Slovak University of Technology in Bratislava
Institute of Information Engineering, Automation, and Mathematics

PROCEEDINGS

17^{th} International Conference on Process Control 2009

Hotel Baník, Štrbské Pleso, Slovakia, June 9 – 12, 2009

ISBN 978-80-227-3081-5

http://www.kirp.chtf.stuba.sk/pc09

Editors: M. Fikar and M. Kvasnica

Navrátil, P., Bobál, V.: Recursive Identification Algorithms Library, Editors: Fikar, M., Kvasnica, M., In *Proceedings of the 17th International Conference on Process Control '09*, Štrbské Pleso, Slovakia, 516–523, 2009.

Full paper online: http://www.kirp.chtf.stuba.sk/pc09/data/abstracts/017.html

RECURSIVE IDENTIFICATION ALGORITHMS LIBRARY

P.Navrátil*, V.Bobál*

** Department of Process Control, Faculty of Applied Informatics*
Tomas Bata University in Zlin
Nad Stráněmi 4511, 760 05 Zlín, Czech Republic
fax : +420 57 603 5196, e-mail :{p1navratil, bobal}@fai.utb.cz

Abstract: This paper presents simple SIMULINK library for recursive parameter estimation of linear dynamic models ARX, ARMAX and OE. Several recursive identification methods were implemented in this library: Least Square Method (RLS), Recursive Leaky Incremental Estimation (RLIE), Damped Least Squares (DLS), Adaptive Control with Selective Memory (ACSM), Instrumental Variable Method (RIV), Extended Least Square Method (RELS), Prediction Error Method (RPEM) and Extended Instrumental Variable Method (ERIV). To cope with tracking the time-variant parameters several forgetting factor and modification of basic algorithm are taken into consideration.

Keywords: Recursive estimation, ARX models, ARMAX models, forgetting factors.

1 INTRODUCTION

There exist many complex packages for system identification purposes in MATLAB and SIMULINK environment. These toolboxes provide solution to wide range of the problems from the area of system identification, e.g. System Identification Toolbox (Ljung 2004) and Continuous Identification Toolbox (Granier 2006).

There also exist many special-purpose programs and libraries for MATLAB and SIMULINK, e.g. Idtool (Cirka 1998). These simple tools provide solution to specific problems from the concrete part of the area of system identification.

The proposed Recursive Identification Algorithms Library (RIA) fall into category of simple libraries for SIMULINK environment and is designed for recursive estimation of the parameters of the linear dynamic models ARX, ARMAX and OE. The Recursive Identification Algorithms Library consists of several user-defined blocks. These blocks implement several recursive identification algorithms: Least Square Method (RLS) and its modifications, Recursive Leaky Incremental Estimation (RLIE), Damped Least Squares (DLS), Adaptive Control with Selective Memory (ACSM), Instrumental Variable Method (RIV), Extended Least Square Method (RELS), Prediction Error Method (RPEM) and Extended Instrumental Variable Method (ERIV). The Recursive Identification Algorithms Library can be used for simulation or real-time experiment (e.g. Real Time Toolbox) in educational process when it is possible to demonstrate the properties and behaviour of the recursive identification algorithms and forgetting factors under various conditions and can be also used in the identification part of self-tuning controllers.

2 MODEL STRUCTURE

The basic step in identification procedure is the choice of suitable type of the model.

General linear model takes the following form:

$$y(k) = \frac{B(q^{-1})}{A(q^{-1})F(q^{-1})}u(k) + \frac{C(q^{-1})}{A(q^{-1})D(q^{-1})}n(k) \quad (1)$$

where

$$\begin{aligned} A\left(q^{-1}\right) &= 1 + a_1 q^{-1} + \ldots + a_{na} q^{-na} \\ B\left(q^{-1}\right) &= b_1 q^{-1} + b_2 q^{-2} + \ldots + b_{nb} q^{-nb} \\ C\left(q^{-1}\right) &= 1 + c_1 q^{-1} + \ldots + c_{nc} q^{-nc} \\ D\left(q^{-1}\right) &= 1 + d_1 q^{-1} + \ldots d_{nd} q^{-nd} \\ F\left(q^{-1}\right) &= 1 + f_1 q^{-1} + \ldots + f_{nf} q^{-nf} \end{aligned} \tag{2}$$

are shift operators polynomials and $y(k)$, $u(k)$ are output and input signals. White noise $n(k)$ is assumed to have zero mean value and constant variance.

