

Version 7.0 of the CONTSID toolbox

A. Padilla^{1,2}, H. Garnier^{1,2}, M. Gilson^{1,2},

¹*Université de Lorraine, CRAN, UMR 7039, 2 rue Jean Lamour,
54519 Vandoeuvre-les-Nancy, France*

²*CNRS, CRAN, UMR 7039, France*

Abstract: The CONtinuous-Time System IDentification (CONTSID) toolbox to be run with MATLAB includes estimation routines to determine and evaluate continuous-time models of dynamic systems directly from discrete-time input-output data. Version 7.0 of the CONTSID toolbox is now fully compatible with the latest version of the System Identification toolbox for MATLAB. The new release introduces important new options for developing models. It includes additional routines for identifying continuous-time linear models from frequency-domain data, as well as linear systems with arbitrary time delay. The toolbox now supports estimation routines for the determination of simple process models from both regularly and irregularly sampled data and additional algorithms for identifying partial differential equation models. This paper presents and illustrates these new features.

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

1. INTRODUCTION

The field of system identification is an extensive and versatile area. It is important to package the identification tools in a user-friendly way. Fortunately, there exists a range of actively maintained toolboxes for dynamic system identification. These include commercially licensed ones such as the comprehensive Mathworks System Identification (SID) toolbox [Ljung and Singh, 2012] and the Gamax Frequency Domain System Identification Toolbox [Kollar et al., 2006]. Non-commercial and freely available packages include the UNIT toolbox which provides several new approaches like the Expectation Maximisation (EM) algorithm for computation of Maximum Likelihood estimates, Bayesian methods or particle filtering techniques for nonlinear system estimation [Ninness et al., 2013], the CAPTAIN toolbox [Young and Taylor, 2012] which is a platform supporting the Data-Based Mechanistic (DBM) approach to the modelling of stochastic, dynamic systems from sampled data and includes the Refined Instrumental Variable (RIV) algorithms and the ITSIE toolbox [Guzman et al., 2012] which has an emphasis on education and training in system identification principles.

Some of the toolboxes above include a few routines for identifying continuous-time (CT) models. However, there was no software entirely dedicated to the direct CT schemes to system identification. An attempt to fill this gap was carried out with the CONtinuous-time System IDentification (CONTSID) toolbox for Matlab. The other motivations for developing the software suite were to provide a platform for evaluating the performance of the direct CT methods relative to existing ones, and also a way for effectively disseminating these lesser-known methods, at the time, to other researchers. The CONTSID toolbox to be run with MATLAB was released in its first version in 1999 [Garnier and Mensler, 1999]. Several releases have been reported since mainly at IFAC SYSID Symposia (see

e.g. [Garnier et al., 2012] and references herein). The key features of the CONTSID toolbox are the following (see Chapter 9 of [Garnier and Wang, 2008]):

- it supports most of the time-domain methods for identifying CT parametric models of linear and nonlinear dynamic systems directly from uniformly or non-uniformly sampled input/output data;
- it provides a wide range of linear model structures (CARX, COE, Hybrid BJ), estimation methods and algorithms for SISO and MIMO systems, including optimal IV-based SRIVC/RIVC routines as well as PEM and subspace-based techniques mainly in open loop, but it also contains a few routines for closed-loop model identification;
- it may be seen as an add-on to the MATLAB System Identification (SID) toolbox.
- the procedures can be executed in the command-line. There is also a graphical user interface (GUI) that lets the user analyse the experimental data, identify and evaluate linear models in an easy way;
- it includes many demos to illustrate its use;
- it is freely available for academic researchers and can be downloaded from www.cran.univ-lorraine.fr/contsid/

Version 7.0 of the CONTSID toolbox is now fully compatible with the latest version of the System Identification toolbox for MATLAB. This release includes also important new options for developing models.

The paper is organized in the following way. An overview of the main aspects of the toolbox is given in Section 2. The new developments are then described and illustrated with the help of simulation examples.

