by clicking on the *Order set* button from the main window

Once we have set the number of samples for the time-delay and the number of coefficients for the polynomial model, the model parameters can then be estimated by using one of the available parametric methods chosen from an unrolling menu:

- in the case of a CT-OE model structure, the user can choose to use the continuous-time output error (COE) method or the simplified refined instrumental variable (SRIVC) method ;
- in the case of a CT-ARX model structure, the user can select one of the six linear transform-based methods which have proven successful. These linear transforms are coupled with simple least squares or auxiliary model-based instrumental variable methods. They require all, a design

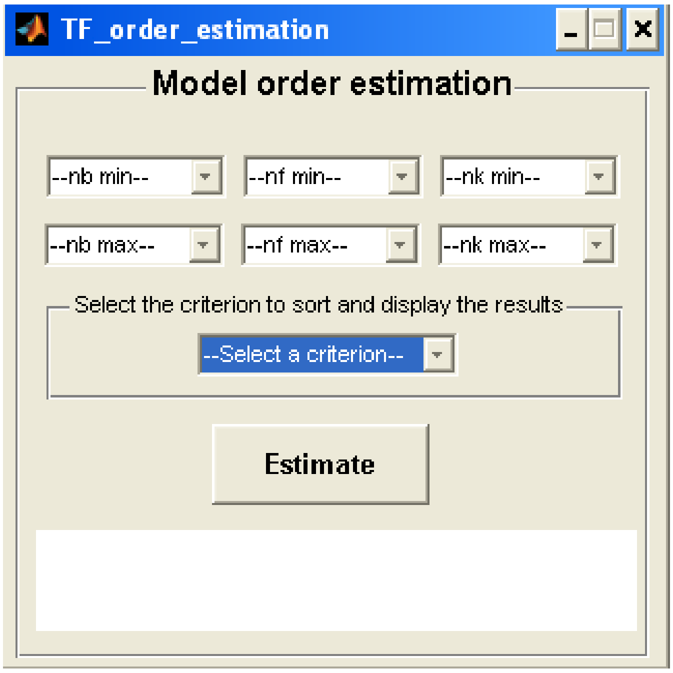
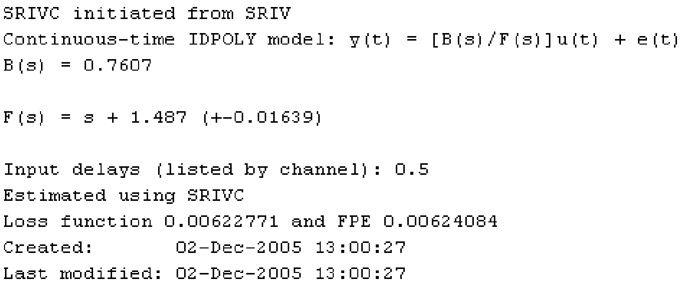


Fig. 3. Model order estimation window



```
SRIVC initiated from SRIV
Continuous-time IDPOLY model: y(t) = [B(s)/F(s)]u(t) + e(t)
B(s) = 0.7607

F(s) = s + 1.487 (+-0.01639)

Input delays (listed by channel): 0.5
Estimated using SRIVC
Loss function 0.00622771 and FPE 0.00624084
Created:       02-Dec-2005 13:00:27
Last modified: 02-Dec-2005 13:00:27
```

Fig. 4. Parameter estimation results

parameter to be specified by the user [7]. The design parameter should be chosen in order to emphasize the frequency band of interest.
At the end, the students are invited to use, at a first choice, the SRIVC method since it can be automatically initiated and has been proven to be powerful in practice. Once chosen the parameter estimation method, the identified model is displayed in the Matlab command window, as illustrated in Figure 4, after a click on the *Parameter estimation* button.

E. Model validation panel

Once a model is estimated, it will appear in the dropdown menu located at the top part of the *Model validation* panel. Several basic model properties can then be evaluated from an unrolling menu by using first the data that were used for model identification:
- *model output comparison*: plots and compares the simulated model output with the measured output, as illustrated in Figure 5. This indicates how well the system dynamics are captured;
- *residual plot*: displays the residuals, *i.e.* the output simulation error;
- *transient response*: displays the model response to an impulse or step excitation signal;