All linear models can be derived from general linear model by simplification. In the Recursive Identification Library following linear dynamic models are taken into consideration. These are ARX, ARMAX, OE models.

ARX model (C=D=F=1):

$$y(k) = \frac{B\left(q^{-1}\right)}{A\left(q^{-1}\right)} u(k) + \frac{1}{A\left(q^{-1}\right)} n(k) \tag{3}$$

ARMAX model (D=F=1):

$$y(k) = \frac{B\left(q^{-1}\right)}{A\left(q^{-1}\right)} u(k) + \frac{C\left(q^{-1}\right)}{A\left(q^{-1}\right)} n(k) \tag{4}$$

OE model (A=C=D=1):

$$y(k) = \frac{B\left(q^{-1}\right)}{F\left(q^{-1}\right)} u(k) + n(k) \tag{5}$$

3 RECURSIVE PARAMETER ESTIMATION

The recursive parameter estimation algorithms are based on the data analysis of the input and output signals from the process to be identified. Many recursive identification algorithms were proposed (Ljung, 1987, Söderström and Stoica, 1989; Wellstead and Zarrop, 1991). In this part several well-known recursive algorithms with forgetting factors implemented in Recursive Identification Algorithms Library are summarized.

3.1 *RLS*

This method can be used for parameter estimate of ARX model. The algorithm can be written in following form:

$$\begin{aligned} \hat{e}(k) &= \boldsymbol{y}(k) - \boldsymbol{\phi}^T(k)\hat{\boldsymbol{\Theta}}(k-1) \\ \boldsymbol{L}(k) &= \frac{\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)}{1 + \boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)} \\ \hat{\boldsymbol{\Theta}}(k) &= \hat{\boldsymbol{\Theta}}(k-1) + \boldsymbol{L}(k)\hat{e}(k) \\ \boldsymbol{C}(k) &= \boldsymbol{C}(k-1) - \boldsymbol{L}(k)\boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1) \end{aligned} \tag{6}$$

where: $\boldsymbol{L}(k)$ denote gain matrix, $\boldsymbol{C}(k)$ is the covariance matrix of the estimated parameters, $\hat{\boldsymbol{\Theta}}(k)$ is the vector that contains the estimated parameters and $\boldsymbol{\phi}(k)$ is the data or regression vector

$$\hat{\boldsymbol{\Theta}}(k) = [a_1, \ldots a_{na}, b_1, \ldots, b_{nb}]^T \tag{7}$$

$$\begin{aligned} \boldsymbol{\phi}^T(k) = [&y(k-1), \ldots, y(k-na), \\ &u(k-1), \ldots, u(k-nb)] \end{aligned} \tag{8}$$

This RLS algorithm assumes that the parameters of the model process are constant. In many cases, however, the estimator will be required to track changes in a set of parameters. To cope with tracking the time-variant parameters some adjustment mechanism must be introduced in the previous basic equations. Several implementations have been proposed (Ljung, 1987; Söderström and Stoica, 1989; Kulhavý and Zarrop, 1993; Corriou 2004).

RLS with exponential forgetting

Covariance matrix is given by

$$\boldsymbol{C}(k) = \frac{1}{\lambda}\left(\boldsymbol{C}(k-1) - \frac{\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)\boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)}{\lambda + \boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)}\right) \tag{9}$$

where $0 < \lambda < 1$ is forgetting factor.

The algorithm is convenient for identification and adaptive control of slowly varying systems.

This method has the main disadvantages that when the inputs is not persistent, and as the old data is discarded in the estimation procedure, the matrix $\boldsymbol{C}(k)$ increases exponentially with rate λ. This is called estimator wind-up.

RLS with variable exponential forgetting

The variable exponential forgetting is given by relation

$$\lambda(k) = \lambda_0 \lambda(k-1) + 1 - \lambda_0 \tag{10}$$

with $\lambda(0) = \lambda_0 \in \langle 0{,}95; 0{,}99 \rangle$

This algorithm is convenient for identification of time-invariant systems and self-tuning controllers.

