

Hugues Garnier • Liuping Wang
Editors

Identification of Continuous-time Models from Sampled Data

 Springer

Hugues Garnier, PhD
Centre de Recherche en Automatique
de Nancy (CRAN)
Nancy-Université
CNRS
Faculté des Sciences et Techniques
54506 Vandoeuvre-les-Nancy
France

Liuping Wang, PhD
RMIT University
School of Electrical and Computing
Engineering
Swanston Street
Melbourne 3000
Victoria
Australia

ISBN 978-1-84800-160-2

e-ISBN 978-1-84800-161-9

DOI 10.1007/978-1-84800-161-9

Advances in Industrial Control series ISSN 1430-9491

British Library Cataloguing in Publication Data
Identification of continuous-time models from sampled data.
- (Advances in industrial control)

1. Linear time invariant systems - Mathematical models -
Congresses 2. Automatic control - Congresses

I. Garnier, Hugues II. Wang, Liuping
629.8'32

ISBN-13: 9781848001602

Library of Congress Control Number: 2007942577

© 2008 Springer-Verlag London Limited

MATLAB® and Simulink® are registered trademarks of The MathWorks, Inc., 3 Apple Hill Drive, Natick, MA 01760-2098, USA. <http://www.mathworks.com>

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licences issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The use of registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

Cover design: eStudio Calamar S.L., Girona, Spain

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

The CONTSID Toolbox: A Software Support for Data-based Continuous-time Modelling

Hugues Garnier¹, Marion Gilson¹, Thierry Bastogne¹ and Michel Mensler²

¹ Nancy-Université, CNRS, France

² Direction de la Recherche, Etudes Avancées, Matériaux - Renault, France

9.1 Introduction

This chapter describes the continuous-time system identification (CONTSID) toolbox for MATLAB[®], which supports continuous-time (CT) transfer function and state-space model identification directly from regularly or irregularly time-domain sampled data, without requiring the determination of a discrete-time (DT) model. The motivation for developing the CONTSID toolbox was first to fill in a gap, since no software support was available to serve the cause of direct time-domain identification of continuous-time linear models but also to provide the potential user with a platform for testing and evaluating these data-based modelling techniques. The CONTSID toolbox was first released in 1999 [15]. It has gone through several updates, some of which have been reported at recent symposia [11, 12, 16]. The key features of the CONTSID toolbox can be summarised as follows:

- it supports most of the time-domain methods developed over the last thirty years [17] for identifying linear dynamic continuous-time parametric models from measured input/output sampled data;
- it provides transfer function and state-space model identification methods for single-input single-output (SISO) and multiple-input multiple-output (MIMO) systems, including both traditional and more recent approaches;
- it can handle irregularly sampled data in a straightforward way;
- it may be seen as an add-on to the system identification (SID) toolbox for MATLAB[®] [26]. To facilitate its use, it has been 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.

The chapter is organised in the following way. Section 9.2 outlines the main steps of the procedure for direct continuous-time model identification. An overview of the identification tools available in the toolbox is given in Section 9.3. An introductory example to the command mode along with a brief

description of the GUI are then presented in Section 9.4. In Section 9.5, the advantages of CT model identification approaches are discussed and illustrated. A few successful application results from real-life process data are described in Section 9.6. Finally, Section 9.7 presents conclusions of the chapter and highlights future developments for the toolbox.

9.2 General Procedure for Continuous-time Model Identification

The procedure to directly determine a continuous-time model of a dynamical system directly from observed time-domain input/output data is similar to the general approach used for traditional DT 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 parametric model estimation 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. More precisely, the iterative procedure involves the following steps:

1. Design an experiment and collect time-domain 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. Estimate the parameters in the chosen model structure according to the input/output data and a given criterion of fit.
5. Examine the finally estimated model properties.

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

As described in the following section, the CONTSID toolbox includes tools for applying the general data-based modelling procedure summarised above.

9.3 Overview of the CONTSID Toolbox

9.3.1 Parametric Model Estimation

The CONTSID toolbox offers a variety of parametric model estimation methods for the most common input/output and state-space model structures.

CT ARX Models

The CT autoregressive with external input (ARX) model structure considered here, takes the form

$$A(p)y(t_k) = B(p)u(t_k - \tau) + e(t_k) \quad (9.1)$$

with

$$B(p) = b_1 p^{n_b-1} + b_2 p^{n_b-2} + \dots + b_{n_b}, \quad (9.2)$$

$$A(p) = p^{n_a} + a_1 p^{n_a-1} + \dots + a_{n_a}, \quad n_a \geq n_b - 1 \quad (9.3)$$

where p denotes the differential operator; $u(t_k)$ and $y(t_k)$ represent the deterministic input and noisy output signals at time instant t_k , respectively; $e(t_k)$ is a zero-mean DT white Gaussian sequence³ with variance σ_e^2 ; τ is a pure time delay in time units. The latter will be assumed in the following to be an integer number related to the sampling time T_s , *i.e.*, $\tau = t_{n_k} = n_k T_s$. Note however that this is not essential in this CT environment, ‘fractional’ time delays can be introduced if required (*e.g.*, see Chapter 11 and [27, 49]).

The CT ARX model structure, when the time delay is supposed to be an integer multiple of the sampling period, is denoted in the SISO case by the triad $[n_a \ n_b \ n_k]$.

Here, the integers n_a and n_b are the number of parameters to be estimated in each polynomial while n_k is the number of delays from input to output. Note that (9.1) also applies in a straightforward manner to the multiple-input case, with n_u input channels

$$A(p)y(t_k) = B_1(p)u_1(t_{k-n_{k_1}}) + \dots + B_{n_u}(p)u_{n_u}(t_{k-n_{k_{n_u}}}) + e(t_k) \quad (9.4)$$

The CT MISO ARX model structure is denoted by $[n_a \ n_{b_1} \ \dots \ n_{b_{n_u}} \ n_{k_1} \ \dots \ n_{k_{n_u}}]$.

Equation (9.1) can also be explicitly written as

$$y^{(n_a)}(t_k) + a_1 y^{(n_a-1)}(t_k) + \dots + a_{n_a} y(t_k) = b_1 u^{(n_b-1)}(t_{k-n_k}) + \dots + b_{n_b} u(t_{k-n_k}) + e(t_k) \quad (9.5)$$

where $x^{(i)}(t_k)$ denotes the i th time derivative of the continuous-time signal $x(t)$ at time instant $t_k = kT_s$.

It may be noted that in contrast to the difference-equation model, the differential-equation model (9.5) is not a linear combination of samples of only the measurable process input and output signals. It also contains time-derivative terms that are not available as measurement data in most practical cases. The general scheme for CT ARX model estimation then requires two stages [17]:

³ The disturbance term is modelled here as a zero-mean discrete-time Gaussian noise sequence. This avoids mathematical difficulties associated with continuous-time stochastic process modelling (see, *e.g.*, [1] and Chapter 2).

- the primary stage that consists in using a preprocessing method to generate some measures of the process signals and their time derivatives. This stage also includes finding an approximating or discretizing technique so that the preprocessing operation can be performed in a purely digital way from sampled input/output data;
- the secondary stage in which the CT parameters are estimated within the framework of a LS or IV-based linear regression methods. Most of the well-known LS or IV-based methods developed for DT parameter estimation can be extended to the CT case with slight modifications.

Therefore, the main difference from conventional DT ARX model identification lies in the primary stage. There is a range of choice for the preprocessing required in the primary stage. Each method is characterised by specific advantages such as mathematical convenience, simplicity in numerical implementation and computation, physical insight, accuracy and others. However, all perform some prefiltering on the process signals. Process signal prefiltering is indeed a very useful and important way to improve the statistical efficiency in system identification and yields lower variance of the parameter estimates. Preprocessing methods developed over the last thirty years are traditionally grouped into three main classes of methods that are summarised below. The main references that have been used as the basis for their implementation along with their acronym used in the toolbox, are also given (see [17]):

- for the linear filters: the state-variable filter (SVF) [52] and the generalised Poisson moment functionals (GPMF) [18, 42];
- for the modulating functions: the Fourier [34] and Hartley modulating functions (HMF) [43];
- for the integral methods:
 - *among the numerical integration methods*: the block-pulse functions (BPF) [9], the trapezoidal-pulse functions (TPF) [9] and the Simpson's rule of integration (SIMPS) [7];
 - *among the orthogonal functions*: the Fourier trigonometric functions (FOURIE) [33], the Walsh functions (WALSH) [8], for the orthogonal polynomials: Hermite (HERMIT) [33], Laguerre (LAGUER) [33], Legendre (LEGEND) [33], first and second kind of Chebychev polynomials (CHEBY1 and CHEBY2) [32];
 - *among the others methods*: the linear integral filter (LIF) [39] and the re-initialised partial moments (RPM) [35].

Several parameter estimation algorithms associated with all implemented preprocessing techniques are available for identifying CT ARX models of the form of (9.1) or (9.5). First, conventional least squares (LS)-based methods have been implemented. In order to overcome the bias problem associated with simple LS-based estimation in the presence of noisy output data, a two-step instrumental variable (IV) estimator where the instruments are built up from an auxiliary model, has also been coupled with all available preprocessing

Table 9.1. Available methods for CT ARX model identification

Preprocessing methods		LS		IV		BCLS WLS	
		SISO	MISO	SISO	MISO	SISO	SISO
linear filters	SVF	✓	✓	✓	✓		
	GPMF	✓	✓	✓	✓	✓	
modulating functions	FMF	✓	✓	✓	✓	✓	✓
	HMF	✓	✓	✓	✓		✓
specific	RPM	✓	✓	✓	✓		
	LIF	✓	✓	✓	✓		
numerical integration	BPF	✓		✓			
	TPF	✓		✓			
	SIMPS	✓		✓			
integral methods	orthogonal functions	FOURIE	✓	✓			
		WALSH	✓	✓			
orthogonal polynomials		CHEBY1	✓	✓			
		CHEBY2	✓	✓			
		HERMIT	✓	✓			
		LAGUER	✓	✓			
		LEGEND	✓	✓			

techniques [17]. A few specific bias-reduction algorithms are also included, like a weighted least squares (WLS) associated with the modulating functions or a bias-compensating least squares (BCLS) technique coupled with the GPMF approach [19]. For details on the different parametric estimation methods for the CT ARX model identification, the reader is referred to [17]. Table 9.1 lists the methods available in the CONTSID toolbox for SISO and MISO CT ARX model identification. The performances of the sixteen pre-filtering methods mentioned above have been thoroughly analysed by Monte Carlo simulation. Simulation [17,30] and real-life [28] studies have shown that integral methods (this does not include the RPM and LIF techniques) can have quite poor performances in the presence of medium to high measurement output noise. In particular, they require the estimation of additional initial condition parameters and have a high sensitivity to their user parameters. However, six of the methods exhibit very good overall performance: these are based on linear filters (GPMF and SVF), on modulating functions (FMF and HMF), and on the two particular types of integral methods (LIF and RPM). The final choice for a particular approach will probably depend on the taste or experience of the user since their global performances are very close. It is, therefore, not necessary to be able to tell which of the approaches is ‘best’. Experience says that each may have its advantages. It is, however, good practice to have them all in one’s toolbox. Furthermore, these methods can be used to get an initial high-quality estimate for iterative parametric estimation methods, presented in the next paragraphs.

CT Hybrid OE Models

The so-called CT hybrid output-error (OE) model structure is given by

$$y(t_k) = \frac{B(p)}{F(p)}u(t_{k-n_k}) + e(t_k) \quad (9.6)$$

with

$$F(p) = p^{n_f} + f_1 p^{n_f-1} + \dots + f_{n_f}, \quad n_f \geq n_b - 1 \quad (9.7)$$

where the noise $e(t_k)$ is assumed to be a zero-mean DT white Gaussian sequence so that no explicit noise modelling is necessary, except in relation to the estimation of the variance of the DT white noise process.

The CT hybrid OE model structure is denoted in the SISO case by the triad $[n_b \ n_f \ n_k]$.

Here, the integers n_b and n_f are the number of parameters to be estimated in each polynomial ; n_k is the number of delays from input to output. For a multiple-input systems, (9.6) becomes

$$y(t_k) = \frac{B_1(p)}{F_1(p)}u_1(t_{k-n_{k_1}}) + \dots + \frac{B_{n_u}(p)}{F_{n_u}(p)}u_{n_u}(t_{k-n_{k_u}}) + e(t_k) \quad (9.8)$$

The CT hybrid MISO OE model structure is denoted by $[n_{b_1} \ \dots \ n_{b_{n_u}} \ n_{f_1} \ \dots \ n_{f_{n_u}} \ n_{k_1} \ \dots \ n_{k_{n_u}}]$.