2. OVERVIEW OF THE CONTSID TOOLBOX

A broad spectrum of system identification techniques has been developed over the past decades. These include the main stream approaches of prediction error identification, subspace identification or instrumental variable (IV) identification. The CONTSID toolbox includes estimation methods of the three types above but one approach has proven to be very easy to use, and generally gives very good results for day-to-day usage (see e.g. [Garnier and Young, 2014]). This method is known as the Simple Refined Instrumental Variable approach for Continuous-time models (SRIVC). It provides an iterative implementation of the optimal instrumental variable solution for linear Continuous-time Output Error (COE) models.

Some developments were done in the last decade to extend this SRIVC method to handle wider practical situations. SRIVC-based methods have thus been developed and implemented in the CONTSID toolbox for Box-Jenkins model identification, closed-loop model identification, block-oriented Hammerstein and Hammerstein-Wiener model identification, as well as LPV model identification. A model order selection algorithm associated to the SRIVC method has also been developed to help the user to automatically search over a whole range of different model orders. The SRIVC-based routines available in the CONTSID toolbox are summarized in Table 1.

The recent developments reported in this paper have also aimed at extending the SRIVC method to handle further practical situations. The latest version of the CONTSID toolbox has the following additions:

- it now supports linear transfer function model identification from frequency-domain data [Gilson et al., 2013];
- it has a new SRIVC-based function to identify time-delay models from irregularly sampled data [Chen et al., 2015];
- it now provides a SRIVC-based routine to estimate low-order (simple) process models;
- it includes a new SRIVC-based method for identifying partial differential equation models [Schorsch et al., 2013];
- the demonstration programs to illustrate the command-line use have been entirely rewritten.

3. CONTSID NOW COMPATIBLE WITH THE LATEST VERSION OF THE SID TOOLBOX

The CONTSID toolbox can be seen as an add-on to the MATLAB SID toolbox. Indeed, to facilitate its use, it has been given a similar setup and exploits the same syntax, data and model objects than the SID toolbox. As such, it is dependent of possible major updates of the mother toolbox. Version 8 of the system identification was released with version R2012a of MATLAB in Spring 2012. The 2012 release was the most reworked and re-engineered version and included interesting new features and model objects [Ljung and Singh, 2012]. Although backward compatibility was tried to be maintained, there were some issues with the new or modified model objects/syntaxes that resulted in the version 6.0 of the CONTSID toolbox not to run anymore with the latest version of MATLAB. Since 2012, we

have received frequent requests to update the CONTSID version so that it becomes compatible with the latest version of the SID toolbox. All CONTSID parametric model estimation functions share the same command structure

```
M = function(data,modstruc);
```

The function name is defined by the abbreviation for the estimation method, as for example, `srivc` for the Simple Refined Instrumental Variable approach for Continuous-models. The input argument `data` is an `iddata` object that contains the output and input data sequences along with the sampling time and intersample behaviour for the input, while `modstruc` specifies the particular structure of the model to be estimated. The resulting estimated model is contained in `M`, which is a model object that stores the various usual information after a model has been estimated.

These information were previously stored in the “EstimationInfo” property. The latter was replaced by a new “Report” property in version 8 of the SID toolbox. The CONTSID routines have been all modified to make use of this new “Report” property.

To illustrate this new feature, a part of one of the demonstration program is presented below. This demo considers the Rao-Garnier test system (see e.g. Garnier and Young [2014]):

$$\begin{cases} x(t) = \frac{-6400s + 1600}{s^4 + 5s^3 + 408s^2 + 416s + 1600}u(t), \\ y(t_k) = x(t_k) + e(t_k). \end{cases} \quad (1)$$

where $e(t_k)$ is a zero-mean discrete-time (DT) white Gaussian noise sequence and s is the differential operator. For data collection, the system is sampled with a sampling time of $T_s = 10$ ms and simulated with a pseudo random binary signal of length $N = 7016$. The *Signal-to-Noise Ratio* (SNR) is equal to 10 dB.

```
B = [-6400 1600];
F = [1 5 408 416 1600];
```

