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

H. Garnier
Centre de Recherche en Automatique de Nancy (CRAN), Nancy-University, CNRS, BP 70239, 54506 Vandoeuvre-les-Nancy Cedex, France

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

Keywords: continuous-time model, discrete-time data, education, Matlab toolbox, software tools, system identification

1. INTRODUCTION

The identification of continuous-time (CT) models is a problem of considerable importance that has applications in virtually all disciplines of science. Early research in system identification focussed on identification of CT models from CT data. Subsequently, however, rapid developments in digital data acquisition and computers have resulted in attention being shifted to the identification of discrete-time (DT) models from sampled data, leading to a complete dominance of these methods over continuous-time model identification techniques. Much less attention has been therefore devoted to CT modelling from DT data and many practitioners appear unaware that such alternative methods not only exist but may be better suited to their modelling problems (H. Garnier and L. Wang (Eds.) (2008)).

The relevance of direct continuous-time model identification methods has been recently illustrated with extensive numerical simulation (Ljung (2003); Rao and Garnier (2004)). A toolbox for Matlab is now available: the CONtinuous-Time System IDentification (CONTSID) toolbox which supports continuous-time transfer function or state-space model identification directly from time-domain sampled data, without requiring the determination of a discrete-time model (Garnier et al. (2006); H. Garnier and L. Wang (Eds.) (2008)).

While undergraduate and graduate courses on DT model identification (mainly based on Ljung’s book (Ljung (1999)) are taught in many universities, there is a lack of courses in data-based CT modelling. For some years now, we have introduced a course at Nancy-University which aims at introducing CT model identification to engineering students. CT models are very appealing to engineers since they can be directly and easily interpreted in a physical manner. Some of the topics that we have been able to introduce in the curriculum include:

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

This paper discusses this experience to introduce data-based continuous-time model identification to engineering students. It is organized as follows. Section 2 outlines the main steps of the procedure for direct continuous-time model identification. An overview of the course organisation is given in Section 3. The main parametric estimation methods are briefly presented in Section 4. The main features of the CONTSID toolbox and its graphical user interface (GUI) are then presented in Section 5. Finally, typical results obtained by the students on a complex flexible robot arm are described in Section 6.

2. PROCEDURE FOR DATA-BASED CT MODELLING

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

- the time-domain sampled input-output data;
- a set of candidate models (the model structure);
- a criterion to select a particular model in the set, based on the information in the data (the identification method).
The identification procedure consists then in repeatedly selecting a model structure, computing the best model in the chosen structure, and evaluating the identified model. The iterative procedure involves the following steps:

1. Design an experiment and collect input-output data from the process to be identified.
2. Examine the data. Remove trends and outliers, and select useful portions of the original data.
3. Select and define a model structure (a set of candidate system descriptions) within which a model is to be estimated.
4. Compute the best model in the model structure according to the input-output data and a given criterion of fit.
5. Examine the obtained model properties.

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

3. THE COURSE FOR ENGINEERING STUDENTS

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

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

The 18-hour course is typically laid out over 9 weeks, with one two-hour lecture per week for the first five weeks, followed by four two-hour computer sessions. The textbook used is (H. Garnier and L. Wang (Eds.) (2008)) and the course outline uses Chapters 1, 4 and 9 of this textbook.

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

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

4. PARAMETRIC ESTIMATION METHODS

Consider a linear, single-input, single-output, CT system whose input \( u(t) \) and output \( y(t) \) are related by a constant coefficient differential equation of order \( n \)

\[
x^{(n)}(t) + a_1x^{(n-1)}(t) + \ldots + a_nx^{(0)}(t) = b_0u^{(m)}(t) + b_1u^{(m-1)}(t) + \ldots + b_mu^{(0)}(t)
\]

where \( x^{(i)}(t) \) denotes the \( i \)th time-derivative of the continuous-time signal \( x(t) \). Equation (1) can also be written in the transfer function (TF) form:

\[
x(t) = \frac{B(p)}{A(p)} u(t),
\]

with

\[
B(p) = b_0p^m + b_1p^{m-1} + \cdots + b_m,
\]

\[
A(p) = p^n + a_1p^{n-1} + \cdots + a_n,
\]

where \( p \) is the differential operator, i.e. \( px(t) = \frac{dx(t)}{dt} \). It is assumed that the input signal \( \{u(t), t_1 < t < t_N\} \) is applied to the system and that the output \( x(t) \) is sampled at discrete times \( t_1, \ldots, t_N \), not necessarily uniformly spaced. The sampled signals are denoted by \( \{u(t_k); x(t_k)\}_{k=1}^{N} \).