Two methods for identifying MISO OE structure-based models with different denominators are available in the toolbox.

The first is based on the iterative simplified refined instrumental variable method for continuous-time model identification (SRIVC: see Chapter 4). This approach involves a method of adaptive prefiltering based on an optimal statistical solution to the problem in this white noise case. This SRIVC method has been recently extended to handle MISO systems described by multiple CT transfer functions with different denominators [13] of the form of (9.8). It is important to mention that for day-to-day usage, the SRIVC algorithm provides a quick and reliable approach to CT model identification and has been used for many years as the algorithm of choice for this in the CAPTAIN toolbox⁴ and, more recently, in the CONTSID toolbox. The application results of the SRIVC method to different real-life processes are further presented in this chapter.

The second method abbreviated by COE (continuous-time output error) implements the Levenberg–Marquardt or Gauss–Newton algorithm via sensitivity functions [38]. In contrast to LS- and IV-based methods, these algorithms rely on a numerical search procedure with a risk to get stuck in local minima and also require a larger amount of computation.

⁴ See <http://www.es.lancs.ac.uk/cres/captain/>.

Table 9.2. Available methods for CT hybrid OE and BJ model identification

Methods	OE		BJ
	SISO	MISO	SISO
COE	✓	✓	
SRIVC	✓	✓	
RIVC			✓

CT Hybrid BJ Models

The so-called CT hybrid Box–Jenkins (BJ) model structure in the SISO case is given by

$$y(t_k) = \frac{B(p)}{F(p)}u(t_{k-n_k}) + \frac{C(q^{-1})}{D(q^{-1})}e(t_k) \tag{9.9}$$

with

$$C(q^{-1}) = 1 + c_1q^{-1} + \dots + c_qq^{-q} \tag{9.10}$$

$$D(q^{-1}) = 1 + d_1q^{-1} + \dots + d_pq^{-p} \tag{9.11}$$

where $e(t_k)$ is a zero-mean DT white Gaussian sequence. Here, the model of the basic dynamic system is in continuous time, while the associated additive noise model is a discrete-time, autoregressive moving-average (ARMA) process (see Chapter 4). This CT hybrid BJ model structure is denoted by the following model order structure $[n_b \ n_c \ n_d \ n_f \ n_k]$.

One of the main advantage of the CT hybrid BJ model is the asymptotic independence of the process and noise estimates. An approach based on the refined optimal IV, denoted by RIVC, has been derived to estimate the parameters of such models (see Chapter 4).

Table 9.2 lists the methods available in the CONTSID toolbox for CT hybrid OE and BJ model identification.

CT State-space Models

Continuous-time state-space models considered in the CONTSID toolbox take the form

$$\begin{cases} \dot{x}(t_k) = Ax(t_k) + Bu(t_k) \\ y(t_k) = Cx(t_k) + Du(t_k) + \xi(t_k) \end{cases} \tag{9.12}$$

where $u(t_k) \in \mathbb{R}^{n_u}$ is the input vector and $y(t_k) \in \mathbb{R}^{n_y}$ the output vector and $x(t_k) \in \mathbb{R}^n$ is the state vector at time t_k , $\xi(t_k) \in \mathbb{R}^{n_y}$ is the possibly coloured output noise vector.

Table 9.3. Available methods for CT state-space model identification

	Canonical model			Fully parameterised model
	LS	IV	BCLS	N4SID
GPMF	✓	✓	✓	✓
FMF				✓
HMF				✓
RPM				✓
LIF				✓

Two types of approaches for CT state-space model identification are available in the CONTSID toolbox. A first family of techniques relies on the *a priori* knowledge of structural indices, and considers the estimation of CT canonical state-space models. From the knowledge of the observability indices, the canonical state-space model can, in a straightforward way, be first transformed into an equivalent input–output polynomial description that is linear-in-its-parameters and therefore more suitable for the parameter estimation problem. A preprocessing method may then be used to convert the differential equation into a set of linear algebraic equations in a similar way to that for CT ARX type of models. The unknown model parameters can finally be estimated by LS-, IV- or BCLS-based algorithms [18, 20]. This scheme has been implemented for the GPMF approach only.

A second class of state-space model identification schemes is based on the subspace-estimation techniques. Most efficient data-based modelling methods, discussed so far, rely on iterative, non-linear optimisation or IV-type methods to fit parameters in a preselected model structure, so as to best fit the observed data. Subspace methods are an alternative class of identification methods that are ‘one-shot’ rather than iterative, and rely on linear algebra.

Moreover, these subspace methods are attractive since canonical forms are not required, while fully parameterised state-space models are estimated directly from sampled I/O data. Most commonly known subspace methods were developed for DT model identification [44]. The association of the more efficient preprocessing methods with subspace methods of the 4SID family [44] has been implemented in the toolbox [4, 29] (see also Chapter 10 in this book). Table 9.3 summarises the methods available in the CONTSID toolbox for CT state-space model identification.

The application results of the GPMF-based subspace algorithm to a multiple-input multiple-output winding process are presented in Section 9.6.3.

9.3.2 Model Order Selection and Validation

The toolbox also includes tools for selecting the model orders as well as for evaluating the estimated model properties.

Model Order Selection

Model order selection is one of the difficult tasks in the system identification procedure. A natural way to find the most appropriate model orders is to compare the results obtained from model structures with different orders and delays. A model order selection algorithm associated to the SRIVC model estimation method allows the user to automatically search over a whole range of different model orders. Two statistical measures are then used for the analysis. The first is the simulation coefficient of determination R_T^2 , defined as follows

$$R_T^2 = 1 - \frac{\sigma_{\hat{e}}^2}{\sigma_y^2} \quad (9.13)$$

where $\sigma_{\hat{e}}^2$ is the variance of the estimated noise $\hat{e}(t_k)$ and σ_y^2 is the variance of the measured output $y(t_k)$. This should be differentiated from the standard coefficient of determination R^2 , where the $\sigma_{\hat{e}}^2$ in (9.13) is replaced by the variance of the final noise model residuals $\hat{\sigma}^2$. R_T^2 is clearly a normalised measure of how much of the output variance is explained by the deterministic system part of the estimated model. However, it is well known that this measure, on its own, is not sufficient to avoid overparametrisation and identify a parsimonious model, so that other order identification statistics are required. In this regard, because the SRIVC method exploits optimal instrumental variable methodology, it is able to utilise the special properties of the instrumental product matrix (IPM) [45, 53]; in particular, the YIC statistic [47] is defined as follows

$$\text{YIC} = \log_e \frac{\hat{\sigma}^2}{\sigma_y^2} + \log_e \{\text{NEVN}\}; \quad \text{NEVN} = \frac{1}{n_\theta} \sum_{i=1}^{n_\theta} \frac{\hat{p}_{ii}}{\hat{\theta}_i^2} \quad (9.14)$$

Here, n_θ is the number of estimated parameters; \hat{p}_{ii} is the i th diagonal element of the block-diagonal SRIVC covariance matrix and so is an estimate of the variance of the estimated uncertainty on the i th parameter estimate. $\hat{\theta}_i^2$ is the square of the i th SRIVC parameter estimate, so that the ratio $\hat{p}_{ii}/\hat{\theta}_i^2$ is a normalised measure of the uncertainty on the i th parameter estimate.

From the definition of R_T^2 , we see that the first term in the YIC is simply a relative measure of how well the model explains the data: the smaller the model residuals, the more negative the term becomes. The normalised error variance norm (NEVN) term, on the other hand, provides a measure of the conditioning of the IPM, which needs to be inverted when the IV normal equations are solved (see *e.g.*, [46]): if the model is overparameterised, then it can be shown that the IPM will tend to singularity and, because of its ill-conditioning, the elements of its inverse will increase in value, often by several orders of magnitude. When this happens, the second term in the YIC tends to dominate the criterion function, indicating overparametrisation.

Although heuristic, the YIC has proven very useful in practical identification terms. It should not, however, be used as a sole arbiter of model order: rather

the combination of R_T^2 and YIC provides an indication of the best parsimonious models that can be evaluated by other standard statistical measures (*e.g.*, the autocovariance of the model residuals, the cross-covariance of the residuals with the input signal $u(t_k)$, *etc.*). Also, within a ‘data-based mechanistic’ (DBM) model setting (see, *e.g.*, [48]), the physical interpretation of the model can often provide valuable information on the model adequacy: for instance, a model with complex eigenvalues caused by overparametrisation may prove incompatible with the non-oscillatory nature of the physical system under study.

The CONTSID toolbox includes a `srivcstruc` routine that allows the user to automatically search over a range of different orders by using the SRIVC algorithm and computes the two loss functions YIC and R_T^2 . The in-line help specifies the required input parameters for the `srivcstruc` function

```
data=iddata(y,u,Ts);
V=srivcstruc(data,[],modstruc);
```

The routine collects in a matrix `modstruc` all the CT hybrid OE model to be investigated so that each row of `modstruc` is of the type $[n_{b_1} \dots n_{b_{nu}} \ n_{f_1} \dots n_{f_{nu}} \ n_{k_1} \dots n_{k_{nu}}]$, where n_{b_j} and n_{f_j} are the number of parameters for the numerator and denominator, respectively, and n_{k_j} represents the number of samples for the delay. Then, a continuous-time model is fitted to the `iddata` set `data` for each of the structures in `modstruc`. For each of these estimated models, the two loss functions YIC and R_T^2 are computed from this estimation data set. The best model structures sorted according to the chosen criterion (‘YIC’ or ‘RT2’) are displayed with

```
selcstruc(V,criterion,nu);
```

where `nu` indicates the number of inputs. The application results of this model order selection procedure are illustrated further in this chapter.

Experiment Design, Model Validation and Simulation

In addition to the parameter estimation and model order determination routines, the toolbox provides several functions in order to generate excitation signals, simulate and examine CT models (see Table 9.4).

A few functions are available to generate excitation signals: `prbs` allows the design of a pseudo-random binary signal of maximum length, while `sineresp` returns the exact steady-state response of a continuous-time model for a sum of sine signals.

Simulated data can then be generated by using the function `simc` that allows the simulation of a CT model under an `idss` or `idpoly` format from a given `iddata` input object from regularly or irregularly sampled data.

Two functions are available for model validation purposes: `comparec` displays the measured output with the identified model output, while `residc` plots

the auto-covariance function of the model residuals and the cross-covariance function of the residuals with the input signal.

Note that most of the functions included in the SID toolbox for the computation and presentation of frequency functions and zeros/poles (`bode`, `zplot`) can be used with the identified CONTSID models.

The main demonstration program called `idcdemo` provides several examples illustrating the use and the relevance of the CONTSID toolbox approaches. These demos also illustrate what might be typical sessions with the CONTSID toolbox.

Table 9.4. CONTSID toolbox programs for experiment design, model simulation, order selection and validation

Program	Description
<code>idcdemo</code>	is the main routine for the CONTSID toolbox demonstration programs
<code>prbs</code>	generates a pseudo-random binary sequence of maximum length
<code>sineresp</code>	generates the exact steady-state response of a CT model for a sum of sine signals
<code>sineresp1</code>	generates the exact steady-state response of a CT first-order model for a sine given arbitrary initial conditions
<code>sineresp2</code>	generates the exact steady-state response of a CT second-order model for a sine given arbitrary initial conditions
<code>simc</code>	simulates a system under its CT <code>idpoly</code> or <code>idss</code> form
<code>comparec</code>	compares measured and model outputs
<code>residc</code>	computes and plots the residuals of a CT model. Plots the autocovariance function of the model residuals, the cross-covariance function of the residuals with the input signal
<code>srivcstruc</code>	computes the fit between simulated and measured outputs for a set of model structure of CT hybrid OE type estimated using the <code>srivc</code> method
<code>selcstruc</code>	helps to choose a model structure from the information obtained as the output from <code>srivcstruc</code>

Recursive Estimation

In many situations, there is a need to estimate the model at the same time as the data is collected during the measurement. The model is then ‘updated’ at each time instant some new data become available. The updating is performed by a recursive algorithm. Recursive versions RLSSVF, RIVSVF and RSRIVC of the LS, IV-based SVF methods and optimal IV technique for CT hybrid OE models are available in the CONTSID toolbox.

