A Toolbox for Recursive Identification of Dynamical Systems

Ľuboš Čirka, Miroslav Fikar

Abstract
The dynamical system IDentification TOOLbox (IDTOOL) for MATLAB/Simulink is presented. This toolbox implements blocks for continuous-time and discrete-time recursive least squares parameter estimation methods, respectively. The work-house routine used is LDDIF.

Key words: Identification, LDDIF, MATLAB, Simulink

Introduction

The most often used estimation methods nowadays are based on minimisation of squares of errors between measured and estimated process output – the least squares method (LS). Main advantages include the possibility of recursivity of calculations, possibility to track time variant parameters, and various modifications to make the algorithm robust. Originally the method was derived for estimation of parameters of discrete-time systems. Modifications are also possible for continuous-time systems.

This paper describes the use of the dynamical system IDentification TOOLbox (IDTOOL). The toolbox contains blocks for the identification of SISO (MIMO) linear time invariant systems. The development is based on the work published in [1], [2]. The toolbox supports only polynomial (transfer function) representation.

Previous versions of IDTOOL used S-function representation implemented in MATLAB for all blocks. However, this is not an option in some cases. For example real-time environment dSPACE can only use S-function in programming language C. Therefore, one of the new features of this version is C representation of blocks.

1. Algorithm

The identification method used as a work-house routine is LDDIF – recursive least squares algorithm with exponential and directional forgetting [2]. To improve the tracking performance, the corrections as suggested by Bittanti [1] are implemented. In the principle, the corrections influence the covariance matrix of the estimated parameters by adding some multiple of identity matrix.

This recursive estimation algorithm for given observation vector $\boldsymbol{\varphi}(t)$ and parameter vector $\boldsymbol{\Theta}(t-1)$ can be described by the equations:

$$\boldsymbol{\varepsilon}(t) = \boldsymbol{y}(t) - \boldsymbol{\varphi}(t)^T \boldsymbol{\Theta}(t-1)$$

$$r(t) = \boldsymbol{\varphi}(t)^T \boldsymbol{P}(t-1)\boldsymbol{\varphi}(t)$$

$$\boldsymbol{k}(t) = \frac{\boldsymbol{P}(t-1)\boldsymbol{\varphi}(t)}{1+r(t)}$$

$$\beta(t) = \begin{cases} \lambda - \dfrac{1-\lambda}{r(t)} & if \quad r(t) > 0 \\ 1 & if \quad r(t) = 0 \end{cases}$$

$$\boldsymbol{P}(t) = \boldsymbol{P}(t-1) - \frac{\boldsymbol{P}(t-1)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}(t)^T \boldsymbol{P}(t-1)}{\beta(t)^{-1} + r(t)} + \delta \boldsymbol{I}$$

$$\boldsymbol{\Theta}(t) = \boldsymbol{\Theta}(t-1) + \boldsymbol{k}(t)\boldsymbol{\varepsilon}(t)$$

where λ is exponential forgetting. The initial values for the algorithm are usually $\lambda = 0.985$, $\boldsymbol{P}(0) = 10^6 \boldsymbol{I}$, $\delta = 0.01$.

2. Implementation

This algorithm has been implemented in Simulink as M-function *lddif_s.m* and C-MEX-function *lddif_c.dll* with the inputs being observation vector $\boldsymbol{\varphi}(t)$ and actual process output $\boldsymbol{y}(t)$, states being the estimated parameter matrix $\boldsymbol{\Theta}(t)$ and the covariance matrix $\boldsymbol{P}(t)$, and outputs being the same as states. As Simulink allows only for vectors and not matrices of signals, all I/S/O vectors/matrices are stacked into corresponding column vectors. This formulation of a parameter estimation method is fairly general and thus any other method can be used in the place of LDDIF.

Of course, there are some parameters significant to LDDIF only (δ, λ) that are needed as well. Furthermore, in the actual implementation, it is possible to specify a priori knowledge about the parameters and their covariance matrix. For a more precise description of the LDDIF implementation see the manual[1].

