

A new Graphical User Interface for the CONTSID toolbox for Matlab

H. Garnier*, M. Gilson*, H. Muller*, F. Chen**

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

** *Wuhan University, Wuhan 430072, China*
fengwei.chen@whu.edu.cn.

Abstract: The main purpose of this contribution is to describe the new features of the latest version 7.4 of the CONTinuous-Time System IDENTification (CONTSID) toolbox for Matlab. The main addition is a new Graphical User Interface (GUI), which allows the user in a friendly and easy way to perform data analysis, model parameter estimation as well as model validation. The recent additions for MISO time-delay transfer function model identification are also briefly introduced.

Keywords: black-box model, continuous-time model, data-driven modelling, graphical user interface, MATLAB toolbox, software tools, system identification.

1. INTRODUCTION

The CONTSID toolbox to be run with MATLAB is entirely dedicated to continuous-time (CT) model identification from sampled data (Garnier and Wang, 2008). It was first released in 1999 (Garnier and Mensler, 1999) and it has been updated on a regular basis since its first release. The latest developments were presented at the last edition of the SYSID Symposium in Stockholm (Garnier and Gilson, 2018). 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 therefore the same data and model objects as well as a similar syntax for the function calls.

The toolbox provides standard parametric estimation techniques such as subspace and prediction-error minimization. However, in the CONTSID toolbox, there is a special focus towards instrumental variable (IV)-based estimation methods (Garnier, 2015). The toolbox can be freely downloaded from:
www.cran.univ-lorraine.fr/contsid/

The identification process was preferably executed in the command-line mode. To facilitate the use of the CONTSID toolbox routines, many tutorial programs have been developed that illustrate typical identification sessions. These tutorial programs were presented at the 2018 edition of the SYSID (Garnier and Gilson, 2018).

Tutorials guide beginners on their first steps on the software code use and are part of a learning process (Moeck et al., 2015). On the other hand, many data-driven modelling software packages such as for example the System Identification toolbox (Ljung, 1997) are equipped with graphical user interfaces (GUIs). GUIs provide a user-friendly environment that helps the user to focus on the aspects of data-driven modelling rather than on the

technicality of the underlying code. The absence of a GUI leaves the user to resolve all the technical challenges related to running the software code. Given the user-friendliness of GUIs, software without a GUI are in a strategic disadvantage, independent of their capabilities or level of sophistication. In fact, the absence of a GUI might even prevent the wide-spread application of a software or toolbox. Compared to tutorials, a GUI is however much more time-consuming to develop and cannot be rapidly adapted to changes and extensions. We believe that both tutorials and GUI are therefore complimentary and very helpful to make any complex software accessible to a large number of users. Both are also complimentary in helping a beginner to learn the complex process of practical data-driven continuous-time model identification.

The previous versions of the CONTSID toolbox included a very basic version of a GUI that needed to be modernized (see *e.g.* Garnier et al. (2008)). The need for a new GUI added to the advantages of GUIs listed above are the main reasons that motivated the developments of a new graphical user interface for the CONTSID toolbox. A last, but not least, motivation was to develop an interactive learning tool which could be used for introductory data-driven continuous-time model identification courses at universities and for engineers in industry. Interactive tools such as GUI are indeed very helpful in education. Although the CONTSID toolbox includes advanced estimation methods for nonlinear systems, the new GUI focuses on low-order process and transfer function models that are useful for linear control purposes.

This paper is organized as follows. Section 2 outlines the key features of the CONTSID toolbox and introduces briefly the recent developments for MISO time-delay transfer function model identification. The new GUI for the CONTSID toolbox is then presented in Section 3. Finally, concluding comments are given in Section 4.