The measured output \( y(t_k) \) is assumed to be corrupted by an additive measurement noise \( v(t_k) \)

\[
y(t_k) = x(t_k) + v(t_k).
\]

The identification problem aims at estimating the parameters of the differential equation model directly from \( N \) sampled measurements of the input and output \( Z^N = \{u(t_k); y(t_k)\}_{k=1}^{N} \).

4.1 The traditional SVF method

Consider first the TF model (1) in the simple noise-free case. This latter can then be written in the linear equation form,

\[
A(p)x(t) = B(p)u(t).
\]

Assume now that a filter with operator model \( F(p) \) is applied to both sides of (3). Then, ignoring transient initial condition effects

\[
A(p)F(p)x(t) = B(p)F(p)u(t).
\]

The minimum-order SVF filter is typically chosen to have the following form\(^2\)

\[
F(p) = \frac{1}{E(p)} = \frac{1}{(p + \lambda)^n}
\]

where \( \lambda \) is the breakpoint frequency.

Let \( F_i(p) \) for \( i = 0, 1, \ldots, n \) be a set of filters defined as

\[
F_i(p) = \frac{p^i}{E(p)} = \frac{p^i}{(p + \lambda)^n}
\]

and \( f_i(t) \) be their corresponding functions in the time domain. By using the filters defined in (6), equation (4) can then be rewritten, in expanded form, as

\[
\{F_0(p) + a_1F_{n-1}(p) + \ldots + a_nF_0(p)\}x(t) = \{b_0F_{m}(p) + \cdots + b_mF_0(p)\}u(t).
\]

---

1 A time-delay on the system input is not considered for simplicity here but is easy to accommodate.
2 The filter d.c. gain can be made unity if this is thought desirable.
In terms of time-domain signals, (7) can be written as
\[ x(n) f(t) + a_1 x(n-1) f(t) + \ldots + a_n x(0) f(t) = b_0 u(m) f(t) \]
where
\[ x_f^{(i)}(t) = f_i(t) * x(t) \]
\[ u_f^{(i)}(t) = f_i(t) * u(t), \]
and * denotes the convolution operator. The filter outputs \( x_f^{(i)}(t) \) and \( u_f^{(i)}(t) \) provide prefiltered time-derivatives of the inputs and outputs in the bandwidth of interest, which may then be exploited for model parameter estimation.

At time-instant \( t = t_k \), considering now the situation where there is additive noise on the output measurement, equation (8) can be rewritten in standard linear regression form as
\[ y_f^{(n)}(t_k) = \phi_f^T(t_k) \theta + \varepsilon(t_k), \]
with
\[ \phi_f^T(t_k) = \left[ -y_f^{(n-1)}(t_k) \ldots - y_f^{(0)}(t_k) u_f^{(m)}(t_k) \ldots u_f^{(0)}(t_k) \right] \]
\[ \theta = [a_1 \ldots a_n b_0 \ldots b_m]^T. \]

Now, from \( N \) available samples of the input and output signals observed at discrete times \( t_1, \ldots, t_N \), not necessarily uniformly spaced, the linear least-squares (LS)-based parameter estimates are given by
\[ \hat{\theta}_N^{LS} = \left[ \sum_{k=1}^{N} \phi_f(t_k) \phi_f^T(t_k) \right]^{-1} \sum_{k=1}^{N} \phi_f(t_k) y_f^{(n)}(t_k). \]

Unfortunately, it is well-known that, except in the special case where \( \varepsilon(t_k) \) is white, LS estimation, such as (12), although simple, is unsatisfactory. For instance, even if the additive noise \( v(t_k) \) is white, the resultant parameter estimates are asymptotically biased and inconsistent. One of the simplest solutions to this asymptotic bias problem is to use IV methods because they do not require a priori knowledge of the noise statistics.