Table 1. SRIVC-based estimation routines available in the CONTSID toolbox

Program	Description
<code>srivc</code>	SRIVC method for COE model identification in open loop
<code>rsrivc</code>	Recursive SRIVC method for COE model identification in open loop
<code>rivc</code>	RIVC method for hybrid Box-Jenkins model identification in open loop
<code>clsrivc</code>	SRIVC-based method for closed loop identification
<code>cl2srivc</code>	Two-stage SRIVC-based method for closed loop identification
<code>hsrivc</code>	SRIVC-based method for Hammerstein model identification
<code>hwsrivc</code>	SRIVC-based method for Hammerstein-Wiener model identification
<code>lpvsrivc</code>	SRIVC-based method for LPV model identification
<code>srivcstruc</code>	SRIVC-based procedure to search for the best linear model structure

```

MO = idpoly(1,B,1,1,F,'Ts',0);
u = prbs(10,7);
Ts = 0.01;
datau = iddata([],u,Ts,'InterSample','zoh');
snrdb = 10;
y = simc(MO,datau,snrdb);

```

We first create an `iddata` object with noisy output `y`, input `u` and sampling interval `Ts`

```
data = iddata(y,u,Ts);
```

We then identify a COE model for this system from the `iddata` object `data` with the iterative IV (`srivc`) method. The extra pieces of information required are the number of numerator and denominator parameters and number of samples for the delay of the model $[n_b \ n_f \ n_k] = [2 \ 4 \ 0]$.

```

Msrivc = srivc(data,[2 4 0])
Continuous-time OE model:
y(t) = [B(s)/F(s)]u(t)
  B(s) = -6468 s + 1552
  F(s) = s^4 + 5.083 s^3 + 409.2 s^2 + 421.3 s
        + 1606

```

```

Fit to estimation data: 71.89
FPE: 1.186, MSE 1.172
Status:
Estimated using Contsid SRIVC method on time
domain data.

```

For any estimation routine, the new `Report` property contains information on the various initialization choices made, nature of estimation data, the value of model parameters and their covariance at the time of estimation. `Report` also contains information on termination criteria for iterative search-based estimations and a record of achieved fit to the data measured by a normalized root mean square (NRMSE) fit value and the final prediction error norm (FPE). For the estimated `Msrivc` model, the report offers the following information:

```

Msrivc.Report
Status: 'Estimated using Contsid SRIVC method
on time domain data'
  Method: 'SRIVC'
  InitialCondition: 'Auto'
    Fit: [1x1 struct]
    Parameters: [1x1 struct]
    OptionsUsed: [1x1 struct]
    RandState: [1x1 struct]
    DataUsed: [1x1 struct]
    Termination: [1x1 struct]

```

The measured and simulated model outputs together with the 95% confidence bounds are plotted in Fig. 1.

4. SRIVC-BASED IDENTIFICATION FROM FREQUENCY-DOMAIN DATA

4.1 Overview

Identification of continuous-time systems can also be performed in the frequency domain. The frequency domain data could be either input/output Fourier transforms or a measured frequency response data that is available or estimated in a particular frequency grid denoted by the frequencies ω_k , $k = 1, \dots, N$. The identification problem is

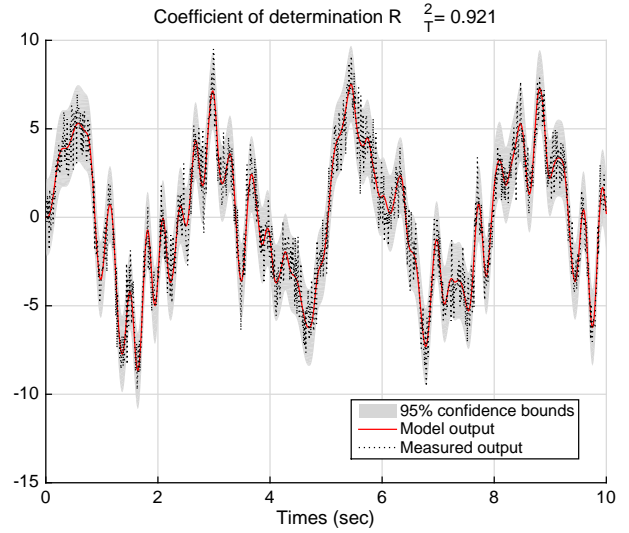


Fig. 1. Simulated model output with the measured output for the Rao-Garnier benchmark.

then to fit a parametric model to the frequency domain data. The SRIVC method has been naturally extended to be applicable in the frequency domain (see *e.g.* [Gilson et al., 2013]).

4.2 Example

We illustrate here how the `CONTSID srivc` routine can now identify a linear model when Fourier transform values of the input and output can be computed from time-domain data.