2. KEY FEATURES AND LATEST DEVELOPMENTS FOR THE CONTSID TOOLBOX

2.1 The key features of the CONTSID toolbox

The key features of the latest version of the CONTSID toolbox for Matlab are summarized below:

- It supports methods for direct continuous-time model identification of dynamical systems from discrete-time data
- Most parameter estimation algorithms exploit the iterative instrumental variable SRIVC/RIVC method (see *e.g.* Garnier (2015)). Table 1 summarizes the main CONTSID toolbox commands for standard linear model SRIVC/RIVC-based identification.

Table 1. Main CONTSID toolbox commands for standard linear model SRIVC/RIVC-based identification

Model type	Estimation commands
Transfer function models	<code>tfstrivc</code>
Process models	<code>procsrivc</code>
Input/output polynomial models	<code>srivc</code> (COE models) <code>rivc</code> (CBJ models)

- It is easily handled by Matlab users as it has been built as an add-on to the system identification toolbox. Its routines use the same model/data objects and function calls, as illustrated below

```
>> model=tfstrivc(data,np,nz,iodelay);
>> present(model);
>> compare(data,model)
```

- The latest version 7.4 is compatible with Matlab 2021a and is freely available for *academic researchers* at: www.cran.univ-lorraine.fr/contsid

- It includes many tutorials that illustrate typical identification sessions. Type

```
>> contsid_demo
```

in the Matlab Command window and the CONTSID demonstration program window will open as shown in Figure 1.

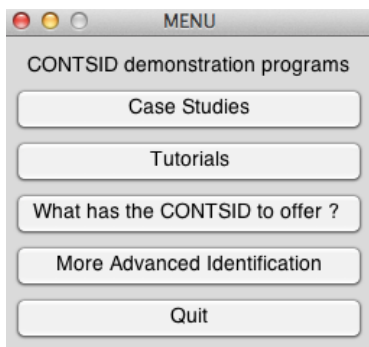


Fig. 1. Main menu window of the CONTSID demonstration programs

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

- Identification of low-order process models;

- Identification of transfer function plus time-delay models;
- Identification of input/output black-box polynomial models;
- Identification of state-space models with free or canonical parametrizations;
- Identification from time-domain 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;
- Identification of partial differential equation (PDE) models;
- On-line identification for tracking time-varying parameter (TVP) models.

The main advantages of the CONTSID methods have been recently discussed (Garnier and Young, 2014) and can be summarized as follows:

- they can cope easily with fast sampled data (fast sampling refers here to the usual rule of thumb given for discrete-time model identification (Ljung, 1999));
- they can handle irregularly sampled data;
- they include inherent data filtering;
- they are robust against measurement noise;
- they require less participation from the user since most routines are free of user-parameter choice, making the application of the system identification procedure much simpler and easier.

2.2 Latest developments for the CONTSID toolbox

The main recent additions for the new release of the CONTSID toolbox include:

- a new GUI that should provide the beginner a user-friendly environment that helps him to focus on the aspects of data-driven modelling rather than on the technicality of the underlying code. This new GUI is described in more detail in the next section.
- a few new routines for estimating transfer function plus time-delay models for MISO 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. On the other hand, as discussed in Pasco et al. (2019), further research and development are needed for multi-input linear model identification. Recent developments have led to extend the SISO version (TFSRIVC/TFRIVC) to estimate transfer function plus time-delay models to the MISO system case (Chen et al., 2020). The developed methods are based on the principle of variable projection, combining a SRIVC/RIVC methods for the rational model parameters and an adaptive search for the time-delay. The updated TFSRIVC/TFRIVC methods have shown to

be effective in terms of both numerical simulations and practical applications (Chen et al., 2020). A new tutorial has been added to illustrate the recent developments for MISO transfer function plus time-delay model identification with the CONTSID toolbox. It can be executed by typing in the command window: `>>contsid_TFRIVC_MISO`