Let us consider the most common IV method, where the instrumental variable is generated by an ‘auxiliary model’ which, as we see later, may be iteratively adapted. In the simplest, non-iterative case, the IV vector is then given by,
\[ \phi_f^T(t_k) = \left[ -\hat{x}_f^{(n-1)}(t_k) \ldots - \hat{x}_f^{(0)}(t_k) u_f^{(m)}(t_k) \ldots u_f^{(0)}(t_k) \right] \]
where \( \hat{x}_f(t_k) = F(p) \hat{x}(t_k) \)
and \( \hat{x}(t_k) \) is the estimated noise-free output calculated from,
\[ \hat{x}(t_k) = \frac{B(p, \hat{\theta}_N^{LS})}{A(p, \hat{\theta}_N^{LS})} u(t_k). \]

The IV-based parameter estimates are then given by
\[ \hat{\theta}_N^{IV} = \left[ \sum_{k=1}^{N} \hat{\phi}_f(t_k) \phi_f^T(t_k) \right]^{-1} \sum_{k=1}^{N} \hat{\phi}_f(t_k) y_f^{(n)}(t_k), \]
provided that the inverse exists. Despite its simplicity, this IV technique is one of six methods that have proven successful in extensive Monte Carlo simulation studies (Garnier et al. (2003)).

### 4.2 Optimal IV method
Disregarding the noise properties, as in the deterministic approaches outlined in the previous section, leads to statistical inefficiency (increased variance of the estimates) and does not provide information on the estimated variance-covariance properties of the parameter estimates. The key idea of stochastic identification is to assume that the disturbing noise \( v(t) \) can be written, at the sampling instants, as filtered, discrete-time, white noise \( v(t_k) \). This avoids the mathematically difficult problem of treating CT random processes.

One particularly successful stochastic identification method is the iterative Simplified Refined Instrumental Variable method for Continuous-time model Identification (SRIVC: see e.g. Young et al. (2008)). This approach involves a method of adaptive prefiltering based on a quasi-optimal statistical solution to the problem when the additive noise \( v(t_k) \) is white. SRIVC is a logical extension of the more heuristically defined SVF.

Following the usual Prediction Error Minimization (PEM) approach (Maximum Likelihood (ML) in the present situation because of the Gaussian assumptions), a suitable error function \( \varepsilon(t) \) is given by the output error (OE),
\[ \varepsilon(t) = y(t) - B(p) A(p) u(t) \]
Minimization of a least squares criterion function in \( \varepsilon(t) \), measured at the sampling instants provides the basis for the output error estimation methods. However \( \varepsilon(t) \) can also be rewritten as
\[ \varepsilon(t) = \frac{1}{A(p)} (A(p) y(t) - B(p) u(t)). \]

Since the operators commute in this linear case, the filter \( F(p) = 1/A(p) \) can be taken inside the brackets to yield
\[ \varepsilon(t) = A(p) y_f(t) - B(p) u_f(t) \]
or,
\[ \varepsilon(t) = y_f^{(n)}(t) + a_n y_f^{(n-1)}(t) + \ldots + a_0 y_f^{(0)}(t) - b_m u_f^{(m)}(t) - \ldots - b_0 u_f^{(0)}(t) \]
where
\[ \left( y_f^{(i)}(t) = f_i(t) * y(t), \right. \] with \( i = 0, \ldots, n \)
\[ u_f^{(i)}(t) = f_i(t) * u(t), \] with \( i = 0, \ldots, m \),
and the set of filters now takes the form
\[ F_i(p) = \frac{p^i}{A(p)}. \]

The associated estimation model can be written at time-instant \( t = t_k \) in the form:
\[ y_f^{(n)}(t_k) = \phi_f^T(t_k) \theta + \varepsilon(t_k) \]
where \( \phi_f^T(t_k) \) and \( \theta \) are defined as in (10) and (11) respectively with \( F_i(p) \) defined in (20). Thus, provided we assume that \( A(p) \) is known, the estimation model (21) forms a basis for the definition of a likelihood function and ML estimation.