2.1 Discrete-time Case

Consider a discrete-time multi-input multi-output (MIMO) system of the form

$$\boldsymbol{A}(z^{-1})\boldsymbol{y}(t) = \boldsymbol{B}(z^{-1})\boldsymbol{u}(t) \qquad (1)$$

where $\boldsymbol{A}[n \times n]$ and $\boldsymbol{B}[n \times m]$ are polynomial matrices given as

[1] http://www.kirp.chtf.stuba.sk/~cirka/idtool

$$\boldsymbol{A}(z^{-1}) = \boldsymbol{I} + \boldsymbol{A}_1 z^{-1} + \ldots + \boldsymbol{A}_{n_a} z^{-n_a} \tag{2}$$

$$\boldsymbol{B}(z^{-1}) = \boldsymbol{B}_1 z^{-1} + \ldots + \boldsymbol{B}_{n_b} z^{-n_b} \tag{3}$$

The matrix of estimated parameters $\boldsymbol{\Theta}$ is therefore of the form

$$\boldsymbol{\Theta}^T = (\boldsymbol{A}_1 \ldots \boldsymbol{A}_{n_a}\ \boldsymbol{B}_1 \ldots \boldsymbol{B}_{n_b}) \tag{4}$$

The needed input data for LDDIF are the actual output vector $\boldsymbol{y}(t)$ together with the observation vector $\boldsymbol{\varphi}(t)$ that are stacked together as

$$\begin{aligned}\boldsymbol{u}_{lddif}^T = [&\boldsymbol{y}^T(t), -\boldsymbol{y}^T(t-1), \ldots, -\boldsymbol{y}^T(t-n_a),\\ &\boldsymbol{u}^T(t-1), \ldots, \boldsymbol{u}^T(t-n_b)]\end{aligned} \tag{5}$$

The delayed signals are constructed using state-space filters.

2.1.1 Example

Let us assume case $n = 2$, $m = 3$, $n_a = 1$, $n_b = 2$ and matrices

$$\boldsymbol{A}_1 = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix},\ \boldsymbol{B}_1 = \begin{pmatrix} b_{111} & b_{112} & b_{113} \\ b_{121} & b_{122} & b_{123} \end{pmatrix} \tag{6}$$

$$\boldsymbol{B}_2 = \begin{pmatrix} b_{211} & b_{212} & b_{213} \\ b_{221} & b_{222} & b_{223} \end{pmatrix} \tag{7}$$

Then

$$\begin{aligned}\boldsymbol{u}_{lddif}^T = [&y_{1,t}, y_{2,t}, -y_{1,t-1}, -y_{2,t-1}, u_{1,t-1},\\ &u_{2,t-1}, u_{3,t-1}, u_{1,t-2}, u_{2,t-2}, u_{3,t-2}]\end{aligned} \tag{8}$$

$$\begin{aligned}\boldsymbol{\Theta}^T = [&a_{11}, a_{12}, b_{111}, b_{112}, b_{113}, b_{211}, b_{212}, b_{213},\\ &a_{21}, a_{22}, b_{121}, b_{122}, b_{123}, b_{221}, b_{222}, b_{223}]\end{aligned} \tag{9}$$

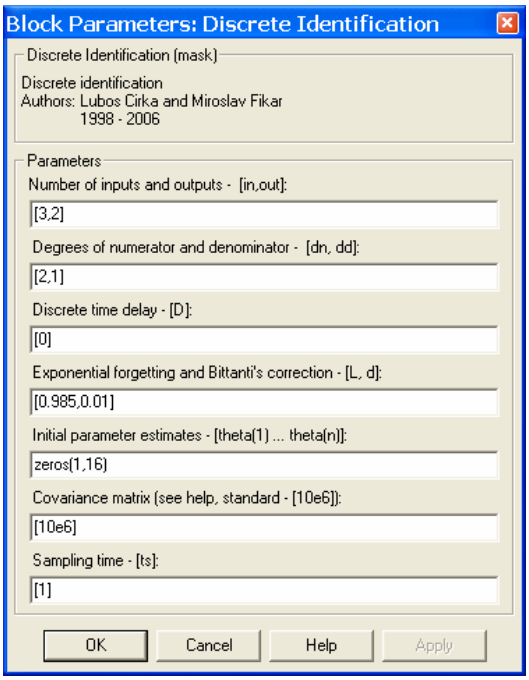

Fig.1 Block Parameters: Discrete Identification

In the Block Parameters (Fig. 1) of Discrete Identification block (Fig. 2), the following data have been entered: Number of inputs and outputs - [in, out]: [3,2], Degrees of numerator and denominator - [dn, dd]: [2,1]. All other parameters of this block can be left at default values.