3. THE NEW GUI FOR THE CONTSID TOOLBOX

The major addition for the new release of the CONTSID toolbox is a new GUI. While the CONTSID toolbox supports basic and more advanced identification methods for linear and nonlinear systems, the new version of the GUI considers the identification of continuous-time simple process or transfer function (plus time-delay) models for linear SISO or MISO systems only. In most practical cases, control problems are solved using PI or PID controllers being tuned using simple process models. A simple form of system identification is often made by the process engineers. Typically, they estimate gain, dead time, and time constants from open loop step responses, or relay feedback tests. The controller parameters are then obtained using simple tuning rules like the popular Ziegler-Nichols rules (see *e.g.* Aström and Hägglund (2005)). Another reason to restrict the GUI to low-order transfer function model identification is linked to the idea of developing interactive learning tools that can be used for introductory courses on data-driven modelling for control at universities and for engineers in industry, as discussed in Guzman et al. (2008). The GUI should therefore be suitable for self-study but it also should ease demonstrations in lectures. For all these reasons, the new GUI limits the estimation process to simple process and transfer function plus delay models of MISO linear systems that are most useful for control purposes.

The traditional model identification workflow to determine a continuous-time model of a dynamical system directly from observed input-output data consists in repeatedly selecting a model structure, computing the best model in the chosen structure, and evaluating the identified model. The iterative procedure can thus be summarized by the following four main stages:

- (1) Design an experiment and collect input-output data from the process to be identified.
- (2) Manage and examine the data. Remove trends and outliers, and select useful portions of the original data.
- (3) Select a model structure within a set of candidate system descriptions. Learn the best parameters in the chosen model structure according to the input-output data and a given criterion of fit.
- (4) Validate the estimated model by examining its 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 or work further on the input-output data (steps 1 and 2).

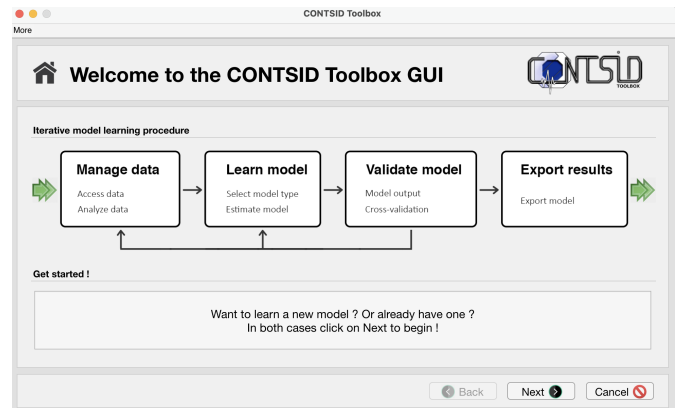


Fig. 2. The Welcome window of the new CONTSID toolbox GUI

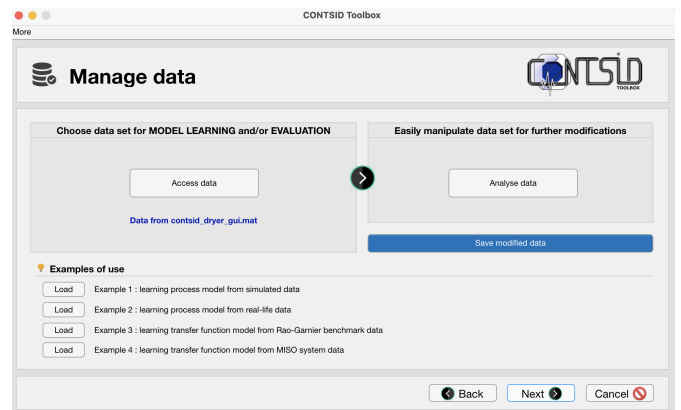


Fig. 3. The Manage data window of the new CONTSID GUI

3.1 Starting the new GUI

The new graphical user interface for the CONTSID toolbox can be started by typing the following command: `>> contsid_gui`

As shown in Figure 2, the new GUI provides a Welcome window, where stages 2 to 4 of the iterative identification procedure are presented in a horizontal way:

- a first **Manage data** block as shown in Figure 3, where data sets can be imported, plotted, pre-treated and selected;
- a second **Learn model** block as shown in Figure 4, where different model structures can be selected and estimated;
- a third **Validate model** block, as shown in Figure 5, where basic properties of the identified model can be examined;
- a fourth and final **Export results** block where the identified model can be easily exported in a *.mat* file to be saved in a chosen folder.