Identification from Non-uniformly Sampled Data

The problem of system identification from non-uniformly sampled data is of importance as this case occurs in several applications (see, *e.g.*, Chapter 11). The case of irregularly sampled data is not easily handled by discrete-time model identification techniques, but as illustrated further in this chapter (see Sections 9.4.1 and 9.6.2), mild irregularity can be easily handled by some of the CONTSID toolbox methods. This is because the differential-equation model is valid whatever the time instants considered and, in particular, it does not assume regularly sampled data, as required in the case of the standard difference-equation model.

Table 9.5 lists the functions available for data-based modelling from irregularly sampled data.

Table 9.5. Available functions for CT model identification from irregularly sampled data

Program	Description
LSSVF	LS-based state-variable filter method for CT ARX models
IVSVF	IV-based state-variable filter method for CT ARX models
COE	non-linear optimisation method for CT hybrid OE models
SRIVC	optimal instrumental variable method for CT hybrid OE models
SIDGPMF	subspace-based generalised Poisson moment functionals method for CT state-space models

9.4 Software Description

The CONTSID toolbox is compatible with MATLAB[®] versions 6.x and 7.x. Two external commercial toolboxes are required: the Control toolbox and the SID toolbox. The current version can be considered as an add-on to the SID toolbox and makes use of the `iddata`, `idpoly` and `idss` objects used in the SID toolbox. It is freely available for academic researchers and can be downloaded from

<http://www.cran.uhp-nancy.fr/contsid/>

All available parametric model estimation functions share the same command structure

```
m = function(data,modstruc)
m = function(data,modstruc,specific_parameters)
```

The input argument `data` is an `iddata` object that contains the output- and input-data sequences along with the sampling time and inter-sample behaviour for the input, while `modstruc` specifies the particular structure of the

model to be estimated. The specific parameters depend on the preprocessing method used. The resulting estimated model is contained in `m`, which is a model object that stores the various usual information. The function name is defined by the abbreviation for the estimation method and the abbreviation for the associated preprocessing technique, as for example, IVSVF for the instrumental variable-based state-variable filter approach or SIDGPMF for subspace-based state-space model identification GPMF approach. Note that help on any CONTSID toolbox function may be obtained from the command window by invoking classically `help name.function`.

9.4.1 Introductory Example to the Command Mode

A part of the first demonstration program is presented in this section. This demo is designed to get the new user started quickly with the CONTSID toolbox: it is straightforward to run the demo by typing `idcdemo1` in the MATLAB® command window) and follow along. This example considers a second-order SISO CT system without delay. The complete equation for the data-generating system has the following form

$$y(t_k) = \frac{3}{p^2 + 4p + 3}u(t_k) + e(t_k) \quad (9.15)$$

where $e(t_k)$ is a zero-mean DT white Gaussian noise sequence. Let us first create an `idpoly` model structure object describing the model. The polynomials are entered in descending powers of the differential operator

```
m0=idpoly(1,[3],1,1,[1 4 3], 'Ts',0);
```

'Ts' and 0 indicate here that the system is time continuous.

We take a PRBS of maximum length with 1016 points as input `u`. The sampling period is chosen to be 0.05 s

```
u = prbs(7,8);
Ts = 0.05;
```

We then create an `iddata` object for the input signal with no output, the input `u` and sampling interval `Ts`. The input inter-sample behaviour is specified by setting the property '`Intersample`' to '`zoh`' since the input is piecewise-constant here

```
datau = iddata([],u,Ts,'InterSample','zoh');
```

The noise-free output is simulated with the `simc` CONTSID routine and stored in `ydet`. We then create an `iddata` object with output `ydet`, input `u` and sampling interval `Ts`

```
ydet = simc(m0,datau);
datadet = iddata(ydet,u,Ts,'InterSample','zoh');
```

We then identify a CT ARX model for this system from the deterministic `iddata` object `datadet` with the conventional least squares-based state-variable filter (`lssvf`) method. The extra pieces of information required are

- the number of denominator and numerator parameters and number of samples for the delay of the model $[n_a \ n_b \ n_k] = [2 \ 1 \ 0]$;
- the cut-off frequency (in rad/s) of the SVF filter, set to 2 here.

The `lssvf` routine can now be used as follows

```
m_lssvf = lssvf(datadet,[2 1 0],2)
```

which leads to⁵

```
CT IDPOLY model: A(s)y(t) = B(s)u(t) + e(t)
A(s) = s^2 + 3.999 s + 2.999
B(s) = 2.999
Estimated using LSSVF
Loss function 6.03708e-15 and FPE 6.07284e-15
```

It will be noted that, not surprisingly, the estimated model coefficients are very close to the true parameters. This is, of course, because the measurements are not noise corrupted. Note that even in the noise-free case, the true parameters are not estimated exactly here. This is due to small simulation errors introduced in the numerical implementation of the continuous-time state-variable filtering for the output signal.

Let us now consider the case when a white Gaussian noise is added to the output samples. The variance of $e(t_k)$ is adjusted to obtain a signal-to-noise ratio (SNR) of 10 dB. The SNR is defined as

$$\text{SNR} = 10 \log \frac{P_{y_{det}}}{P_e} \quad (9.16)$$

where P_e represents the average power of the zero-mean additive noise on the system output (*e.g.*, the variance) while $P_{y_{det}}$ denotes the average power of the noise-free output fluctuations.

```
snr=10;
y = simc(m0,datau,snr);
data = iddata(y,u,Ts);
```

The input/output data are displayed in Figure 9.1. The use of this noisy output in the basic `lssvf` routine will inevitably lead to biased estimates. A bias-reduction algorithm based on the two-step instrumental variable technique where the instruments are built up from an auxiliary model (`ivsvf`) can be used instead

⁵ Note that in the Matlab[®] System Identification (SID) toolbox, the variable 's' instead of 'p' is used to denote the differential operator. The CONTSID toolbox makes use of the SID object models ; therefore the CONTSID estimated models are displayed with the 's' variable.

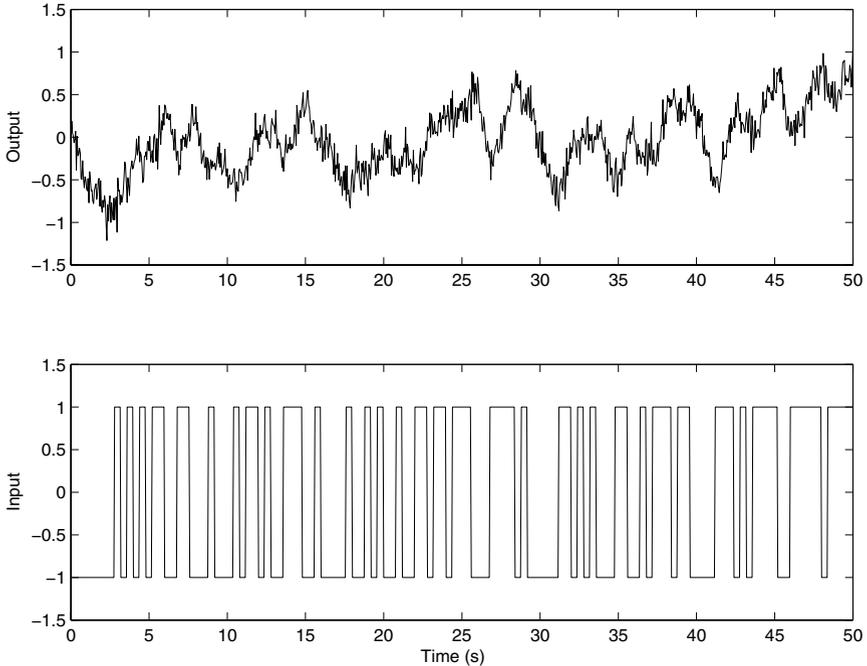


Fig. 9.1. Input–output data (noisy case) – SNR=10 dB

```
mivsvf=ivsvf(data,[2 1 0],2)
```

which leads to

CT IDPOLY model: $A(s)y(t) = B(s)u(t) + e(t)$

$A(s) = s^2 + 3.988 s + 3.076$

$B(s) = 3.008$

Estimated using IVSVF

Loss function 0.217742 and FPE 0.219032

It will be noted now that the parameters are close to the true ones. However, this basic IV-based SVF method suffers from two drawbacks, even if it is asymptotically unbiased:

- it requires the *a priori* knowledge of a user parameter: the cut-off frequency of the SVF filter here;
- it is suboptimal, in the sense that the variance of the estimates is not minimal (it depends of the SVF filter mainly).

It is better, therefore, to use the optimal (in this white output measurement noise context) iterative IV method (`srivc`) that overcomes the two latter drawbacks. The searched model now takes the form of a CT hybrid OE model

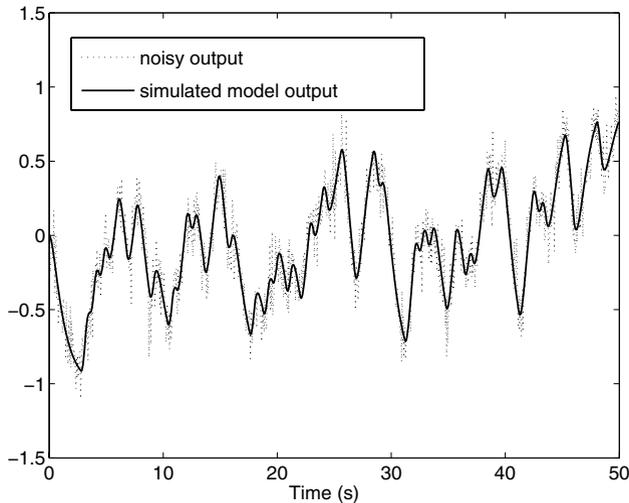


Fig. 9.2. Noisy and simulated SRIVC model outputs

(9.6). The model structure becomes $[n_b \ n_f \ n_k] = [1 \ 2 \ 0]$.

The `srivc` routine can now be used as follows

```
msrivc = srivc(data,[1 2 0]);
```

The estimated parameters together with the estimated standard deviations can be displayed

```
present(msrivc);
```

which leads to

```
CT IDPOLY model: y(t) = [B(s)/F(s)]u(t) + e(t)
B(s) = 3.002 (+-0.1113)
F(s) = s2 + 3.992 (+-0.1619) s + 3.067 (+-0.1061)
Estimated using SRIVC
Loss function 0.0135763 and FPE 0.0136567
```

Let us now compare the model output for the input signal with the measured output. This can be done easily by using the `comparec` CONTSID routine

```
comparec(data,msrivc,1:1000);
```

which plots the measured and the simulated model outputs. As can be seen in Figure 9.2, they coincide very well. We can also check the residuals of this model, and plot the autocovariance of the residuals and the cross-covariance between the input and the residuals by using the CONTSID `residc` routine

```
residc(data,msrivc);
```

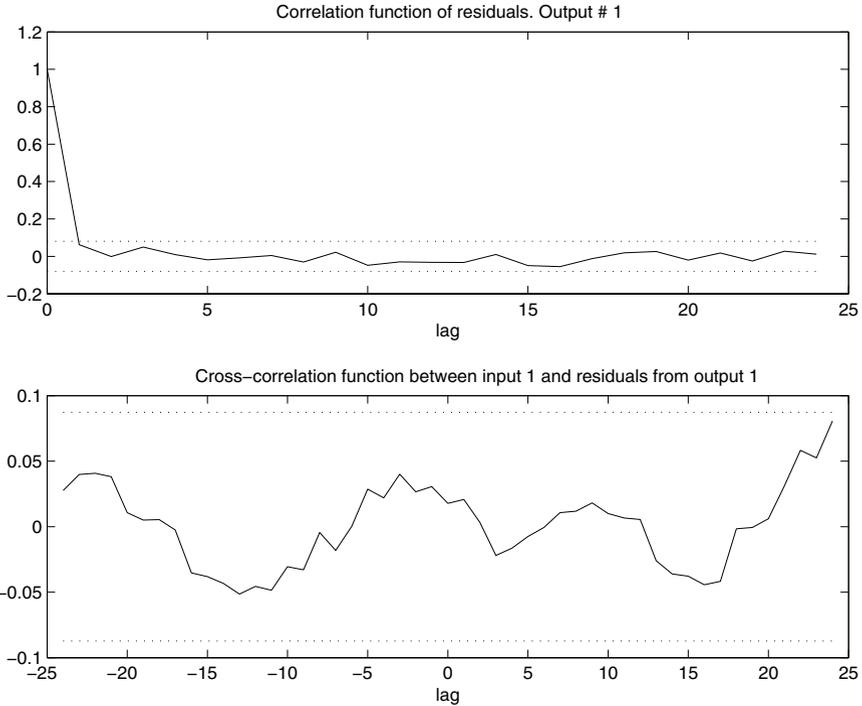


Fig. 9.3. Correlation test for the SRIVC model

From Figure 9.3, it may be seen that the residuals are white and totally uncorrelated with the input signal. We can thus be satisfied with the model. Let us finally compare the Bode plots of the estimated model and the true system

```
bode(msrivc,'sd',3,'fill',m0,'y--')
```

The confidence regions corresponding to three standard deviations are also displayed. From Figure 9.4, it may be observed that the Bode plots coincide very well with narrow confidence regions.