There are two problems with this formulation. The obvious one is, of course, that \( A(p) \) is not known a priori. The
less obvious one is that, in practical applications, we cannot assume that the noise $v(t_k)$ will have the nice white noise properties assumed above: it is likely that the noise will be a coloured noise process, say $\xi(t_k)$. Both of these problems can be solved by employing a ‘relaxation’ optimization procedure that adaptively adjusts an initial estimate $A(p, \theta^0)$ of $A(p, \theta^i)$ iteratively until it converges on an optimal estimate of $A(p)$. And the coloured noise problem is solved conveniently by exploiting IV estimation within this iterative optimization algorithm.

Of course, if the noise $v(t_k) = \xi(t_k)$ is coloured, then the method is not quasi-optimal in statistical terms. However, experience has shown that it is robust and normally yields estimates with reasonable statistical efficiency (i.e. low but not minimum variance). However, albeit at the cost of increased complexity, it is possible to use a hybrid approach in the coloured noise case, where the noise modelling, as well as the noise-derived parts of the prefiltering, are carried out in discrete-time terms (Young et al. (2008)).

5. SOFTWARE ASPECTS

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

5.1 The CONTSID toolbox

The key features of the CONTSID toolbox are (Garnier et al. (2006, 2009)):

- it supports most of the time-domain methods developed over the last thirty years (Garnier et al. (2003)) for identifying linear dynamic continuous-time parametric models from measured input/output DT data;
- it provides transfer function and state-space model identification methods for single-input single-output and multiple-input multiple-output systems, including both traditional and more recent approaches;
- it can handle mild irregularly sampled data in a straightforward way;
- it may be seen as an add-on to the system identification (SID) toolbox. To facilitate its use, it has been indeed given a similar setup to the SID toolbox;
- it provides a flexible graphical user interface (GUI) that lets the user analyse the experimental data, identify and evaluate models in an easy way;
- it can be freely downloaded from http://www.cran.uhp-nancy.fr/contsid/

5.2 The GUI for the CONTSID toolbox

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

- a model estimation panel in the middle where different model structures and identification methods to directly estimate a CT transfer function model can be tested;
- a model validation panel in the right part where basic properties of the identified model can be examined.

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

**Model estimation panel**

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

\[
A(p)y(t_k) = B(p)u(t_k) + e(t_k) \quad (22)
\]

\[
y(t_k) = \frac{B(p)}{F(p)} u(t_k) + e(t_k) \quad (23)
\]

where $p$ denotes the differential operator, i.e. $p := \frac{d}{dt}$; $u(t_k)$ and $y(t_k)$ represent the input and output signals at time-instant $t = kT_s$ respectively; $e(t_k)$ is a DT white Gaussian sequence; $A(p)$, $B(p)$ and $F(p)$ are polynomials in $p$. Equation (22) defines what is called a CT-ARX model structure while equation (23) defines a so-called CT-OE model structure. The user is thus invited to choose the type and the structure of his model in the two unrolling menus at the top of the *model estimation* panel, as shown in Figure 1.

After selecting the model structure, the user has to specify the polynomial orders and the time-delay of the model to be estimated. A first option is to deduce an estimate of the number of samples for the time-delay from an estimation of the impulse response by correlation analysis. Then, if the TF model order is not known *a priori*, the *Order search* button allows the user to automatically search over a whole range of different model orders. The user can choose several available criteria to sort and display the estimation results in the Matlab workspace. From these results, the user can select the best model orders and then set the order of the final model to be estimated by clicking on the *Order set* button from the main window.

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

- in the case of a CT-ARX model structure, the student can select the SVF approach coupled with simple least squares or auxiliary model-based instrumental variable methods. The design parameter should be chosen in order to emphasize the frequency band of interest;
- in the case of a CT-OE model structure, the student can select the simplified refined instrumental variable (SRIVC) method.

At the end, the students are invited to use, at a first choice, the SRIVC method since it can be automatically initiated and has been proven to be powerful in practice. Once chosen the parameter estimation method, the identified
model is displayed in the Matlab command window after a click on the Parameter estimation button.