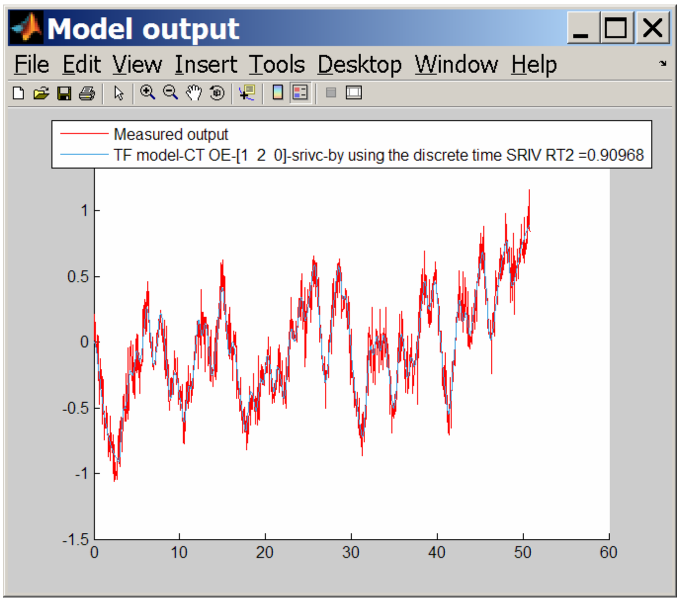


Fig. 5. Measured and simulated model outputs

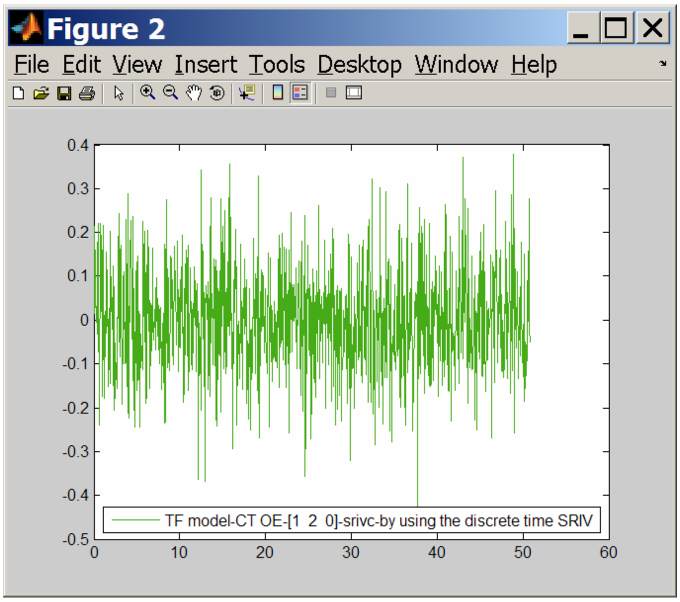


Fig. 6. Residual plot

- *frequency response*: displays the Nyquist or Bode plots to show damping levels and resonance frequencies;
- *zeros and poles*: plots the poles and zeros of the identified models and tests for zero-pole cancelation indicating overparameterized modelling;
- *correlation test*: displays the sample auto-covariance function of the residuals and the sample cross-covariance function between the excitation signal and the residuals, as illustrated in Figure 7.

If a cross-validation data set is available, then traditional cross-validation tests consist in comparing the measured and simulated model outputs and analyzing the residuals.

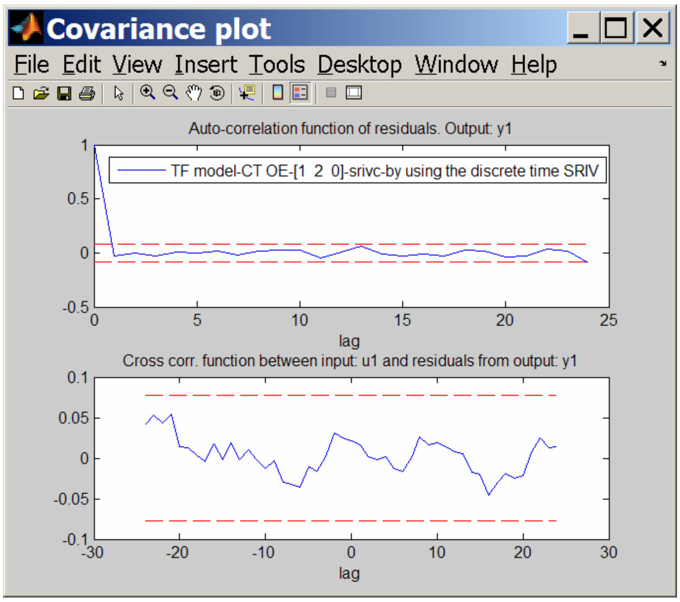

Covariance plot
File Edit View Insert Tools Desktop Window Help
Auto-correlation function of residuals. Output: y1
TF model-CT OE-[1 2 0]-srivc-by using the discrete time SRIV
1
0.5
0
-0.5
0
5
10
15
20
25
lag
Cross corr. function between input: u1 and residuals from output: y1
0.1
0.05
0
-0.05
-0.1
-30
-20
-10
0
10
20
30
lag

Fig. 7. Correlation test

V. Complex Flexible Robot Arm

Real-life data analysis provide students with an appreciation for the diversity of tasks and difficulties that must be dealt with by engineers when building a continuous-time model from measured data.