As previously mentioned, non-uniformly sampled data can be handled easily by some CONTSID toolbox methods [21]. This is now illustrated here. The data-generating system has the following form

$$y(t_k) = \frac{5}{p^2 + 2.8p + 4} u(t_k) + e(t_k) \quad (9.17)$$

The input signal is chosen as the sum of three sines,

$$u(t) = \sin(0.714t) + \sin(1.428t) + \sin(2.142t)$$

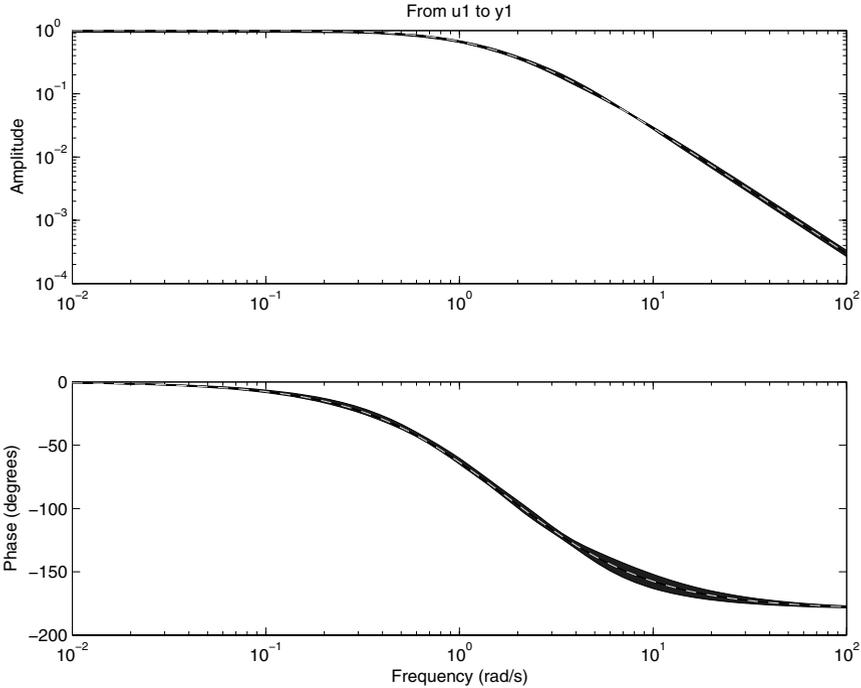


Fig. 9.4. Bode plot of the SRIVC model together with a 3 standard deviation uncertainty region

A non-uniform sampling setup similar to the one used in [23] is chosen. The distance between two sampling instants is denoted by $h_k = t_{k+1} - t_k$. We assume that $\underline{h} \leq h_k \leq \bar{h}$, where $\underline{h} > 0$ and \bar{h} are the finite lower and upper bounds, respectively. A uniform probability density function $U(\underline{h}, \bar{h})$ is used to describe the variations of the sampling interval, *i.e.*, $h_k \sim U(0.01s, 0.1s)$. 3000 data points are used for the identification. Analytic expressions are used to compute the noise-free output in order to avoid errors due to numerical simulations. A zero-mean white noise is then added to the system output in order to get a signal-to-noise ratio of 10 dB. The simulated output is stored in y . Figure 9.5 displays a short section of 3 s of the sampled records and reveals the non-uniform sampling intervals.

We first create a `iddata` object with output y , input u , and the available time instant stored in the vector t . The input inter-sample behaviour is specified by setting the property 'Intersample' to 'foh' since the input is not piecewise-constant here

```
data = iddata(y,u,[],'SamplingInstants',t,'InterSample','foh');
```

The optimal IV algorithm `srivc` can now be used for the appropriate model structure

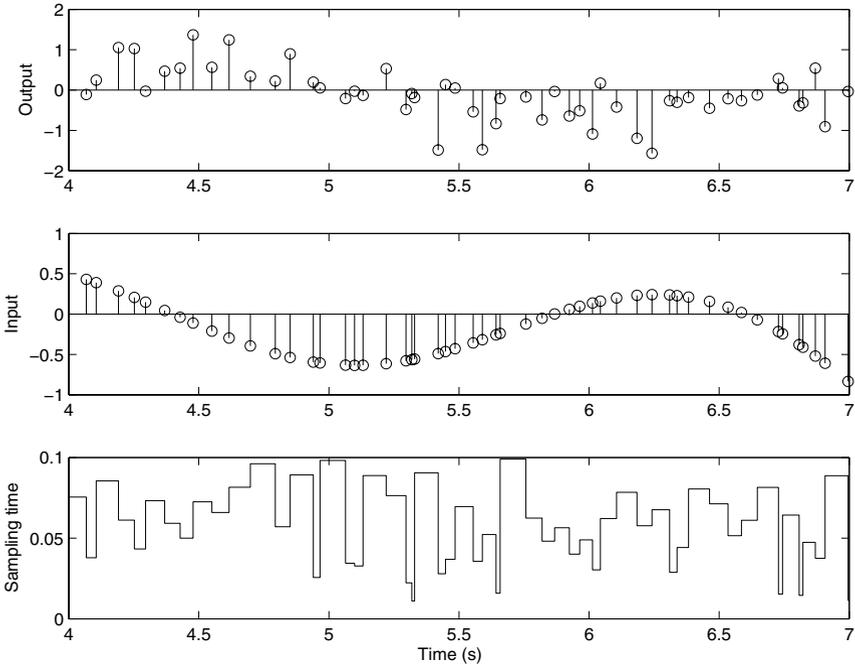


Fig. 9.5. A section of the noisy output, input signal and sampling interval value in the case of irregularly sampled data – SNR=10 dB

```
msrivc=srivc(data,[1 2 0]);
```

The estimated parameters and standard errors can then be displayed with

```
present(msrivc);
```

which leads to

```
Continuous-time IDPOLY model:
y(t) = [B(s)/F(s)]u(t)
B(s) = 4.958 (+-0.0909)
F(s) = s2 + 2.796 (+-0.05637) s + 3.994 (+-0.05056)
Loss function 0.1748 and FPE 0.1752
Estimated using SRIVC
```

The estimated parameters are very close to the true ones.

As previously, traditional model validation tests can be performed to further examine the quality of the estimated model.

9.4.2 The Graphical User Interface

The graphical user interface (GUI) for the CONTSID toolbox provides a main window, as shown in Figure 9.6, which is divided into three basic parts:

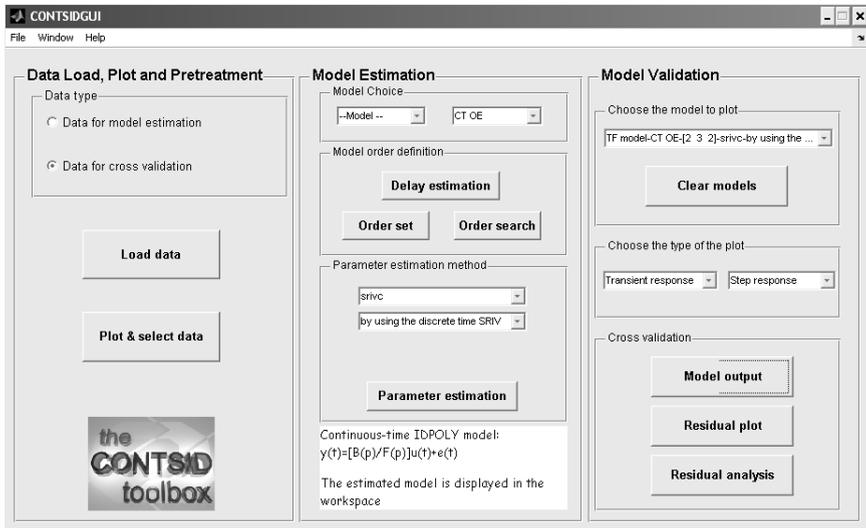


Fig. 9.6. The main window of the CONTSID GUI

- a data panel on the left part where data-sets can be imported, plotted, pretreated and selected;
- a model estimation panel in the middle where CT ARX and CT hybrid OE transfer function model structures and associated parameter estimation methods 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.

The Data Panel

As shown in Figure 9.6, the GUI lets the user define two data-sets: one for identifying the model and one, if sufficient data are available, for cross-validation purposes.

Importing Measured Data

By clicking on the *Load data* button, time-domain sampled data from a *.mat* file can be imported for systems with single- or multiple- input and single-output channels. From this window, the input and output variables can be specified along with the type of sampling scheme (regular or irregular), the sampling time (T_s) and the assumption on the input inter-sample behaviour (piecewise-constant (zoh) or continuous).

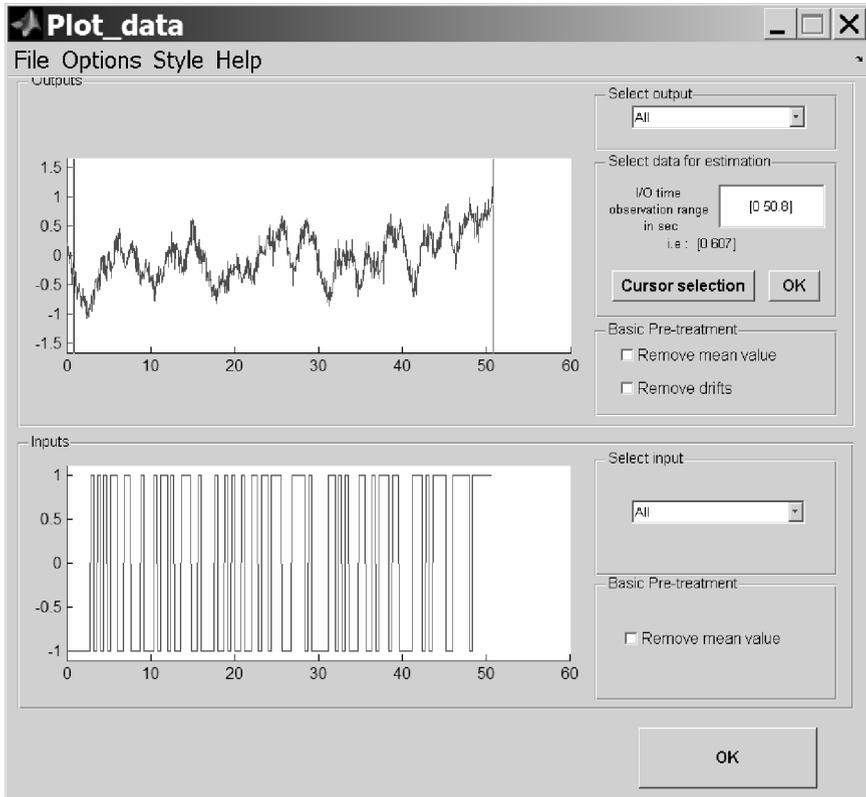


Fig. 9.7. Data plot and pretreatment GUI window

Preprocessing and Selecting Observed Data

After the data have been imported, basic operations for data analysis and preprocessing can be applied. An example of the window obtained after a click on the button *Plot & select data* is displayed in Figure 9.7. This window also allows the preprocessing of data including offset, drift removal and the display of the results after the operation.

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

- the data-sets include erroneous values that it is essential to eliminate;
- if only one data-set is available, it is advisable to divide the data-set into two parts, the first for model estimation purposes and the second reserved for cross-validation purposes.