Model validation panel
Once a model is estimated, it will appear in the drop-down menu located at the top part of the Model validation panel. Several basic model properties can then be evaluated from an unrolling menu by using first the data that were used for model identification:

- **model output comparison**: plots and compares the simulated model output with the measured output. This indicates how well the system dynamics are captured;
- **residual plot**: displays the residuals, i.e. the output simulation error;
- **transient response**: displays the model response to an impulse or step excitation signal;
- **frequency response**: displays the Nyquist or Bode plots to show damping levels and resonance frequencies;
- **zeros and poles**: plots the poles and zeros of the identified models and tests for zero-pole cancelation indicating over-parameterized modelling;
- **correlation test**: displays the sample auto-covariance function of the residuals and the sample cross-covariance function between the excitation signal and the residuals.

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

6. COMPLEX FLEXIBLE ROBOT ARM

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

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

Process description and modelling purpose
The robot arm is installed on an electrical motor. The modelling aim is here to design a control law based on a model between the measured reaction torque of the structure on the ground to the acceleration of the flexible arm. The robot arm is described in more detail in (Kollar (1994)).

Experiment design
The excitation signal is a multi-sine. The sampling period is set to 2 ms. Measurements are made with anti-aliasing filters. $K = 10$ periods each of length $M = 4096$ are exactly measured and a record of $N = KM = 40,960$ data points is collected.

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

Different model structures in the range $[n_b \ n_z \ n_k] = [4 \ 4 \ 0]$ to $[7 \ 6 \ 0]$ have been computed for the 3rd period data set. The other data set periods were kept for model validation purposes.

The 7 best models sorted according to Young’s information criterion (YIC) (see e.g. H. Garnier and L. Wang (Eds.) (2008)) are given in Table 1. From this table, the first
model with $[n_0\ n_1\ n_2]=[6\ 6\ 0]$ seems to be quite clear cut (it has the most negative YIC=$-0.19$, with the highest associated coefficient of determination $R^2=0.977$).

Table 1. Best SRIVC model orders for the robot arm data set

<table>
<thead>
<tr>
<th>$n_0$</th>
<th>$n_1$</th>
<th>$n_2$</th>
<th>YIC</th>
<th>$R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>6</td>
<td>0</td>
<td>-0.19</td>
<td>0.977</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>-0.56</td>
<td>0.966</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>0</td>
<td>0.46</td>
<td>0.959</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>0</td>
<td>-3.49</td>
<td>0.950</td>
</tr>
</tbody>
</table>

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

$$\hat{G}(s) = \frac{20.87(s - 6.185)(s^2 - 1.69s + 710.6)(s^2 + 8.435s + 2.012s)}{(s^2 + 1.03s + 2094)(s^2 + 0.9808s + 9905)(s^2 + 2.693s + 7.042s)}$$

This estimated model is characterised by three, lightly damped dynamic modes, as defined in Table 2.

Table 2. Eigenvalues and dynamic modes for the robot arm SRIVC model

<table>
<thead>
<tr>
<th>Real</th>
<th>Imag</th>
<th>Damping</th>
<th>Nat. Freq. (rad/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.52</td>
<td>45.76</td>
<td>0.0113</td>
<td>45.76</td>
</tr>
<tr>
<td>-0.52</td>
<td>-45.76</td>
<td>0.0113</td>
<td>45.76</td>
</tr>
<tr>
<td>-0.49</td>
<td>99.52</td>
<td>0.0049</td>
<td>99.52</td>
</tr>
<tr>
<td>-0.49</td>
<td>-99.52</td>
<td>0.0049</td>
<td>99.52</td>
</tr>
<tr>
<td>-1.35</td>
<td>265.37</td>
<td>0.0051</td>
<td>265.37</td>
</tr>
<tr>
<td>-1.35</td>
<td>-265.37</td>
<td>0.0051</td>
<td>265.37</td>
</tr>
</tbody>
</table>

Model validation
Figure 2 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 $R^2>0.95$. There is also a very good agreement between the ETFE and the frequency response of the estimated SRIVC model, as shown in Figure 3.


Fig. 2. Cross-validation results on a short section of the 8th-period data set

Fig. 3. Comparison of ETFE (‘×’) and SRIVC model (solid line) frequency responses for the flexible robot arm

7. CONCLUSION

This paper has outlined the main features of the data-based continuous-time modelling course that is taught for engineering students at Nancy-University. In particular the CONTSID toolbox which is so useful to perform the general identification procedure has been described and its use has been illustrated to model a complex flexible robot arm. Future developments for the course would be to introduce nonlinear system identification aspects to the students.

REFERENCES