Moving forward and backward between the different windows can be done by clicking on the *Back* and *Next* buttons available at the bottom right corner of every window.

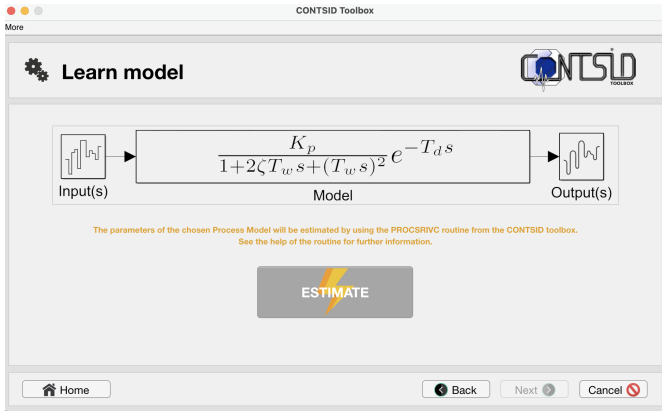


Fig. 4. The Learn model window of the new CONTSID GUI

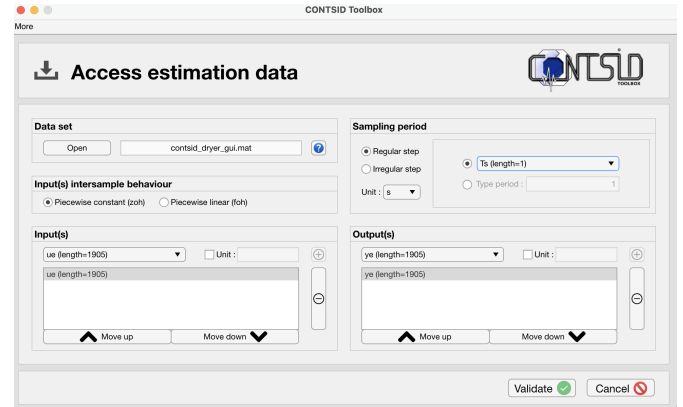


Fig. 6. The Access data window of the new CONTSID GUI

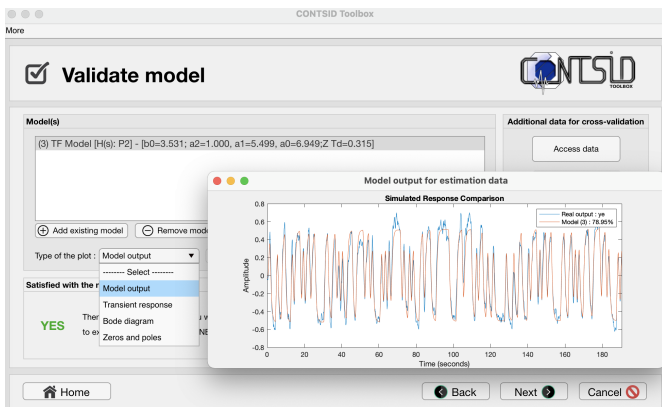


Fig. 5. Validate model window of the new CONTSID GUI

3.2 The Manage data window

When you select the *Access data* button as displayed in Figure 3, you can then load different data sets for model learning and/or for validation purpose. As it can be noted from the bottom part of Figure 3, four examples are available to illustrate the use of the new GUI. The beginner has simply to select one of the tutorials and follow along by clicking *Next* at every new window.

Access data window. By clicking on the *Open* button, you can import time-domain sampled data from a *.mat* file for systems with multiple input and output channels, as illustrated in Figure 6. 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 piecewise linear (foh)). Note that help about the use of the window can be obtained by selecting the blue question mark button in the middle of the window which results in the information displayed in Figure 7.