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

Model Estimation Panel

While the CONTSID toolbox supports transfer function and state-space model identification methods, the GUI lets the user estimate CT ARX and hybrid OE models only. The user is thus invited to choose the type of model structure in the unrolling menu at the top right of the *model estimation* panel, as shown in Figure 9.6.

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 orders are not known *a priori*, the *Order search* button allows the user to automatically search over a whole range of different model orders. 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 the number of samples for the time delay and the number of coefficients for the polynomial model have been set, the model parameters can then be estimated by using one of the available parametric estimation methods chosen from an unrolling menu

- in the case of a CT hybrid 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 preprocessing-based methods that have proven successful. These preprocessing methods are coupled with conventional least squares or basic auxiliary model-based instrumental variable methods. These all require a user parameter to be specified by the user [17] which should be chosen in order to emphasise the frequency band of interest.

Once the parameter estimation method is chosen, the identified model is displayed in the command window after a click on the *Parameter estimation* button.

Model Validation Panel

Once a model is estimated, it appears in the drop-down menu located at the top part of the *Model Validation* panel (see Figure 9.6). 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. This indicates how well the system dynamics are captured;

- *residual plot*: displays the residuals;
- *transient response*: displays the model response to an impulse or step excitation signal;
- *frequency response*: displays the Nyquist or Bode plots to show damping levels and resonance frequencies;
- *zeros and poles*: plots the zeros and poles of the identified models and tests for zero-pole cancelation indicating overparameterised modelling;
- *correlation test*: displays the autocovariance function of the residuals and the cross-covariance function between the input signal and the residuals.

If a cross-validation data-set is available, then traditional cross-validation tests consist of comparisons between the measured and simulated model outputs and analysis of the residuals.

9.5 Advantages and Relevance of the CONTSID Toolbox Methods

There are two fundamentally different time-domain approaches to the problem of obtaining a CT model of a naturally CT system from its sampled input/output data:

- the indirect approach that involves two steps. First, a DT model for the original CT system is obtained by applying the DT model estimation methods and the DT model is then transformed into CT form;
- the direct approach where a CT model is obtained straightaway using CT model identification methods discussed in this chapter.

The indirect approach has the advantage that it uses well-established DT model identification methods [24, 41]. Examples of such methods, which are known to give consistent and statistically efficient estimates under very general conditions, are gradient-optimisation procedures, such as the maximum likelihood and prediction error methods in the SID toolbox; and iterative, relaxation procedures, such as the refined instrumental variable (RIV) method in the CAPTAIN toolbox.

On the surface, the choice between the two approaches may seem trivial. However, some recent studies have shown some serious shortcomings of the indirect route through DT models. Indeed, an extensive analysis aimed at comparing direct and indirect approaches has been discussed recently. The simulation model used in this analysis provides a very good test for CT and DT model estimation methods: it was first suggested by Rao and Garnier [36] (see also [17, 37, 38]); further investigations by Ljung [25] confirmed the results. This example illustrates some of the well-known difficulties that may appear in DT modelling under less standard conditions such as rapidly sampled data or relatively wide-band systems:

- relatively high sensitivity to the initialisation. DT model identification often requires computationally costly minimisation algorithms without even guaranteeing convergence (to the global optimum). In fact, in many cases, the initialisation procedure for the identification scheme is a key factor to obtain satisfactory estimation results compared to direct methods (see, *e.g.*, [50] for a recent reference);
- numerical issues in the case of fast sampling because the eigenvalues lie close to the unit circle in the complex domain, so that the model parameters are more poorly defined in statistical terms;
- *a priori* knowledge of the relative degree is not easy to accommodate;
- non-inherent data prefiltering in the gradient-based methods (adaptive prefiltering is an inherent part of the DT RIV method in CAPTAIN).

Further, the question of parameter transformation between a DT description and a CT representation is non-trivial. First, the zeros of the DT model are not as easily translatable to CT equivalents as the poles [2]; second, due to the discrete nature of the measurements they do not contain all the information about the CT signals. To describe the signals between the sampling instants some additional assumptions have to be made, for example, assuming that the excitation signal is constant between the sampling intervals (zero-order hold assumption). Violation of these assumptions may lead to severe estimation errors [40].

The advantages of direct CT identification approaches over the alternative DT identification methods can be summarised as follows:

- they directly provide differential-equation models whose parameters can be interpreted immediately in physically meaningful terms. As a result, they are of direct use to scientists and engineers who most often derive models in differential-equation terms based on natural laws and who are much less familiar with ‘black-box’ discrete-time models;
- they can estimate a fractional time-delay system;
- the estimated model is defined by a unique set of parameter values that are not dependent on the sampling interval T_s ;
- there is no need for conversion from discrete to continuous time, as required in the indirect route based on initial DT model estimation;
- the direct continuous-time methods can easily handle the case of mild irregularity in the sampled data;
- the *a priori* knowledge of the relative degree is easy to accommodate and therefore allows the identification of more parsimonious models than in discrete time;
- they also offer advantages when applied to systems with widely separated modes;
- they include inherent data filtering;
- they are well suited in the case of very rapid sampling. This is particularly interesting since modern data-acquisition equipment can provide nearly continuous-time measurements and, therefore, make it possible to acquire

data at very high sampling frequencies. Note that, as mentioned in [25], the use of prefiltering and decimation strategies may lead to better results in the case of DT modelling, but these may not be so obvious for practitioners. The CONTSID toolbox methods are free of these difficulties.

All these advantages will facilitate for the user the application of the general data-based modelling procedure. In the following, these advantages are illustrated with the help of a simulated benchmark system.

The SYSID'2006 Benchmark System

Here, the performance of the CONTSID toolbox techniques is illustrated by applying them to a benchmark example that was prepared for the 14th IFAC Symposium on System Identification (SYSID'06) in Newcastle, Australia⁶.

The intent of the benchmark was to set up a format in which rigorous comparisons between competing techniques for the identification of CT models from sampled data, including time- and frequency-domain approaches, could be undertaken. The goal was also to collect and analyse quantitative results in order to understand similarities and differences among the approaches and to highlight the strengths and weaknesses of each approach.

Two benchmark data sets were generated. Both were simulated continuous-time systems based closely on mechatronic systems. Data corresponding to these two benchmarks were sent to participants to apply their preferred technique.

Unfortunately, the associated Benchmark Session at SYSID was cancelled because referees felt that insufficient submitted papers were acceptable (only one of the papers submitted to the proposed benchmark session got even close to the correct model, demonstrating the difficulty of the benchmark exercise). This section presents the CONTSID toolbox results obtained on the benchmark data set 1, in which the additive measurement noise is a simple white additive noise. The second benchmark data set is more difficult since the white measurement noise is replaced by non-stationary noise (similar results were obtained using the CAPTAIN toolbox routines [50], where a modified example with coloured additive noise is considered).

The SYSID Benchmark data set 1 is obtained from

$$\begin{cases} x(t) = G_o(p)u(t), & \text{subject to zero initial conditions} \\ y(t_k) = x(t_k) + e(t_k) \end{cases} \quad (9.18)$$

where the measurement noise $e(t_k)$ is a zero-mean DT white Gaussian sequence.

The system is a linear, fourth-order, non-minimum phase system with complex poles where the Laplace transfer function is given by

⁶ The data can be downloaded from
<http://sysid2006benchmark.cran.uhp-nancy.fr/>.

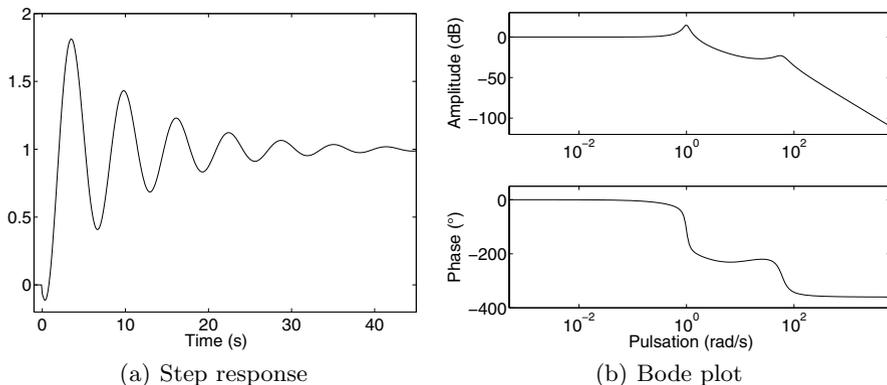


Fig. 9.8. Step response and Bode plot for the SYSID'2006 benchmark system

$$G_o(s) = \frac{K(-T_1s + 1)(T_2s + 1)e^{-\tau s}}{\left(\frac{s^2}{\omega_{n,1}^2} + \frac{2\zeta_1 s}{\omega_{n,1}} + 1\right)\left(\frac{s^2}{\omega_{n,2}^2} + \frac{2\zeta_2 s}{\omega_{n,2}} + 1\right)} \tag{9.19}$$

with $\tau = 0.035$ s, $K = 1$, $T_1 = 1/2$ s, $T_2 = 1/15$ s, $\omega_{n,1} = 1$ rad/s, $\zeta_1 = 0.1$, $\omega_{n,2} = 60$ rad/s, $\zeta_2 = 0.25$.

The system has one *fast* oscillatory mode with relative damping 0.25 and one *slow* oscillatory mode with relative damping 0.1 spread over one decade and a half. The system has a small time delay of 35 ms and is non-minimum phase, with a zero in the right half-plane.

The step response and the Bode plot of the system are shown in Figure 9.8. The settling time of the system is about 40 s.

The variance of the additive noise is adjusted to obtain a signal-to-noise ratio (see (9.16)) SNR= 8 dB.

The sampling period has been chosen as $T_s = 5$ ms (sampling frequency $\omega_s = 1256$ rad/s) that corresponds to about 20 times the bandwidth and is therefore higher than the usual rule of thumb given for discrete-time model identification⁷.

The input signal is a pseudo-random binary sequence of maximum length (± 1.0) and the complete data set consists of 6138 input/output samples. A section of the input/output data is plotted in Figure 9.9.

The model order selection procedure (see Section 9.3.2) was applied for different model structures. The best models sorted according to YIC are presented in Table 9.6. From this table, the correct model order structure $[\mathbf{n}_b \ \mathbf{n}_f \ \mathbf{n}_k] = [3 \ 4 \ 7]$ is quite clear cut. It has indeed the fourth most negative YIC=-11.33 with the highest associated $R_T^2 = 0.962$. The subsequent single-run SRIVC model estimation results are shown in Table 9.7, where it can

⁷ The usual rule of the thumb for discrete-time model identification is to choose the sampling frequency about ten times the bandwidth.

Table 9.6. Best SRIVC model orders for the SYSID'2006 benchmark system. N_{iter} denotes the number of iterations for the SRIVC algorithm to converge

n_b	n_f	n_k	YIC	R_T^2	N_{iter}
2	2	7	-12.65	0.937	2
2	2	8	-12.61	0.936	2
2	4	7	-12.24	0.936	4
3	4	7	-11.33	0.962**	4
3	4	8	-11.18	0.961	4
4	5	7	-10.89	0.962	4
2	4	8	-10.67	0.912	4
4	4	8	-9.95	0.962	4

be seen that the algorithm provides good parameter estimates with relatively small standard error (SE) estimates. Figure 9.10 shows that the step and frequency responses of the single run SRIVC estimated model are hardly distinguishable from those of the true model. It is interesting to mention here that indirect estimation using the discrete-time ARX, IV4 and PEM algorithms, followed by conversion to continuous time using the MATLAB® `d2cm` function, failed at this fast sampling rate because the algorithms did not converge on acceptable discrete-time models.

9.6 Successful Application Examples

In this section, identification results for three real-life processes selected from robotic, biological and electro-mechanical fields are summarised. They illustrate the use of the CONTSID toolbox methods to identify both continuous-time transfer function and state-space models directly from regularly and irregularly sampled data.

Additional successful experiences of the CONTSID toolbox methods with other real-life data processes [10, 13, 14, 28, 31] have already been reported including an industrial binary distillation column [10] but also in the case of biomedical systems [14] and environmental process data [51].

9.6.1 Complex Flexible Robot Arm

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 [22].