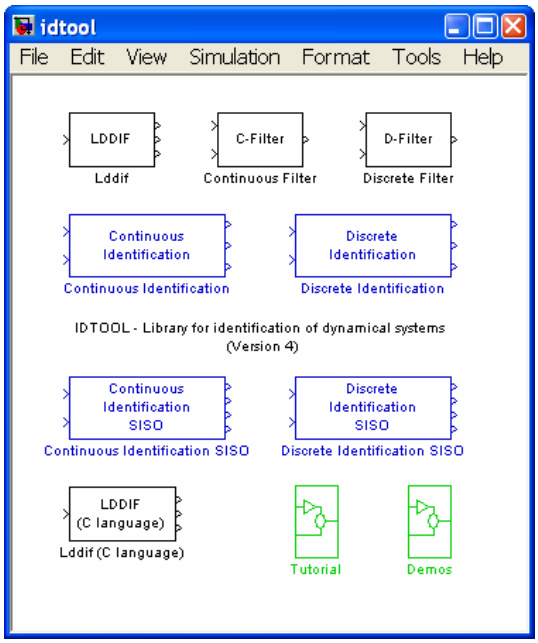

Fig.2 Library for identification of dynamical systems

2.2 Continuous-time Case

Consider a continuous-time MIMO system of the form

$$\boldsymbol{A}(s)\boldsymbol{y}(t) = \boldsymbol{B}(s)\boldsymbol{u}(t) \tag{10}$$

where $\boldsymbol{A}[n \times n]$ and $\boldsymbol{B}[n \times m]$ are polynomial matrices of the form

$$\boldsymbol{A}(s) = \boldsymbol{I} + \boldsymbol{A}_1 s + \ldots + \boldsymbol{A}_{n_a} s^{n_a} \tag{11}$$

$$\boldsymbol{B}(s) = \boldsymbol{B}_0 + \boldsymbol{B}_1 s + \ldots + \boldsymbol{B}_{n_b} s^{n_b} \tag{12}$$

or in the time domain as

$$\begin{aligned}\boldsymbol{y}(t) = &-\boldsymbol{A}_1 \dot{\boldsymbol{y}}(t) - \boldsymbol{A}_2 \ddot{\boldsymbol{y}}(t) - \ldots - \boldsymbol{A}_{n_a} \boldsymbol{y}^{(n_a)}(t)\\ &+ \boldsymbol{B}_0 \boldsymbol{u}(t) + \boldsymbol{B}_1 \dot{\boldsymbol{u}}(t) + \ldots + \boldsymbol{B}_{n_b} \boldsymbol{u}^{(n_b)}(t)\end{aligned} \tag{13}$$

The matrix of estimated parameters $\boldsymbol{\Theta}$ is therefore of the form

$$\boldsymbol{\Theta}^T = (\boldsymbol{A}_1 \ldots \boldsymbol{A}_{n_a}\ \boldsymbol{B}_0\ \boldsymbol{B}_1 \ldots \boldsymbol{B}_{n_b}) \tag{14}$$

The needed input data for LDDIF are the actual output vector $\boldsymbol{y}(t)$ together with the observation vector $\boldsymbol{\varphi}(t)$ that are stacked together as

$$\begin{aligned}\boldsymbol{u}_{lddif}^T = [&\boldsymbol{y}^T(t), -\dot{\boldsymbol{y}}^T(t), \ldots, -\boldsymbol{y}^{(n_a)^T}(t),\\ &\boldsymbol{u}^T(t), \dot{\boldsymbol{u}}^T(t), \ldots, \boldsymbol{u}^{(n_b)^T}(t)]\end{aligned} \tag{15}$$