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 *Analyse data* button in the right part of the Access data window as shown in Figure 3, is displayed in Figure 8. The loaded input/output data are easily plotted. This window also allows the

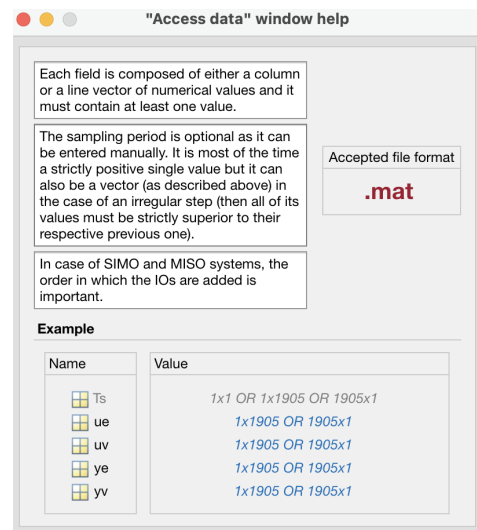


Fig. 7. Help information about the Access data window

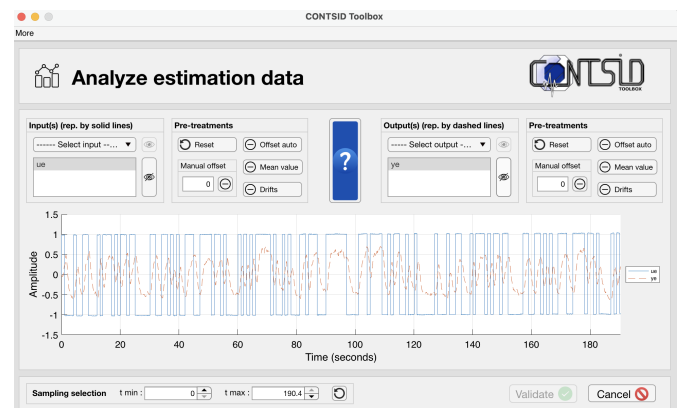


Fig. 8. Data plot and pretreatment window

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 can happen mainly for two reasons:

- these data include erroneous values or outliers 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 *Sampling selection* box at the bottom of Figure 8 allows the user to easily define the portion of the measured data that he wants to select.

3.3 Select model type window

While the CONTSID toolbox supports advanced model identification methods, the GUI lets you estimate two types of models only, using the following two SRIVC-based learning algorithms (see for more detail Young et al. (2008); Garnier (2015); Chen et al. (2017, 2018)):

- PROCSRIVC to estimate low-order plus delay process models. Process models are popular for describing system dynamics in many industries. The advantages of these models are that they are simple, support transport delay estimation, enforced integration is possible and the model parameters have an easy interpretation as poles and zeros. Examples of this type of model structures include first order models of the form:

$$G(s) = \frac{K_p}{1 + T_{p1}s} e^{-T_d s} \quad (1)$$

where s denotes the Laplace variable but also second-order models with two real poles with or without zero of the form

$$G(s) = \frac{K_p}{(1 + T_{p1}s)(1 + T_{p2}s)} e^{-T_d s} \quad (2)$$

$$G(s) = \frac{K_p(1 + T_z s)}{(1 + T_{p1}s)(1 + T_{p2}s)} e^{-T_d s} \quad (3)$$

or cascaded first and second order models of the form

$$G(s) = \frac{K_p}{(1 + 2\zeta T_w s + (T_w s)^2)(1 + T_{p3}s)} e^{-T_d s} \quad (4)$$

where K_p represents the steady-state gain, T_{p1}, T_{p2}, T_{p3} are time constants, ζ is the damping factor, T_w is the inverse of the undamped natural frequency, T_z determines the process zero, and T_d is the time delay.

- TFSRIVC to estimate transfer function plus time-delay models of the form

$$G(s) = \frac{B(s)}{A(s)} e^{-T_d s} \quad (5)$$

where $B(s)$ and $A(s)$ are polynomials of arbitrary degrees.