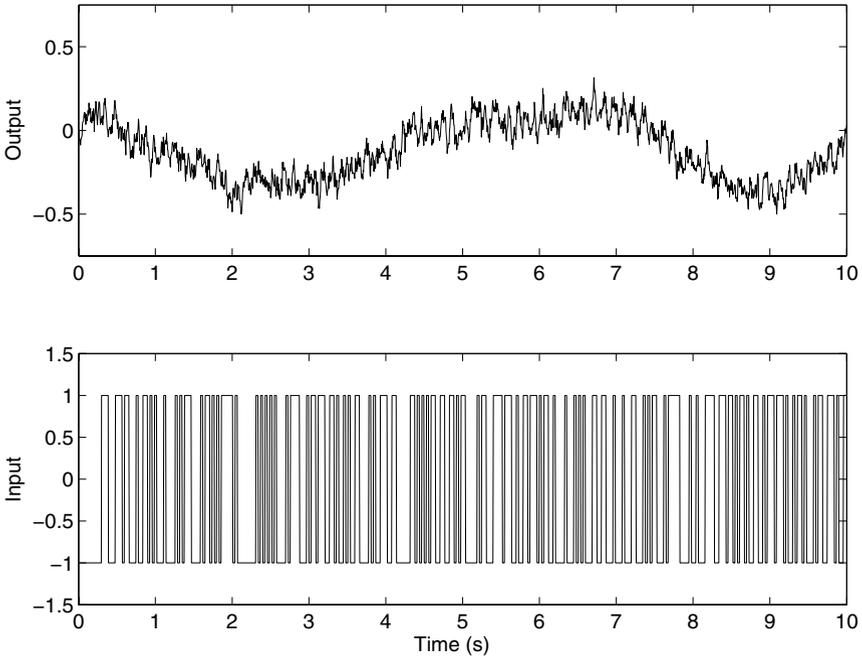


Fig. 9.9. A section of the SYSID'2006 benchmark data set 1

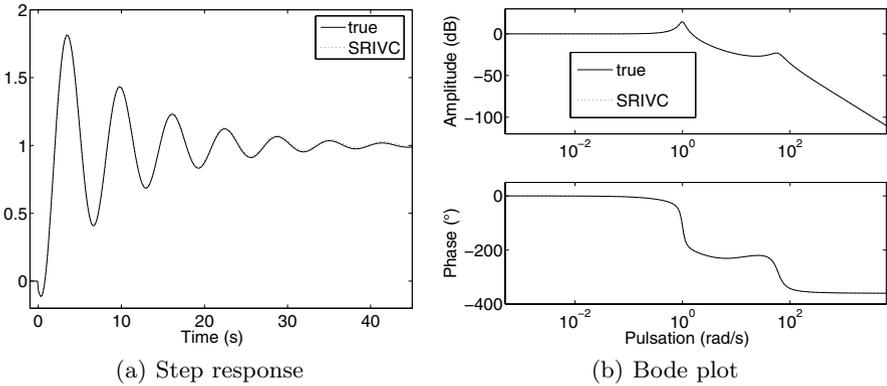


Fig. 9.10. Comparison of true and SRIVC model step and frequency responses (SYSID'2006 benchmark)

Table 9.7. SYSID'2006 benchmark system estimation results

Method	Value	\hat{b}_0	\hat{b}_1	\hat{b}_2	\hat{a}_1	\hat{a}_2	\hat{a}_3	\hat{a}_4
	True	-120	-1560	3600	30.2	3607	750	3600
SRIVC	$\hat{\theta}$	-114.6	-1493	3543	29.32	3524	732.9	3509
	SE	2.7	21.2	38.7	0.92	35.8	9.0	36.2

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. The data set over the 3rd period is displayed in Figure 9.11.

Model Order Determination

The empirical transfer function estimate (ETFE) obtained from the 3rd period data set is displayed in Figure 9.12. 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 for the 3rd period data set. The other data set periods were kept for model validation purposes⁸.

The 7 best models sorted according to YIC are given in Table 9.8. 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 $R_T^2 = 0.977$).

Identification Results

The process identification 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

$$\hat{G}(s) = \frac{20.87(s - 618.5)(s^2 - 1.698s + 710.6)(s^2 + 8.435s + 2.012e4)}{(s^2 + 1.033s + 2094)(s^2 + 0.9808s + 9905)(s^2 + 2.693s + 7.042e4)} \quad (9.20)$$

This estimated model is characterised by three, lightly damped dynamic modes, as defined in Table 9.9.

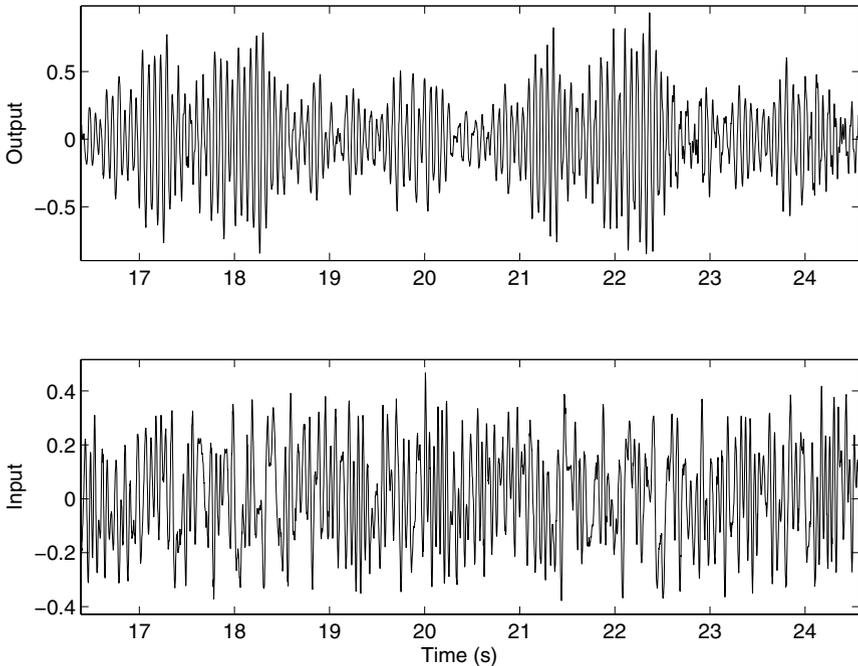


Fig. 9.11. The data set over the 3rd-period robot arm data set

Model Validation

Figure 9.13(a) 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 9.13(b).

9.6.2 Uptake Kinetics of a Photosensitising Agent into Cancer Cells

Process Description and Modelling Purpose

Figure 9.14(a) depicts the basic material used in *in vitro* experiments for studying the uptake kinetics of a photosensitising drug into living cancer cells. Cells are seeded in culture wells and are exposed at time $t_0 = 0$ to a photosensitising drug D . The purpose of this study is the estimation of the uptake yield

⁸ Similar identification results have been obtained from all of the other 9 period data sets.

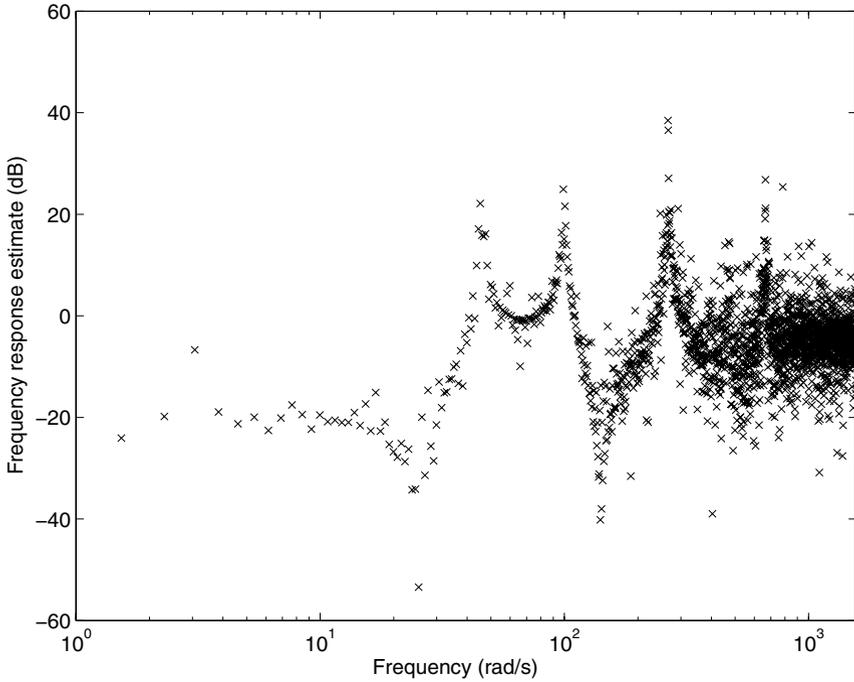


Fig. 9.12. Empirical transfer function estimate for the robot arm

(ρ) and the initial uptake rate (v_0). These biological parameters allow the biologists to discriminate the uptake characteristics of different photosensitisers and thus to choose the suitable photosensitising agent for the treatment of a given cancer cell line [3].

Experiment Design

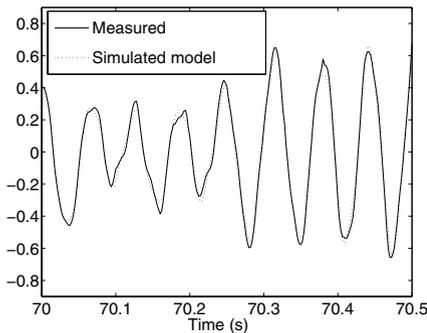
The input variable $u(t)$ of this biological process is a step signal that corresponds to the amount of drug injected into the well from time t_0 . The magnitude of the step is given by $u_0 = 5 \times 10^{-3} \mu\text{mol} \cdot \text{L}^{-1}$. $x(t)$ and $y(t)$ denote the extracellular quantity of photosensitising drug and the amount of drug absorbed by the cells, respectively. The process output is $y_m(t)$, the measurement of $y(t)$, given by a spectrofluorimeter at times $\{t_k\} = \{1, 2, 4, 6, 8, 14, 18, 24h\}$. Therefore, we are confronted with a model identification problem from non-uniformly sampled data. In this study, it is assumed that $y_m(t_k) = y(t_k) + e(t_k)$ where $e(t_k) \sim \mathcal{N}(0, \sigma_e^2)$, is the measurement noise. Two experiments have been carried out with two different protein concentrations $[Se] = 0\%$ et $[Se] = 9\%$ in the culture medium.

Table 9.8. Best SRIVC model orders for the robot arm data set

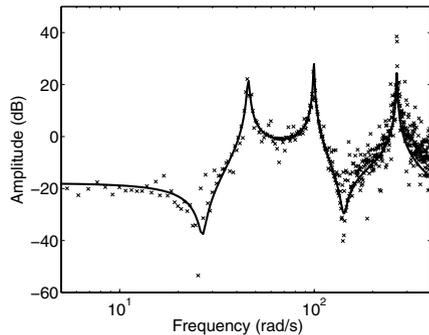
n_b	n_f	n_k	YIC	R_T^2
6	6	0	-9.19	0.977**
4	4	0	-8.56	0.940
7	6	0	-8.03	0.977
5	6	0	-7.41	0.976
5	4	0	-7.01	0.940
6	5	0	-5.56	0.966
4	5	0	-4.86	0.959
4	6	0	-3.49	0.950

Table 9.9. Eigenvalues and dynamic modes for the robot arm SRIVC model

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



(a) Cross-validation



(b) Amplitude Bode plot

Fig. 9.13. Cross-validation results on a short section of the 8th-period data set and comparison of ETFE ('x') and SRIVC model (solid line) frequency responses for the flexible robot arm

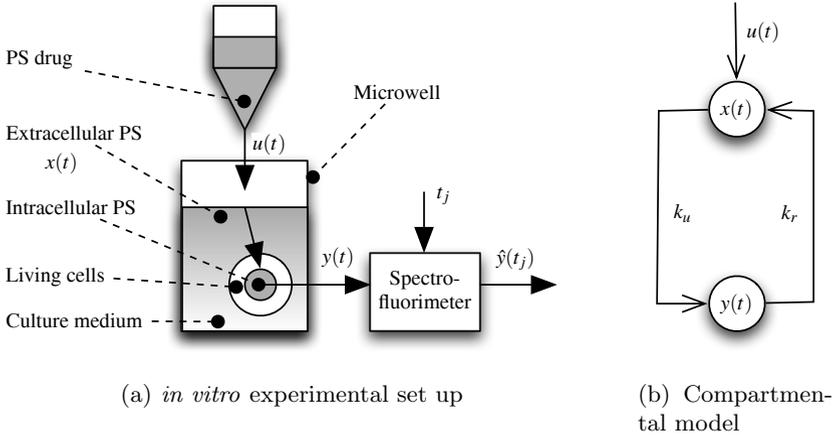


Fig. 9.14. Biological process

Model Structure Selection

The *in vitro* uptake of the photosensitising agent into cancer cells can be described by a linear model with two compartments, as shown in Figure 9.14(b). The two compartments are associated with the extracellular and intracellular volumes, respectively. Parameters k_u and k_r are the uptake and release rates respectively. Differential equations of this compartmental model are defined as follows

$$\frac{dx(t)}{dt} = k_r y(t) - k_u x(t) + \frac{du(t)}{dt} \tag{9.21}$$

$$\frac{dy(t)}{dt} = k_u x(t) - k_r y(t) \tag{9.22}$$

with $x(0) = y(0) = 0$. Substitution of $x(t)$ from (9.21) into (9.22), yields

$$\frac{1}{k_u + k_r} \frac{dy(t)}{dt} + y(t) = \frac{k_u}{k_u + k_r} u(t) \tag{9.23}$$

Accordingly, the equivalent first-order model used in the parameter estimation step is

$$G_{[Se]}(p) = \frac{b_0}{p + f_0} \tag{9.24}$$

with $b_0 = k_u$ and $f_0 = k_u + k_r$. The uptake yield rate and the initial uptake rate of the photosensitiser uptake process are given by $\rho = b_0/f_0$ and $v_0 = b_0$.