Consider the Rao-Garnier system defined in (1). We choose a multisine input with low frequency contents that is fast sampled:

```

u = idinput(500,'sine',[0 0.5]);
Ts=0.05;
u = iddata([],u,Ts,'intersamp','bl');

```

`Ts=0.05` is the sampling interval, and `'bl'` indicates that the input is band-limited, *i.e.* in continuous time it consists of sum of sinusoids with frequencies below half the sampling frequency. Correct simulation of such a system should be done in the frequency domain:

```

U = fft(u);
U.Ts = 0;
Y = sim(MO,U);
N=length(Y.N);

```

Add some white Gaussian noise to the data:

```

rng(235,'twister');
Y.y = Y.y + 0.1*(randn(N,1)+1i*randn(N,1));
datafreq = [Y U];

```

```

M = srivc(datafreq,[2 4])
M =

```

```

Continuous-time OE model:
y(t) = [B(s)/F(s)]u(t) + e(t)
  B(s) = -6268 s + 1560
  F(s) = s^4 + 5.011 s^3 + 409 s^2 + 426.4 s
        + 1666

```

```
Status:
```

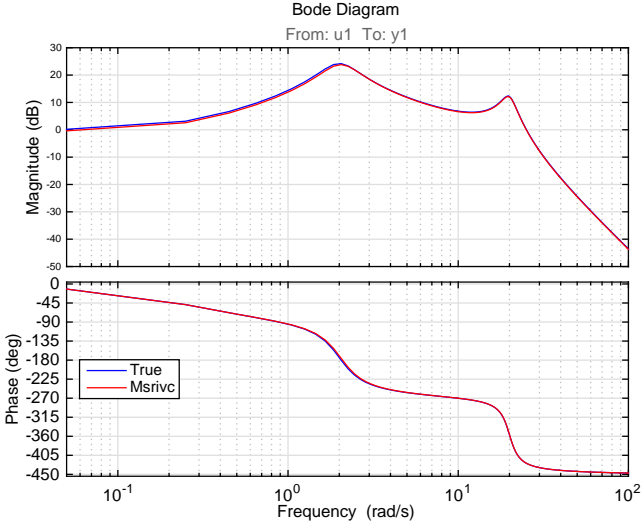


Fig. 2. Bode plot of the true system and the estimated model using `srivc` from frequency-domain data.

Estimated using Contsid SRIVC method
on frequency domain data.
Fit to estimation data: 82.81
FPE: 5.057e-02, MSE 4.821e-02

The very good agreement between the estimated model and the true system can be observed from the Bode plot displayed in Fig. 2.

5. SRIVC-BASED IDENTIFICATION FOR TIME-DELAY MODELS

5.1 Overview

An important additional part of the model structure is the existence of a pure time-delay parameter so that the model takes the general form:

$$\begin{cases} x(t) = \frac{B(s)}{F(s)}u(t - T_d), \\ y(t_k) = x(t_k) + e(t_k). \end{cases} \quad (2)$$

Unlike the situation in DT identification, where the time delay T_d is assumed to be an integral number of sampling intervals and is often absorbed into the definition of the numerator polynomial (as leading zero-valued parameters), the time-delay parameter for CT system models is normally associated directly with the input signal and can have a non-integral value. The SRIVC method has been recently coupled to a gradient-based search for the time-delay which makes it possible to simultaneously identify the continuous-time model parameters and time-delays. To guarantee the iterative search to converge to the global minimum, special care is required for the initial value of the time-delay estimate and a pre-filtering strategy has been recently suggested [Chen et al., 2015]. The approach has been developed to work with both regularly and irregularly sampled data.

The CONTSID toolbox provides now a new SRIVC-based function called `tdsrivc` that allows the user to identify pure time-delay models.