The user is thus invited to choose the type of model to be estimated amongst the options available at the top of the *Select model type* window, as shown in Figure 9. After selecting the model type, the user has to specify the polynomial degrees and the possible time-delay of the model to be estimated.

Then, if the model order is not known *a priori*, the *Best structure search* button allows the user to automatically search over a selected range of possible model orders, as illustrated in Figure 10. For each of these estimated models, two traditional criteria YIC and R_T^2 (Garnier et al., 2008) are computed from the estimation data set. From the results displayed under the form of a figure (not shown here due to space limit), the user can select the

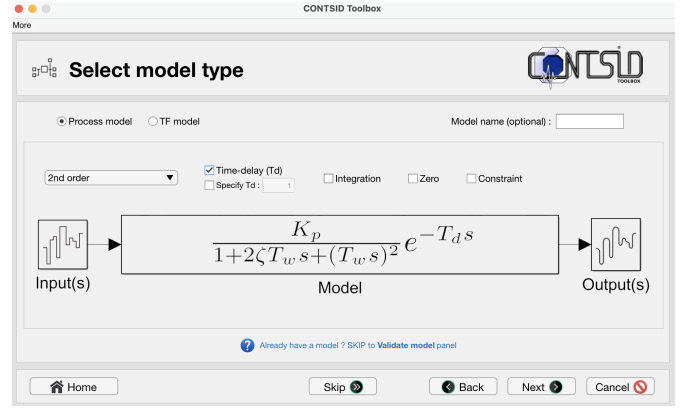


Fig. 9. Select model type window

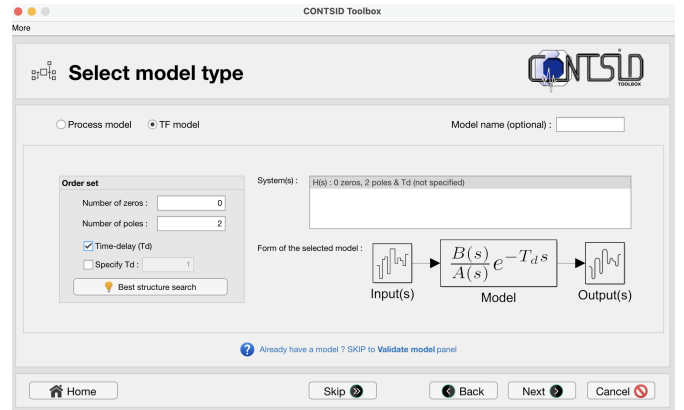


Fig. 10. Best structure search window

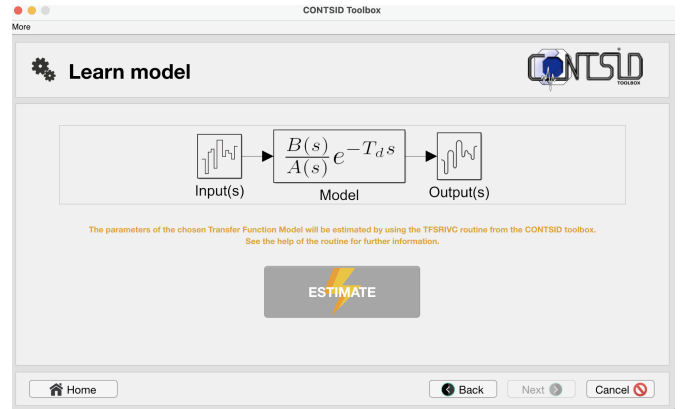


Fig. 11. Learn window

best model orders according to several available criteria and then set the order of the final model to be estimated by clicking on the *Return to the structure set* button.

3.4 Learn model window

Once we have set the structure of the model to be identified, the parameters can then be estimated/learned by clicking on the *Learn* button, as shown in Figures 4 and 11 depending on the selected process model or transfer function model form.