At the end of the course, the students are asked to form teams to deal with real data sets coming from different application areas: mechanical, robotics, environment. They are asked to apply the whole data-based CT modelling procedure and write a report summarizing their identification results. Typical results obtained by the students on a complex flexible robot arm are described in this section.

Process Description and Modelling Purpose

The robot arm is installed on an electrical motor. The modelling aim is here to design a control law based on a model between the measured reaction torque of the structure on the ground to the acceleration of the flexible arm. The robot arm is described in more detail in [8].

Experiment Design

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.

Model Order Determination

The empirical transfer function estimate (ETFE) obtained from the 3rd period data set is displayed in Figure 8. From this figure, we can have a good indication about the model orders of the system. Indeed, we 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 $[\mathtt{n_b}\ \mathtt{n_f}\ \mathtt{n_k}] =$ [4 4 0] to [7 6 0] have been computed for the 3rd period data set. The other data set periods were kept for model validation purposes.

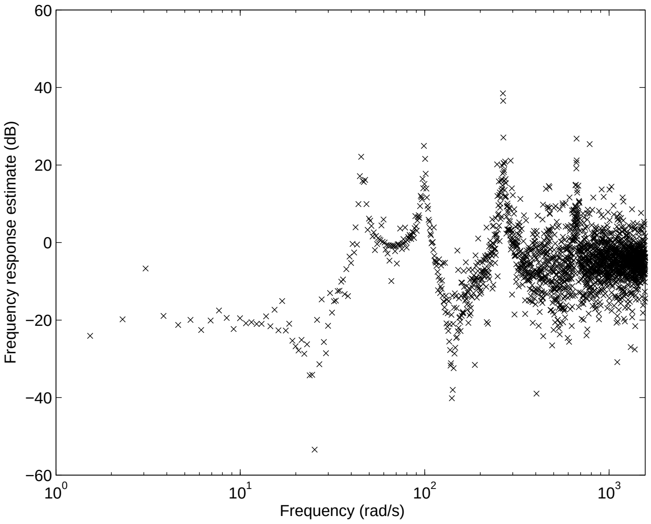


Fig. 8. Empirical transfer function estimate for the robot arm

The 7 best models sorted according to Young's information criterion (YIC) (see e.g. [1]) are given in Table I. From this table, the first model with $[\mathtt{n_b}\ \mathtt{n_f}\ \mathtt{n_k}]$= [6 6 0] seems to be quite clear cut (it has the most negative YIC=−9.19, with the highest associated coefficient of determination $R_T^2 = 0.977$).

TABLE I
Best SRIVC model orders for the robot arm data set

$\mathtt{n_b}$	$\mathtt{n_f}$	$\mathtt{n_k}$	YIC	R_T^2
6	6	0	−9.19	0.977**
4	4	0	−8.56	0.940
7	6	0	−8.03	0.977
5	6	0	−7.41	0.976
5	4	0	−7.01	0.940
6	5	0	−5.56	0.966
4	5	0	−4.86	0.959
4	6	0	−3.49	0.950

Identification Results

The model parameter estimation is performed with the SRIVC algorithm on the third-period data set. The identification result is given as the [6 6 0] Laplace transfer function model

$$\hat{G}(s) = \frac{20.87(s-618.5)(s^2-1.698s+710.6)(s^2+8.435s+2.012e4)}{(s^2+1.033s+2094)(s^2+0.9808s+9905)(s^2+2.693s+7.042e4)} \quad (3)$$

This estimated model is characterised by three, lightly damped dynamic modes, as defined in Table II.

Model Validation

Figure 9 compares the simulated SRIVC model output with the measured output series, over a short section of 0.4 s in the 8th-period data set. It can be noticed that the simulated output matches the measured data quite well, with $R_T^2 > 0.95$. There is also a very good agreement between the ETFE and the frequency response of the estimated SRIVC model, as shown in Figure 10.