RLS with fixed directional forgetting

To solve the problem of estimator wind-up, an estimator with directional forgetting can be used. This estimator forgets the information only in the directions in which new information is gathered and assures the convergence of the estimations and avoids large changes in the parameters.

Covariance matrix is

$$\boldsymbol{C}(k)=\boldsymbol{C}(k-1)-\frac{\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)\boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)}{\varepsilon^{-1}+\boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)} \quad (11)$$

and directional forgetting factor

$$\varepsilon(k-1)=\lambda'-\frac{1-\lambda'}{\boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)} \quad (12)$$

where λ' can be chosen as in exponential forgetting algorithm.

RLS with adaptive directional forgetting

Detailed description of this algorithm can be found in (Kulhavý, 1987).

$$\varepsilon(k)=\varphi(k)-\frac{1-\varphi(k)}{\xi(k-1)} \quad (13)$$

where

$$\xi(k-1)=\boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)\boldsymbol{\phi}(k) \quad (14)$$

The value of adaptive directional forgetting factor is

$$\varphi(k)=\left\{1+(1+\rho)\left[\ln(1+\xi(k-1))\right]+\left[\frac{(\upsilon(k-1)+1)\eta(k-1)}{1+\xi(k-1)+\eta(k-1)}-1\right]\frac{\xi(k-1)}{1+\xi(k-1)}\right\}^{-1} \quad (15)$$

and

$$\eta(k)=\frac{\hat{e}^2(k)}{\lambda(k)}$$

$$\upsilon(k)=\varphi(k)\left[\upsilon(k-1)+1\right] \quad (16)$$

$$\lambda(k)=\varphi(k)\left[\lambda(k-1)+\frac{\hat{e}^2(k-1)}{1+\xi(k-1)}\right]$$

RLS with exponential forgetting matrix

This technique is able to cope with the cases where parameters have distinct rates of change in time. Here, is described a recursive estimation algorithm with exponential forgetting matrix factors in order to provide distinct information discounts for each parameter.

The RLS with exponential forgetting matrix is governed by the following equations (Ljung 1987):

$$\Lambda(k-1)=\Omega\boldsymbol{C}(k-1)\Omega^T \quad (17)$$

$$\boldsymbol{L}(k)=\frac{\Lambda(k-1)\boldsymbol{\phi}(k)}{1+\boldsymbol{\phi}^T(k)\Lambda(k-1)\boldsymbol{\phi}(k)} \quad (18)$$

$$\boldsymbol{C}(k)=\Lambda(k-1)\left(\boldsymbol{I}-\frac{\boldsymbol{\phi}(k)\boldsymbol{\phi}^T(k)\Lambda(k-1)}{1+\boldsymbol{\phi}^T(k)\Lambda(k-1)\boldsymbol{\phi}(k)}\right) \quad (19)$$

with

$$\Omega=\begin{bmatrix}\frac{1}{\sqrt{\lambda_1}} & 0 & 0\\ 0 & \ddots & 0\\ 0 & 0 & \frac{1}{\sqrt{\lambda_1}}\end{bmatrix} \quad (20)$$

representing a matrix with diagonal elements equal to square roots of the forgetting factors associated to each column of the regression vector $\boldsymbol{\phi}$.

RLS with constant trace algorithm

Constant trace algorithm could also be used to keep the matrix $\boldsymbol{C}(k)$ limited, by scaling the matrix at each iteration in a way that trace of $\boldsymbol{C}(k)$ is constant. The regularized constant-trace algorithm is given by the following equations:

$$\hat{\boldsymbol{\Theta}}(k)=\hat{\boldsymbol{\Theta}}(k-1)+\boldsymbol{L}(k)\left(y(k)-\boldsymbol{\phi}^T(k)\hat{\boldsymbol{\Theta}}(k-1)\right) \quad (21)$$

$$\boldsymbol{L}(k)=\frac{\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)}{\lambda+\boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)} \quad (22)$$

$$\bar{\boldsymbol{C}}(k)=\frac{1}{\lambda}\left(\boldsymbol{C}(k-1)-\frac{\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)\boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)}{1+\boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)}\right) \quad (23)$$

$$\boldsymbol{C}(k)=c_1\frac{\bar{\boldsymbol{C}}(k)}{tr\left(\bar{\boldsymbol{C}}(k)\right)}+c_2\boldsymbol{I} \quad (24)$$

in which c_1 and c_2 have positive values given by,

$$\frac{c_1}{c_2}=10000,\ \boldsymbol{\phi}^T(k)\boldsymbol{\phi}(k)c_1 \gg 1 \quad (25)$$