Identification Results

The process identification is performed with the `srivc` algorithm on two *in vitro* data sets. The estimated transfer function models for the two protein concentrations take the form

$$\hat{G}_{0\%}(p) = \frac{162.7(\pm 21.45)}{p + 0.333(\pm 0.056)}; \quad \hat{G}_{9\%}(p) = \frac{41.1(\pm 7.18)}{p + 0.535(\pm 0.12)} \quad (9.25)$$

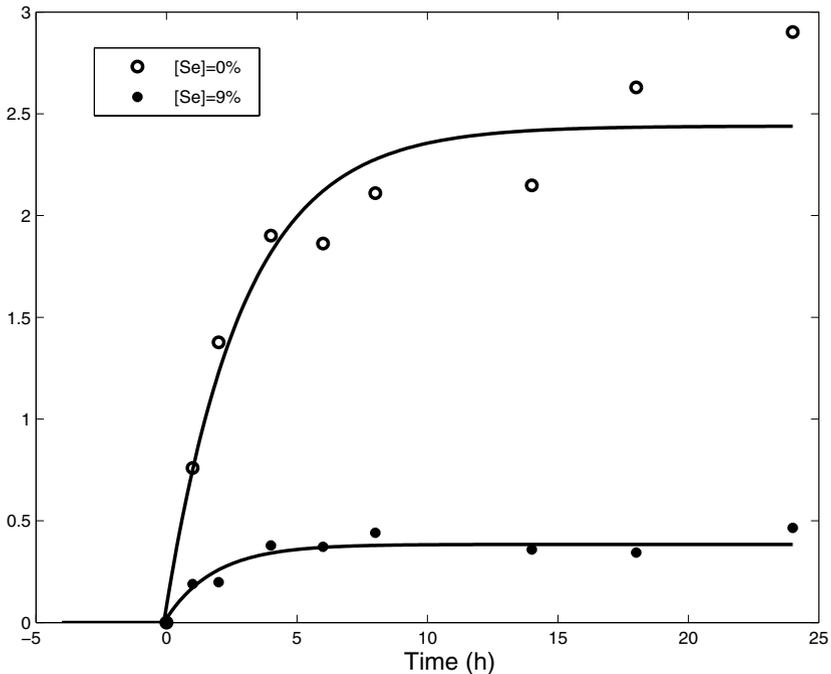


Fig. 9.15. Measured (‘•’ and ‘◦’) and simulated uptake responses (solid lines)

Model Validation

Figure 9.15 shows the two estimation data sets and the simulated uptake responses obtained from $\hat{G}_{0\%}$ and $\hat{G}_{9\%}$. In experimental biology, identical experiments often produce different outcomes. This variability of the measurements is mainly due to the high sensitivity of living cells to external disturbances. Accordingly, cross-validation tests are usually not applicable in such a biological context. Here, however, the identified model has been validated by biologists [6].

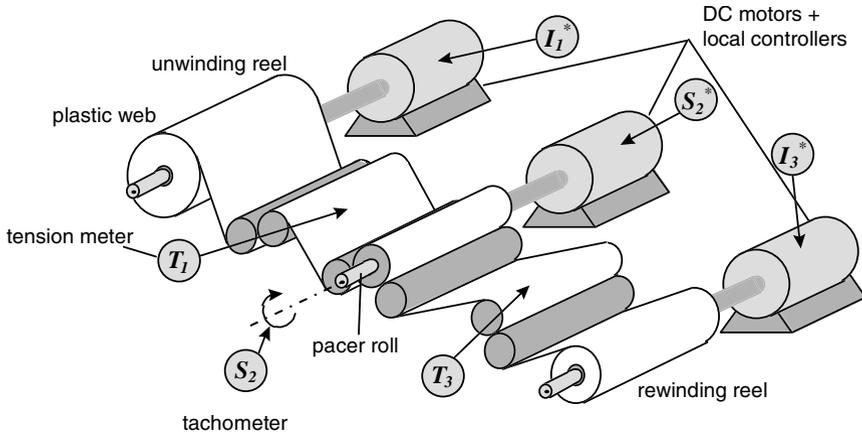


Fig. 9.16. Winding process

9.6.3 Multi-variable Winding Process

Pilot description

The present section turns to a multiple-input, multiple-output system based on a winding pilot plant [5]. Winding systems are, in general, continuous, non-linear processes. They are encountered in a wide variety of industrial plants such as rolling mills in the steel industry, plants involving web conveyance including coating, papermaking and polymer film extrusion processes. The main role of a winding process is to control the web conveyance in order to avoid the effects of friction and sliding, as well as the problems of material distortion that can also damage the quality of the final product.

As illustrated in Figure 9.16, the main part of this MIMO pilot plant is a winding process composed of a plastic web and three reels. Each reel is coupled with a direct-current motor via gear reduction. The angular speed of each reel (S_1 , S_2 , S_3) is measured by a tachometer, while the tensions between the reels (T_1 , T_3) are measured by tension meters. At a second level, each motor is driven by a local controller. Two PI control loops adjust the motor currents (I_1) and (I_3) and a double PI control loop drives the angular speed (S_2). The setpoints of the local controllers (I_1^* , S_2^* , I_3^*) constitute the manipulated inputs of the winding system $u(t) = [I_1^*(t) S_2^*(t) I_3^*(t)]^T$. Essentially, driving a winding process comes down to controlling the web linear velocity and the web tensions (T_1) and (T_3) around a given operating point. Consequently, the output variables of the winding system are $y(t) = [T_1(t) T_3(t) S_2(t)]^T$. The process is described in more detail in [4]. The relevant MATLAB® files in the CONTSID toolbox are *idcdemo7.m* and *winding.mat*.

Experiment Design

Discrete-time internal binary sequences were used as excitation signals. The sampling period is set to 10 ms. The mean and linear trend of the signals were removed and the resulting input/output signals are shown in Figures 9.18 and 9.17.

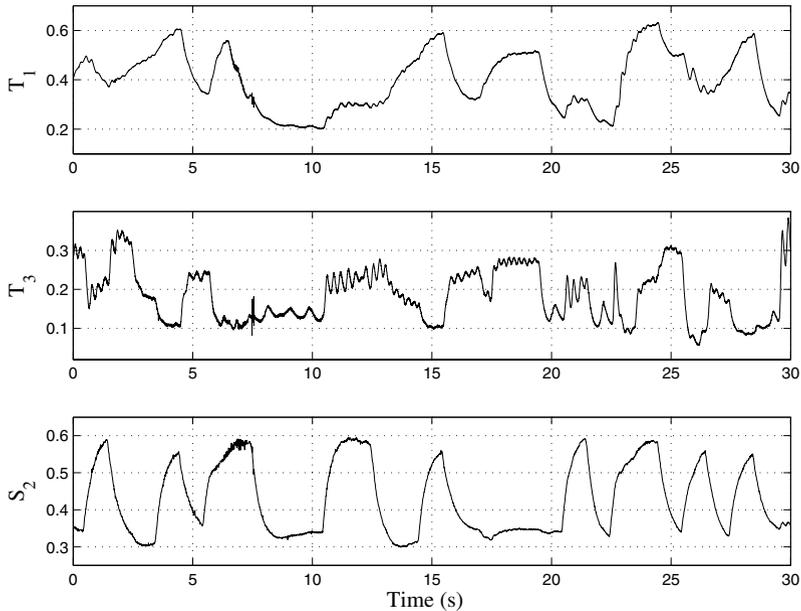


Fig. 9.17. Output signals for the winding process

Model Structure Selection

The system order $n = 3$ has been chosen by analysing the evolution of the mean square error between the process and model outputs with respect to n . No significant decrease has been observed for $n > 3$. Note, however, that the algorithm makes it possible to estimate the system order along with the model parameters if it is not known *a priori*.

Identification Results

The process identification is performed with the 4SID-based GPMF algorithm `sidgpmf`. The identification result is given as a CT state-space model that takes the form

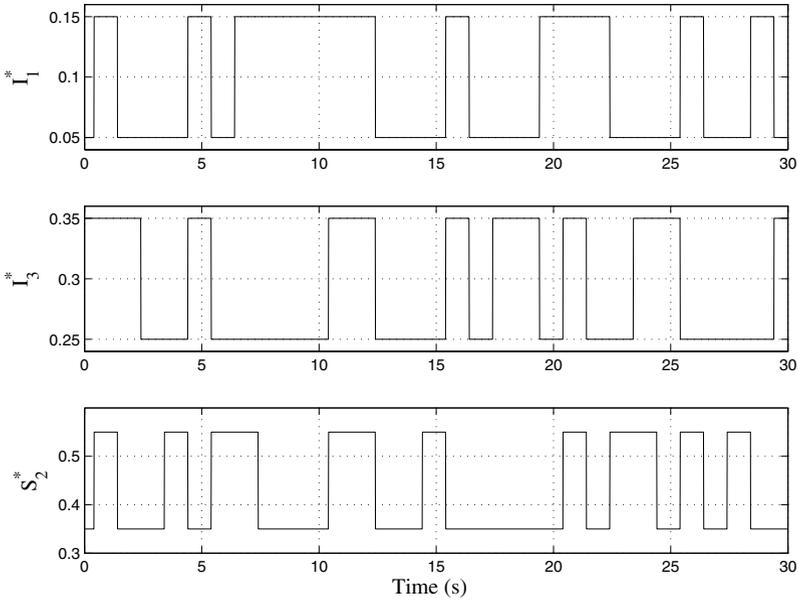


Fig. 9.18. Input signals for the winding process

$$\begin{cases} \dot{x}(t_k) = Ax(t_k) + Bu(t_k) \\ y(t_k) = Cx(t_k) + Du(t_k) + \xi(t_k) \end{cases} \quad (9.26)$$

with

$$\left(\begin{array}{c|c} \begin{matrix} -1.6414 & -0.9874 & -0.4773 \\ -0.1261 & -2.7725 & -1.3205 \\ 0.4517 & 2.1746 & -4.2674 \end{matrix} & \begin{matrix} 4.4994 & -3.1047 & -4.0889 \\ 2.0652 & -3.3796 & -9.0513 \\ 11.7889 & 9.6855 & -15.4186 \end{matrix} \\ \hline \begin{matrix} -1.1073 & 0.4345 & -0.0536 \\ 0.1442 & -0.1717 & -0.2537 \\ -0.2047 & -0.4425 & 0.1120 \end{matrix} & \mathbf{0} \end{array} \right) \quad (9.27)$$

Model Validation

Cross-validation results are plotted in Figure 9.19 where it may be observed that there is a very good agreement with quite high values for the coefficient of determination.