3.5 Validate model window

Once a model is estimated, it will appear as highlighted in light blue in the box located at the top panel of the *Validate model* window, as shown in Figure 5. Several basic model properties can then be evaluated from an unrolling menu, as illustrated in Figure 5, by using first the data that was used for model learning:

- *Model output*: plots and compares the simulated model output with the measured output. This indicates how well the system dynamics are captured;
- *Transient response*: displays the model response to an impulse or step excitation signal;
- *Bode diagram*: displays the 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 over-parameterized modelling.

If a second data set is available, then traditional cross-validation tests can be done which consists in comparing the measured and simulated model outputs on a fresh data set that was not used to learn the parameters. If the model is good enough, then proceed to the final *Export results* window; otherwise go back to the different stages of the learning model workflow by using the *back* or *home* button.

3.6 Export results window

Once the user is satisfied with the quality of his model, the selected model can be easily exported in a .mat file by clicking on the *Export model* button.

4. CONCLUSION

The new key features of the CONTSID toolbox for Matlab have been presented. In particular the new Graphical User Interface (GUI) has been described and discussed. In its current version, the CONTSID GUI supports the direct identification of SISO and MISO continuous-time simple process and transfer function models from regularly and irregularly sampled data. These methods have proven successful in many practical applications. The new GUI of the CONTSID toolbox provides a user-friendly environment that will help the user to focus on the result analysis of data-driven modelling rather than on the technicality of the toolbox commands. The educational/pedagogical aspects of the GUI are most essential. The new GUI has been used in different data-driven model identification courses at University of Lorraine. First feedback from the students has been extremely positive. It is therefore hoped that the new easy-to-use GUI for the CONTSID toolbox will guide the inexperienced user or learning beginners through the complex process of practical data-driven continuous-time model identification.

REFERENCES

Aström, K. and Hägglund, T. (2005). *Advanced PID Control*. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709.

Chen, F., Gilson, M., Garnier, H., and Liu, T. (2017). Robust time-domain output error method for identifying continuous-time systems with time delay. *Systems and Control Letters*, 102, 81 – 92.

Chen, F., Young, P., Garnier, H., Deng, Q., and Kazimierczuk, M. (2020). Data-driven modeling of wireless power transfer systems with multiple transmitters. *IEEE Transactions on Power Electronics*, 35(11), 11363–11379.

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 Gilson, M. (2018). CONTSID: a matlab toolbox for standard and advanced identification of black-box continuous-time models. In *18th IFAC Symposium on System Identification (SYSID'2018)*. Stockholm (Sweden).

Garnier, H., Gilson, M., Bastogne, T., and Mensler, M. (2008). CONTSID toolbox: a software support for continuous-time data-based modelling, in *Identification of continuous-time models from sampled data*. Springer, London, H. Garnier and L. Wang (Eds.), 249–290.

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, 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.

Guzman, J., Aström, K., Dormido, S., Hägglund, T., Pignet, Y., and Berenguel, M. (2008). Interactive learning module: Basic modelling and identification concepts. In *17th Triennial IFAC World Congress on Automatic Control*. Seoul (South Korea).

Ljung, L. (1997). Developments for the system identification toolbox for Matlab. In *11th IFAC Symposium on System Identification*, 969–973. Kitakyushu (Japan).

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).

Moeck, C., Hunkeler, D., and Brunner, P. (2015). Tutorials as a flexible alternative to GUIs: An example for advanced model calibration using pilot points. *Environmental Modelling & Software*, 66, 78–86.

Pascu, V., Garnier, H., Ljung, L., and Janot, A. (2019). Benchmark problems for continuous-time model identification: design aspects, results and perspectives. *Automatica*, 107, 511–517.

Young, P.C., Garnier, H., and Gilson, M. (2008). Refined instrumental variable identification of continuous-time hybrid Box-Jenkins models, in *Identification of continuous-time models from sampled data*. Springer, London, H. Garnier and L. Wang (Eds.), 91–132.