TABLE II

Eigenvalues and dynamic modes for the robot arm SRIVC model

Real	Imag.	Damping	Nat. Freq. (rad/s)
–0.52	+ 45.76	0.0113	45.76
–0.52	− 45.76	0.0113	45.76
–0.49	+ 99.52	0.0049	99.52
–0.49	− 99.52	0.0049	99.52
–1.35	+ 265.37	0.0051	265.37
–1.35	− 265.37	0.0051	265.37

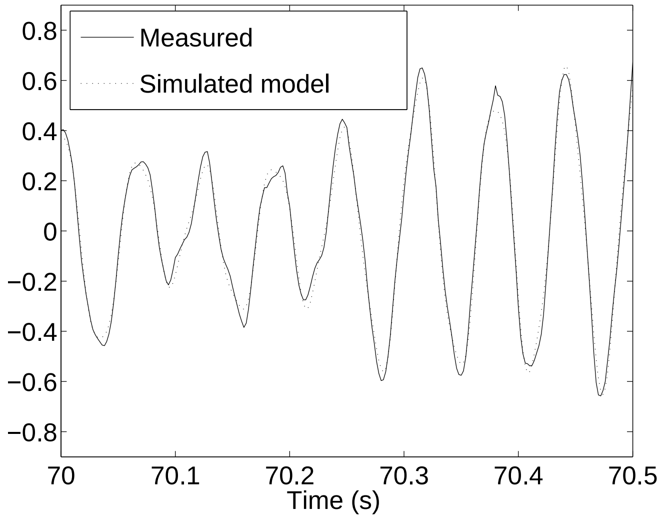


Fig. 9. Cross-validation results on a short section of the 8th-period data set

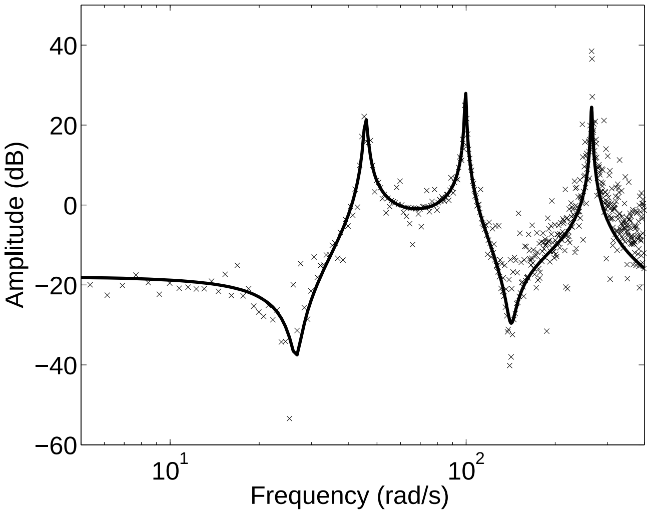


Fig. 10. Comparison of ETFE ('×') and SRIVC model (solid line) frequency responses for the flexible robot arm

VI. Conclusion

This paper has outlined the main features of the data-based continuous-time modelling course that is taught for engineering students at Nancy-University. In particular the Graphical User Interface (GUI) of the CONTSID toolbox which is so useful to perform the general identification procedure has been described and illustrated. Future developments for the course would be to introduce nonlinear system identification aspects to the students.

Références

[1] H. Garnier and L. Wang (Eds.), Ed., *Identification of continuous-time models from sampled data.* London: Springer-Verlag, 2008.

[2] L. Ljung, "Initialisation aspects for subspace and output-error identification methods," in *European Control Conference (ECC'2003)*, Cambridge (U.K.), September 2003.

[3] G. Rao and H. Garnier, "Identification of continuous-time systems: direct or indirect ?" *Systems Science*, vol. 30, no. 3, pp. 25–50, 2004.

[4] H. Garnier, M. Gilson, and O. Cervellin, "Latest developments for the Matlab CONTSID toolbox," in *14th IFAC Symposium on System Identification (SYSID'2006)*, Newcastle (Australia), March 2006, pp. 714–719.

[5] L. Ljung, *System identification. Theory for the user*, 2nd ed. Upper Saddle River: Prentice Hall, 1999.

[6] H. Garnier, M. Gilson, and V. Laurain, "The CONTSID toolbox for Matlab: extensions and latest developments," in *15th IFAC Symposium on System Identification (SYSID'2009)*, Saint-Malo (France), July 2009.

[7] H. Garnier, M. Mensler, and A. Richard, "Continuous-time model identification from sampled data. Implementation issues and performance evaluation," *International Journal of Control*, vol. 76, no. 13, pp. 1337–1357, 2003.

[8] I. Kollar, *Frequency Domain System Identification Toolbox Users's Guide.* Mass.: The Mathworks, Inc., 1994.