Because the desired derivatives of $\boldsymbol{u}(t), \boldsymbol{y}(t)$ are not known, both input $\boldsymbol{u}(t)$ and output $\boldsymbol{y}(t)$ are filtered by a diagonal matrix $\boldsymbol{A}_f^{-1}$ so that

$$\begin{aligned}&\boldsymbol{A}\boldsymbol{y}(t) = \boldsymbol{B}\boldsymbol{u}(t) \Leftrightarrow \boldsymbol{A}_f^{-1}\boldsymbol{A}\boldsymbol{y}(t) = \boldsymbol{A}_f^{-1}\boldsymbol{B}\boldsymbol{u}(t)\\ &\Leftrightarrow \boldsymbol{A}\boldsymbol{y}_f(t) = \boldsymbol{B}\boldsymbol{u}_f(t)\end{aligned} \tag{16}$$

and thus the filtered variables are given as

$$\boldsymbol{A}_f \boldsymbol{y}_f(t) = \boldsymbol{y}(t) \tag{17}$$

$$\boldsymbol{A}_f \boldsymbol{u}_f(t) = \boldsymbol{u}(t) \tag{18}$$

As it can be seen from (16) the same relation holds for unfiltered and filtered signals. Therefore, eq. (13) can be written with the subscripts $_f$. The desired filtered variables and their derivatives are easily obtained from (17) and (18).

The filter in IDTOOL is of the form

$$F(s) = \boldsymbol{A}_f = (T_f s + 1)^{n_f} \boldsymbol{I} \tag{19}$$

The time constant T_f should generally be smaller than any time constant of the identified system. The polynomial order n_f must be at least equal to degree of the system denominator degree n_a.

2.2.1 Example

Let us assume case $n = 2$, $m = 2$, $n_a = 1$, $n_b = 0$ and matrices

$$\boldsymbol{A}_1 = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad \boldsymbol{B}_0 = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \tag{20}$$

Then

$$\boldsymbol{u}_{lddif}^T = [y_1, y_2, -\dot{y}_1, -\dot{y}_2, u_1, u_2] \tag{21}$$

$$\boldsymbol{\Theta}^T = [a_{11}, a_{12}, b_{11}, b_{12}, a_{21}, a_{22}, b_{21}, b_{22}] \tag{22}$$

For the system assumed, let us consider the filter with $n_f = 1$. From (17) and (18) follows

$$\begin{aligned} T_f \dot{y}_{1f} + y_{1f} &= y_1 \\ T_f \dot{y}_{2f} + y_{2f} &= y_2 \\ T_f \dot{u}_{1f} + u_{1f} &= u_1 \\ T_f \dot{u}_{2f} + u_{2f} &= u_2 \end{aligned} \tag{23}$$

Defining the states

$$\begin{aligned} x_1 &= y_{1f} \quad & x_5 &= u_{1f} \\ x_2 &= y_{2f} \quad & x_6 &= u_{2f} \\ x_3 &= \dot{y}_{1f} \quad & x_7 &= \dot{u}_{1f} \\ x_4 &= \dot{y}_{2f} \quad & x_8 &= \dot{u}_{2f} \end{aligned} \tag{24}$$

the equations (23) can be transformed into the corresponding state-space form. Using the available states, (13) now reads

$$\dot{x}_1 = -a_{11}x_3 - a_{12}x_4 + b_{11}x_5 + b_{12}x_6 \tag{25}$$

$$\dot{x}_2 = -a_{21}x_3 - a_{22}x_4 + b_{21}x_5 + b_{22}x_6 \tag{26}$$

and may directly be used for identification.

In the Block Parameters (Fig. 3) of Continuous Identification block (Fig. 2), the following data have been entered: Number of inputs and outputs - [in, out]: [2,2], Degrees of numerator and denominator - [dn, dd]: [0,1], Time constant of filters - [Tc]: [1]. All other parameters of this block can be left at default values.