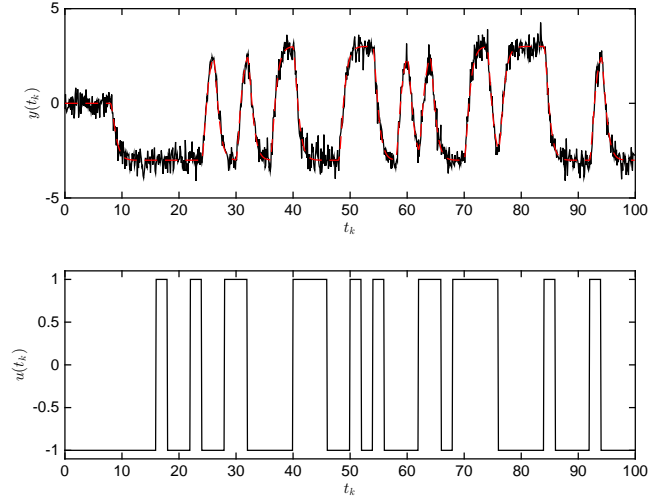


Fig. 3. Estimation data and simulated model output (red dashed line) using `tdsrivc`.

5.2 Example

The CONTSID function `tdsrivc` is easy to use. A short example will illustrate the typical steps involved in identifying a model using `tdsrivc` from regularly sampled data.

Consider the continuous-time system

$$\begin{cases} x(t) = \frac{12}{s^2 + 4s + 4}u(t - T_d), \\ y(t_k) = x(t_k) + e(t_k). \end{cases} \quad (3)$$

The input is a pseudo random binary sequence. The number of observations is $N = 1000$ and the sampling time is $T_s = 0.1$ s. A white noise with a SNR of 15 dB is added to the simulated output. Notice that the input arguments for the `tdsrivc` routine are just the data and the number of numerator and denominator parameters to be estimated $[\mathbf{n}_b \ \mathbf{n}_f] = [1 \ 2]$, *i.e.*,

```
M = tdsrivc(data, [1 2]);
```

The estimated model is an `idtf` MATLAB object,

```
M =
    11.03
    exp(-7.97*s) * -----
                    s^2 + 3.781 s + 3.67
Continuous-time identified transfer function.
Parameterization:
  Number of poles: 2   Number of zeros: 0
Status:
Estimated using Contsid TDSRIVC method on time
domain data
```

A plot of the input, output and simulated model output is shown in Fig. 3.

6. SRIVC-BASED IDENTIFICATION FOR SIMPLE PROCESS MODELS

6.1 Overview

In industrial practice, identification for control really is the determination of simple process models of order limited at most to 3, adjusted from transient or possibly relay

experiments, followed by the tuning of PI(D)-regulator based on the identified simple process model parameters. The family of simple process models is characterized by static gain, possible dead time and dominating time-constants. Examples of this type of model structures include first order models of the form:

$$G(s) = \frac{K_p}{1 + T_{p1}s} e^{-T_d s} \quad (4)$$

but also second-order models with two real poles of the form

$$G(s) = \frac{K_p(1 + T_z s)}{(1 + T_{p1}s)(1 + T_{p2}s)} e^{-T_d s} \quad (5)$$

or second order models with complex poles of the form

$$G(s) = \frac{K_p(1 + T_z s)}{1 + 2\zeta T_w s + (T_w s)^2} e^{-T_d s} \quad (6)$$

Fitting low-order process models is just a special case of the `tdsrivc` identification method, presented in the previous section. Thus, the CONTSID toolbox now includes a `procsrivc` routine, which internally calls `tdsrivc`. `procsrivc` builds low-order models of at most order 3 for control designs. It works for both regularly and irregularly sampled data.

6.2 Example

A short example will illustrate the typical steps involved in identifying a model using `procsrivc` from a step response.

Consider a second-order system of the form of (6), where $K_p = 1$, $T_z = 0.05$, $\zeta = 0.1$, $T_w = 0.05$ and $T_d = 0.7$. The input signal is a unit step and a white noise with a SNR of 15 dB is added to the noise-free output. The data is irregularly sampled depending on the changes on the corrupted output. The number of resulting observations is $N = 153$ and the sampling interval varies in the range $[0.01, 0.08]$ s (see bottom plot in Fig. 4). The routine is called using an initial guess for the time delay as follows

```
init_sys = idproc('P2DU');
init_sys.Structure.Td.Value = 0.4;
```

The model is estimated by:

```
M = procsrivc(data,init_sys)
M =
```

Process model with transfer function:

$$G(s) = \frac{K_p}{1 + 2\zeta T_w s + (T_w s)^2} * \exp(-T_d s)$$

$K_p = 0.99938$
 $T_w = 0.049551$
 $\zeta = 0.10223$
 $T_d = 0.70053$
 parameterization:
 'P2DU'

Status:

Estimated using Contsid PROCSRIVC method

Fit to estimation data: 72.02

FPE: 1.618e-02, MSE 1.248e+00

The irregularly sampled output and the simulated model output are compared in Fig. 4.