Exponential Forgetting and Resetting Algorithm

This modification of RLS places upper and lower bounds on the trace of the covariance matrix while maintaining a robustly valued forgetting factor. The algorithm takes the following form:

$$\hat{\boldsymbol{\Theta}}(k)=\hat{\boldsymbol{\Theta}}(k-1)+\alpha\boldsymbol{L}(k)\hat{e}(k) \quad (26)$$

$$\boldsymbol{L}(k)=\frac{\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)}{\lambda+\boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)} \quad (27)$$

$$\boldsymbol{C}(k)=\frac{1}{\lambda}\left[C(k-1)-\boldsymbol{L}(k)\boldsymbol{\phi}^{T}(k)C(k-1)\right] + \beta\boldsymbol{I}-\gamma C(k-1)^{2} \quad (28)$$

$$\sigma_{\min}\boldsymbol{I}\le C(k-1)\le\sigma_{\max}\boldsymbol{I} \qquad \forall k \quad (29)$$

$$\sigma_{\min}\approx\frac{\beta}{\alpha-\eta},\sigma_{\max}\approx\frac{\eta}{\gamma}+\frac{\beta}{\eta},\eta=\frac{1-\lambda}{\lambda}$$
$$\alpha=0,5;\beta=\gamma=0,005;\lambda=0,95; \quad (30)$$
$$\sigma_{\min}=0,01;\sigma_{\max}=10$$

3.2 *RWLS*

The recursive weighted least square (Nelles 2001) where the weighting data $\boldsymbol{\phi}(k)$ is denoted as $q(k)$ becomes

$$\hat{e}(k)=\boldsymbol{y}(k)-\boldsymbol{\phi}^{T}(k)\hat{\boldsymbol{\Theta}}(k-1)$$
$$\boldsymbol{L}(k)=\frac{\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)}{\boldsymbol{\phi}^{T}(k)\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)+\frac{\lambda}{q(k)}} \quad (31)$$
$$\hat{\boldsymbol{\Theta}}(k)=\hat{\boldsymbol{\Theta}}(k-1)+\boldsymbol{L}(k)\hat{e}(k)$$
$$\boldsymbol{C}(k)=\boldsymbol{C}(k-1)-\boldsymbol{L}(k)\boldsymbol{\phi}^{T}(k)\boldsymbol{C}(k-1)$$

where $0<\lambda<1$ is forgetting factor.

3.3 *RLIE*

The recursive leaky incremental estimation (Zhou and Cluett, 1996) can be describes as follows:

$$\hat{\boldsymbol{\Theta}}(k)=\gamma\hat{\boldsymbol{\Theta}}(k-1)+\Gamma\hat{\boldsymbol{\Theta}}(k) \quad (32)$$

$$\Gamma\hat{\boldsymbol{\Theta}}(k)=\Gamma\hat{\boldsymbol{\Theta}}(k-1)+C(k)\boldsymbol{\phi}(k)\Big(y(k) -\gamma\boldsymbol{\phi}^{\boldsymbol{T}}(k)\hat{\boldsymbol{\Theta}}(k-1)-\boldsymbol{\phi}^{\boldsymbol{T}}(k)\Gamma\hat{\boldsymbol{\Theta}}(k-1)\Big) \quad (33)$$

$$\boldsymbol{C}(k)=\frac{1}{\lambda}\left(\boldsymbol{C}(k-1)-\frac{\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)\boldsymbol{\phi}^{T}(k)\boldsymbol{C}(k-1)}{\lambda+\boldsymbol{\phi}^{T}(k)\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)}\right) \quad (34)$$

where Γ denotes the stabilizing operator, defined as

$$\Gamma=1-\gamma q^{-1} \quad (35)$$

and $\gamma\in[0,1]$ is the stabilizing parameters which is preselected by the user.