3. Content of the Toolbox

The files included in the toolbox are as follows

lddif_s.m – the actual computational routine (M-file S-function)

lddif.c, lddif_c.c – the actual computational routine (C-file S-functions)

lddif_c.dll – the actual computational routine (MEX-file S-function)

idtool.mdl – simulink blocks

slblocks.m – defines the block library for a specific Toolbox or Blockset

id_demo1.mdl – demo identification of a continuous–time SISO system

id_demo2.mdl – demo identification of a discrete–time SISO system

id_demo3.mdl – demo identification of a continuous–time MIMO system

id_demo4.mdl – demo identification of a discrete–time MIMO system

tutorXX.mdl – simulink tutorial files

contents.m – contents of the files

demos.m – demo list information for IDTOOL Toolbox

info.xml – plugin of IDTOOL to the Start button (Block Library, Help, Demos, Product Page (Web))

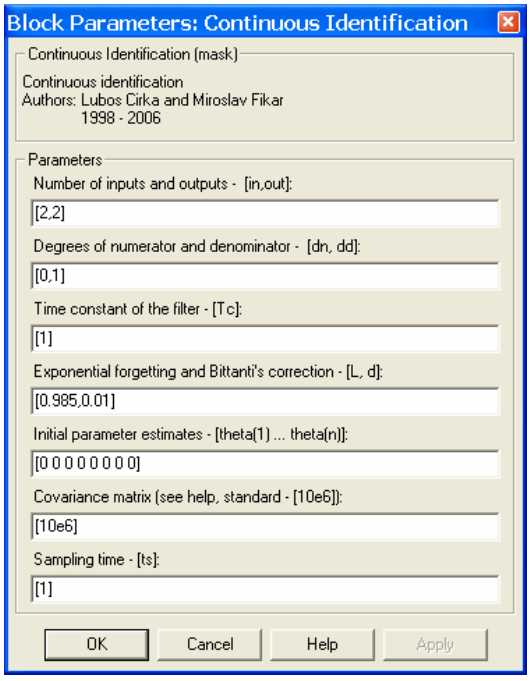

Fig.3 Block Parameters: Continuous Identification

4. Use of the Toolbox

Requirements

The IDTOOL toolbox v4.0 can be used under MATLAB 6.x and 7.x (older versions of MATLAB are not supported), and will run on most hardware. External toolboxes are not required.

Installing the Toolbox

The installation procedure is very easy. Just add the path to the directory where you have copied all the files to the stan-

dard MATLAB path. To load the blocks, simply write *idtool* at the MATLAB command line.

Help

To see a list of all the commands and functions that are available type *help idtool*. To get help on any of the toolbox commands, such as *lddif_s*, type help *lddif_s*.

5. Conclusions

The IDTOOL toolbox written in MATLAB language and C language, has been presented for system identification on the basis of work-house routine LDDIF. It has been designed as a tool for students and researchers. Toolbox provides a possibility to better understand different modifications of base recursive least squares algorithm as well as automation of the routine tasks.

The functions described in this paper are freely available for academic research per www[2]. A zipped file with the latest collection of MATLAB routines can be downloaded from there.

Acknowledgments

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under grants No. 1/3081/06 and 1/4055/07 and within the framework of the European Social Fund (PhD Students for Modern Industrial Automation in SR, JPD 3 2005/NP1-047, No 13120200115)

References

[1] BITTANTI, S., BOLZERN, P., CAMPI, M.: Exponential convergence of a modified directional forgetting identification algorithm. *Systems & Control Letters*, 14:131-137, 1990.

[2] KULHAVÝ, R., KÁRNÝ, M.: Tracking of slowly varying parameters by directional forgetting. In *Proc. 9th IFAC World Congr., Budapest, Hungary*, pages 79-83, 1984.

Ing. Ľuboš Čirka, PhD.
prof. Dr. Ing. Miroslav Fikar

Slovak University of Technology in Bratislava
Faculty of Chemical and Food Technology
Department of Information Engineering and Process Control
Radlinského 9
812 37 Bratislava
Tel.: ++421 2 52495269
E-mail: lubos.cirka@stuba.sk

[2] http://www.kirp.chtf.stuba.sk/~cirka/idtool