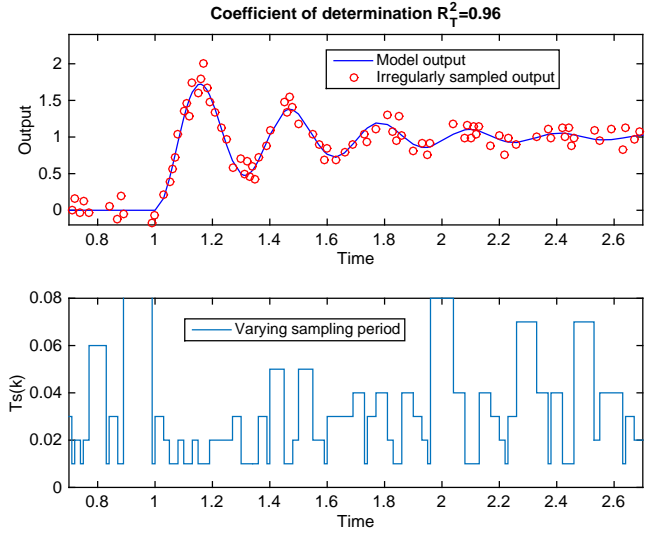


Fig. 4. Comparison between irregularly sampled output and model output using `procsrivc`.

7. SRIVC-BASED IDENTIFICATION FOR PDE MODELS

7.1 Overview

A large variety of industrial and environmental systems involve spatio-temporal behaviour. Typical examples are heat transmission and/or exchange in engineering structures and the transport/dispersion of contaminants in rivers and the atmosphere. These systems are intrinsically distributed parameter systems whose description usually requires the introduction of partial differential equations (PDE). The CONTSID toolbox now includes a few routines for identifying CT PDE models from spatio-temporal sampled data where the adaptively updated filters and auxiliary model used in the traditional SRIVC method are replaced by their two-dimensional, spatio-temporal equivalents [Schorsch et al., 2013].

7.2 Example

A short example will illustrate the typical steps involved in identifying and verifying a PDE model. The chosen example is concerned with heat diffusion along a bar of length $l = 1$ m, where it is assumed that the bar temperature $y(x, t)$ is the output and the heat source $u(x, t)$ is the input, with the bar initially set at zero uniform temperature throughout. The general conduction equation is driven by the following PDE

$$\begin{cases} \frac{\partial \hat{y}}{\partial t}(x, t) - a \frac{\partial^2 \hat{y}}{\partial x^2}(x, t) = bu(x, t) \\ 0 \leq t \leq 30\text{mn} \quad 0 \leq x \leq 1\text{m} \\ y(x_\ell, t_k) = \hat{y}(x_\ell, t_k) + e(x_\ell, t_k) \end{cases} \quad (7)$$

with zero initial and boundary conditions, and a parameter vector

$$\theta = [a \ b]^T = [8.418 \times 10^{-5} \ 1]^T$$

The rod is heated by a source signal represented by $u(t, x)$ in its middle which takes the form of step pulses with different time lengths. The total duration of the

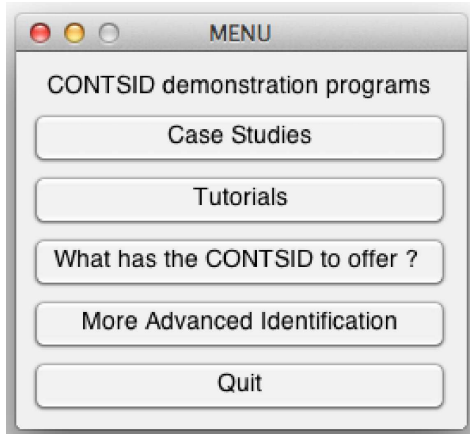


Fig. 5. Main window of the new demonstration program.