9.7 Conclusions

This chapter has outlined the main features of the MATLAB® CONTSID toolbox and illustrated its potential in practical applications. The toolbox,

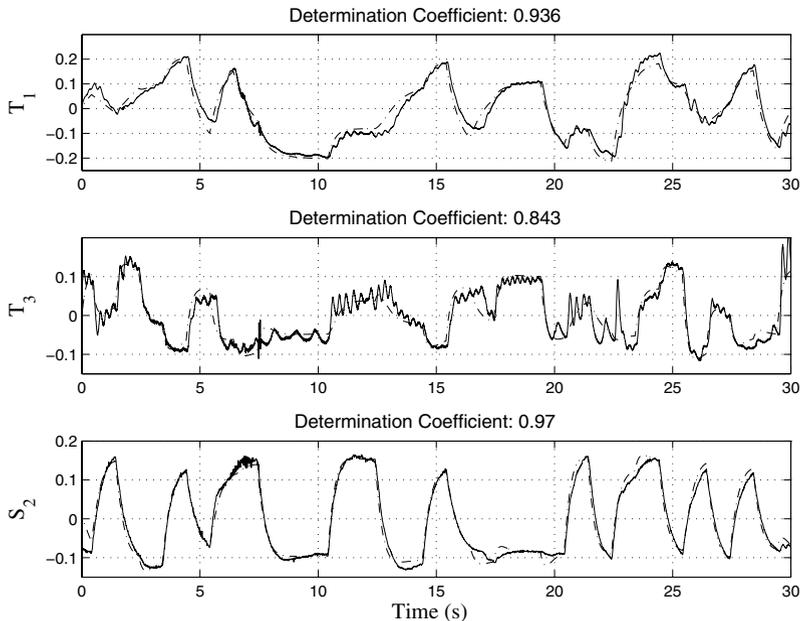


Fig. 9.19. Cross-validation results for the winding process. Measured (solid line) and model (dashdot line) outputs.

which provides access to most of the time-domain continuous-time model identification techniques that allow for the identification of continuous-time models from discrete-time data, is in continual development. Planned new releases will incorporate routines to solve errors-in-variables and closed-loop identification problems, as well as non-linear continuous-time model identification techniques.

Acknowledgements

The authors wish to thank Eric Huselstein and Hamza Zbali for the developments of the CONTSID toolbox and the CONTSID graphical user interface, respectively.

They are also grateful to Professor Istvan Kollar from the Budapest University of Technology and Economics, Hungary who kindly provided us with the robot-arm data used in Section 9.6.1.

The authors are very grateful to Professor Peter Young for his help in reading and commenting on this chapter during its preparation. Of course, the authors remain responsible for any errors or omissions in this final version.

References

1. K.J. Åström. *Introduction to Stochastic Control Theory*. Academic Press, New York, 1970.
2. K.J. Åström, P. Hagander, and J. Sternby. Zeros of sampled systems. *Automatica*, 20(1):31–38, 1984.
3. M. Barberi-Heyob, P.-O. Védrine, J.-L. Merlin, R. Millon, J. Abecassis, M.-F. Poupon, and F. Guillemin. Wild-type p53 gene transfer into mutated p53 HT29 cells improves sensitivity to photodynamic therapy via induction of apoptosis. *International Journal of Oncology*, 24:951–958, 2004.
4. T. Bastogne, H. Garnier, and P. Sibille. A PMF-based subspace method for continuous-time model identification. Application to a multivariable winding process. *International Journal of Control*, 74(2):118–132, 2001.
5. T. Bastogne, H. Noura, P. Sibille, and A. Richard. Multivariable identification of a winding process by subspace methods for a tension control. *Control Engineering Practice*, 6(9):1077–1088, 1998.
6. T. Bastogne, L. Tirand, M. Barberi-Heyob, and A. Richard. System identification of photosensitiser uptake kinetics in photodynamic therapy. *6th IFAC Symposium on Modelling and Control in Biomedical System*, Reims, France, September 2006.
7. Y.C. Chao, C.L. Chen, and H.P. Huang. Recursive parameter estimation of transfer function matrix models via Simpson’s integrating rules. *International Journal of Systems Science*, 18(5):901–911, 1987.
8. C.F. Chen and C.H. Hsiao. Time-domain synthesis via Walsh functions. *IEE Proceedings*, 122(5):565–570, 1975.
9. H. Dai and N.K. Sinha. *Use of numerical integration methods*, in N.K. Sinha and G.P. Rao (eds), *Identification of Continuous-Time Systems. Methodology and Computer Implementation*, pages 79–121, Kluwers Academic Publishers: Dordrecht, Holland, 1991.
10. H. Garnier. Continuous-time model identification of real-life processes with the CONTSID toolbox. *15th IFAC World Congress*, Barcelona, Spain, July 2002.
11. H. Garnier, M. Gilson, and O. Cervellin. Latest developments for the MATLAB® CONTSID toolbox. *14th IFAC Symposium on System Identification*, Newcastle, Australia, pages 714–719, March 2006.
12. H. Garnier, M. Gilson, and E. Huselstein. Developments for the MATLAB® CONTSID toolbox. *13th IFAC Symposium on System Identification*, Rotterdam, The Netherlands, pages 1007–1012, August 2003.
13. H. Garnier, M. Gilson, P.C. Young, and E. Huselstein. An optimal IV technique for identifying continuous-time transfer function model of multiple input systems. *Control Engineering Practice*, 46(15):471–486, April 2007.
14. L. Cu villon, E. Laroche, H. Garnier, J. Gangloff, and M. de Mathelin. Continuous-time model identification of robot flexibilities for fast visual servoing. *14th IFAC Symposium on System Identification*, Newcastle, Australia, pages 1264–1269, March 2006.
15. H. Garnier and M. Mensler. CONTSID: a continuous-time system identification toolbox for Matlab. *5th European Control Conference*, Karlsruhe, Germany, September 1999.
16. H. Garnier and M. Mensler. The CONTSID toolbox: a MATLAB® toolbox for CONTinuous-Time System IDENTification. *12th IFAC Symposium on System Identification*, Santa Barbara, USA, June 2000.

17. H. Garnier, M. Mensler, and A. Richard. Continuous-time model identification from sampled data. Implementation issues and performance evaluation. *International Journal of Control*, 76(13):1337–1357, 2003.
18. H. Garnier, P. Sibille, and T. Bastogne. A bias-free least squares parameter estimator for continuous-time state-space models. *36th IEEE Conference on Decision and Control*, San Diego, USA, Vol. 2, pages 1860–1865, December 1997.
19. H. Garnier, P. Sibille, H.L. NGuyen, and T. Spott. A bias-compensating least-squares method for continuous-time system identification via Poisson moment functionals. *10th IFAC Symposium on System Identification*, Copenhagen, Denmark, pages 3675–3680, July 1994.
20. H. Garnier, P. Sibille, and A. Richard. Continuous-time canonical state-space model identification via Poisson moment functionals. *34th IEEE Conference on Decision and Control*, New Orleans, USA, Vol. 2, pages 3004–3009, December 1995.
21. E. Huselstein and H. Garnier. An approach to continuous-time model identification from non-uniformly sampled data. *41st IEEE Conference on Decision and Control*, Las Vegas, USA, December 2002.
22. I. Kollar. *Frequency Domain System Identification Toolbox Users's Guide*. The Mathworks, Inc., Mass., 1994.
23. E.K. Larsson and T. Söderström. Identification of continuous-time AR processes from unevenly sampled data. *Automatica*, 38(4):709–718, 2002.
24. L. Ljung. *System Identification. Theory for the User*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 1999.
25. L. Ljung. Initialisation aspects for subspace and output-error identification methods. *European Control Conference*, Cambridge, UK, September 2003.
26. L. Ljung. SID: System identification toolbox for MATLAB®. <http://www.mathworks.com/access/helpdesk/help/toolbox/ident/ident.shtml>, 2006.
27. K. Mahata and H. Garnier. Identification of continuous-time Box-Jenkins models with arbitrary time delay. *Submitted to the 46th Conference on Decision and Control*, New Orleans, USA, December 2007.
28. M. Mensler, H. Garnier, and E. Huselstein. Experimental comparison of continuous-time model identification methods on a thermal process. *12th IFAC Symposium on System Identification*, Santa Barbara, USA, June 2000.
29. M. Mensler, K. Wada. Subspace method for continuous-time system identification. *32nd ISCIE International Symposium on Stochastic Systems Theory and Its Applications*, Tottori, Japan, November 2000.
30. M. Mensler, H. Garnier, A. Richard, and P. Sibille. Comparison of sixteen continuous-time system identification methods with the CONTSID toolbox. *5th European Control Conference*, Karlsruhe, Germany, September 1999.
31. M. Mensler, S. Joe, and T. Kawabe. Identification of a toroidal continuously variable transmission using continuous-time system identification methods. *Control Engineering Practice*, 14(1):45–58, January 2006.
32. B.M. Mohan and K.B. Datta. Analysis of time-delay systems via shifted Chebyshev polynomials of the first and second kinds. *International Journal of Systems Science*, 19(9):1843–1851, 1988.
33. P.N. Paraskevopoulos. System analysis and synthesis via orthogonal polynomial series and Fourier series. *Mathematics and Computers in Simulation*, 27:453–469, 1985.

34. A.E. Pearson and Y. Shen. Weighted least squares / MFT algorithms for linear differential system identification. *32nd IEEE Conference on Decision and Control*, San Antonio, USA, pages 2032–2037, 1993.
35. M. Djamai, E. Tohme, R. Ouvrard, and S. Bachir. Continuous-time model identification using reinitialized partial moments. Application to power amplifier modeling. *14th IFAC Symposium on System Identification*, Newcastle, Australia, March 2006.
36. G.P. Rao and H. Garnier. Numerical illustrations of the relevance of direct continuous-time model identification. *15th IFAC World Congress*, Barcelona, Spain, July 2002.
37. G.P. Rao and H. Garnier. Identification of continuous-time models: direct or indirect? *Invited semi-plenary paper for the XV International Conference on Systems Science*, Wroclaw, Poland, September 2004.
38. G.P. Rao and H. Unbehauen, Identification of continuous-time systems, *IEE Proceedings - Control Theory and Applications*, 153(2):185-220, March 2006.
39. S. Sagara and Z.Y. Zhao. Numerical integration approach to on-line identification of continuous-time systems. *Automatica*, 26(1):63–74, 1990.
40. J. Schoukens, R. Pintelon, and H. Van Hamme. Identification of linear dynamic systems using piecewise constant excitations: use, misuse and alternatives. *Automatica*, 30(7):1953–1169, 1994.
41. T. Söderström and P. Stoica. *System Identification*. Series in Systems and Control Engineering. Prentice Hall, Englewood Cliffs, 1989.
42. H. Unbehauen and G.P. Rao. *Identification of Continuous Systems*. Systems and control series. North-Holland, Amsterdam, 1987.
43. H. Unbehauen and G.P. Rao. Identification of continuous-time systems: a tutorial. *11th IFAC Symposium on System Identification*, Kitakyushu, Japan, Vol. 3, pages 1023–1049, July 1997.
44. P. Van Overschee and B. De Moor. *Subspace Identification for Linear Systems - Theory, Implementation, Applications*. Kluwer Academic Publishers, Boston, USA, 1996.
45. P.E. Wellstead. An instrumental product moment test for model order estimation. *Automatica*, 14:89–91, 1978.
46. P.C. Young. *Recursive Estimation and Time-series Analysis*. Springer-Verlag, Berlin, 1984.
47. P.C. Young. *Recursive estimation, forecasting and adaptive control*. in C.T. Leondes (ed), *Control and Dynamic Systems: Advances in Theory and Applications*, pages 119–166, Vol 31, Academic Press, 1989.
48. P.C. Young. Data-based mechanistic modeling of engineering systems. *Journal of Vibration and Control*, 4:5–28, 1998.
49. P.C. Young. The data-based mechanistic approach to the modelling, forecasting and control of environmental systems. *Annual Reviews in Control*, 30:169–182, 2006.
50. P.C. Young. The Refined Instrumental Variable (RIV) method: unified estimation of discrete and continuous-time transfer function models. *Journal Européen des Systèmes Automatisés*, in press, 2008.
51. P.C. Young and H. Garnier. Identification and estimation of continuous-time data-based mechanistic (DBM) models for environmental systems. *Environmental Modelling and Software*, 21(8):1055–1072, August 2006.

52. P.C. Young and A.J. Jakeman. Refined instrumental variable methods of time-series analysis: Part III, extensions. *International Journal of Control*, 31:741–764, 1980.
53. P.C. Young, A.J. Jakeman, and R. McMurtries. An instrumental variable method for model order identification. *Automatica*, 16:281–296, 1980.