experiment is 30 minutes and the sampling period is here set to 0.5 sec. One hundred sensors are available and spatio-temporal response over the whole bar is obtained by discretization of the PDE. The output is corrupted by a zero-mean normally distributed 2-D discrete-time noise, with a SNR of 10 dB. We first create two data structures: `data1` containing the input and output data and `data2` containing the time and space measurements. We then identify a PDE model from the two `iddata` objects with the iterative SRIVC-based method dedicated to the considered heat diffusion modeling problem. The third input argument of the function `Filter` defines the SVF filter used to initiate the iterative SRIVC search.

```
theta=pdesrivc4heatdiffusion(data1,data2,Filter)
theta: 8.3671e-05  0.99434
Estimated using Contsid PDESRIVC4HEATDIFFUSION
method on time domain data
```

More details on the use of this new function can be obtained by executing the associated new demo.

8. NEW DEMONSTRATION PROGRAMS

Making demonstrations programs to illustrate the use of a toolbox is an important part of any software development project. The demonstration programs for the CONTSID toolbox have been entirely revisited. The new demos can still be run by entering: `idcdemo`. The user is then invited to select from a menu as shown in Fig. 5 between several options illustrating what might be typical sessions with the CONTSID toolbox.

Table 2. New SRIVC-based estimation routines available in the CONTSID toolbox

Program	Description
<code>srivc</code>	SRIVC method for COE model identification from time or frequency-domain data
<code>tdsrivc</code>	SRIVC method for COE plus input time-delay model identification
<code>procsrivc</code>	SRIVC-based method for low-order (simple) process model identification

9. 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 is now fully compatible with the latest version of the system identification toolbox and includes new techniques to handle wider and practical situations that are summarized in Table 2. The new version can handle frequency-domain data, low-order process model identification and time-delay system identification from irregularly sampled data. It also now supports a few routines dedicated to the more challenging identification problem of partial differential equation models.

REFERENCES

- F. Chen, H. Garnier, and M. Gilson. Robust identification of continuous-time models with arbitrary time-delay from irregularly sampled data. *Journal of Process Control*, 25:19–27, 2015.
- H. Garnier and M. Mensler. CONTSID: a continuous-time system identification toolbox for Matlab. In *5th European Control Conference (ECC'99)*, Karlsruhe (Germany), September 1999.
- H. Garnier and L. Wang (Eds). *Identification of continuous-time models from sampled data*. Springer-Verlag, London, 2008.
- H. Garnier and P.C. Young. The advantages of directly identifying continuous-time transfer function models. *International Journal of Control*, 87(7-9):1319–1338, 2014.
- H. Garnier, M. Gilson, V. Laurain, and B. Ni. Developments for the CONTSID toolbox. In *16th IFAC Symposium on System Identification (SYSID'2012)*, Brussels (Belgium), July 2012.
- M. Gilson, J. Welsh, and H. Garnier. Frequency-domain instrumental variable based method for wide band system identification. In *American Control Conference (ACC'2013)*, Washington, DC (USA), June 2013.
- J.L. Guzman, D.E. Rivera, S. Dormido, and M. Berenguel. An interactive software tool for system identification. *Advances in Engineering Software*, 45(1):115 – 123, 2012.
- I. Kollar, R. Pintelon, and J. Schoukens. Frequency domain system identification toolbox for Matlab: characterizing nonlinear errors of linear models. In *14th IFAC Symposium on System Identification (SYSID'2006)*, pages 726–731, Newcastle (Australia), April 2006.
- L. Ljung and R. Singh. Version 8 of the system identification toolbox. In *16th IFAC Symposium on System Identification (SYSID'2012)*, Brussels (Belgium), July 2012.
- B. Ninness, A. Wills, and A. Mills. UNIT: a freely available system identification toolbox. *Control Engineering Practice*, 21(5):631 – 644, 2013.
- J. Schorsch, H. Garnier, M. Gilson, and P. C. Young. Instrumental variable methods for identifying partial differential equation models. *International Journal of Control*, 86(12):2325 – 2335, 2013.
- P.C. Young and C.J. Taylor. Recent developments in the CAPTAIN Toolbox for Matlab. In *16th IFAC Symposium on System Identification (SYSID'2012)*, Brussels (Belgium), July 2012.