3.4 *DSL*

Damped least squares (DLS) algorithm is an extended version of the recursive simple least square (RLS) algorithm.

The DLS criterion is

$$J(\hat{\boldsymbol{\Theta}})=\sum_{k=t-N}^{t}\lambda^{t-k}\left[y(k)-\boldsymbol{\phi}^{T}(k)(k)\hat{\boldsymbol{\Theta}}(k)\right]^{2} +\left[\boldsymbol{\Lambda}_{d}(k)\left(\hat{\boldsymbol{\Theta}}(t)-\hat{\boldsymbol{\Theta}}(t-1)\right)\right]^{2} \quad (36)$$

The weighting matrix $\boldsymbol{\Lambda}_d(k)$ is diagonal and weights the parameters variations. For an n-parameters model,

$$\Lambda_{d}(t)=diag\left[\alpha_{1}(t) \quad \alpha_{2}(t)\ldots\alpha_{n}(t)\right] \quad (37)$$

A standart form of the DLS algorithm is given (Lambert 1987)

$$\hat{\boldsymbol{\Theta}}(k)=\hat{\boldsymbol{\Theta}}(k-1)+\boldsymbol{L}\left[y(k)-\hat{\boldsymbol{\Theta}}^{T}(k-1)\boldsymbol{\phi}(k)\right] +C(k)\lambda(k)\Lambda_{d}(k)\left[\hat{\boldsymbol{\Theta}}(k-1)-\hat{\boldsymbol{\Theta}}(k-2)\right] \quad (38)$$

$$\boldsymbol{L}(k)=\frac{\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)}{\lambda(k)+\boldsymbol{\phi}^{T}(k)\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)} \quad (39)$$

$$\boldsymbol{C}(k)=\frac{1}{\lambda(k)}\left(\boldsymbol{C}'(k)-\frac{\boldsymbol{C}'(k)\boldsymbol{\phi}(k-1)\boldsymbol{\phi}^{T}(k-1)\boldsymbol{C}'(k)}{\lambda(k)+\boldsymbol{\phi}^{T}(k-1)\boldsymbol{C}'(k)\boldsymbol{\phi}(k-1)}\right) \quad (40)$$

$$\boldsymbol{C}'(k)=\boldsymbol{C}(k-1)-\sum_{i=1}^{n}\frac{\boldsymbol{C}'_{i-1}(k-1)\mathrm{r_i r_i^T}\boldsymbol{C}'_{i-1}(k-1)\alpha'_{i}(k)}{1+\mathrm{r_i^T}\boldsymbol{C}'_{i-1}(k-1)\mathrm{r_i}\alpha'_{i}(k)} \quad (41)$$

$$\boldsymbol{C}'_{1}(k-1)=\boldsymbol{C}'_{i-1}(k-1)-\frac{\boldsymbol{C}'_{i-1}(k-1)\mathrm{r_i r_i^T}\boldsymbol{C}'_{i-1}(k-1)\alpha'_{i}(k)}{1+\mathrm{r_i^T}\boldsymbol{C}'_{i-1}(k-1)\mathrm{r_i}\alpha'_{i}(k)} \quad (42)$$

$$\boldsymbol{C}'_{0}(k-1)=\boldsymbol{C}(k-1) \quad (43)$$

where $\boldsymbol{r}_{\mathrm{i}}$ are the succesive basic vectors, e.g.

$$r_{i}=[1..0\cdots0]^{T} \quad (44)$$

and

$$\alpha'_{i}(k)=\frac{\alpha_{i}(k)-\lambda(k)\alpha_{i}(k-1)}{\lambda(k)} \quad (45)$$

3.5 *ACSM*

Adaptive control with selective memory (Hill and Ydstide, 2004) updates parameter estimates only when there is new information present. The information increases and estimator eventually stops.

The algorithm consists of several steps:

Step 0: Choose $r_0>0,\boldsymbol{\Theta}(0),\boldsymbol{C}(0)>0,1<M_0<\infty$

Set $r(0) = r_0 > 0, \sigma = 1 - \frac{1}{M_0}, \varepsilon_0 = \frac{1}{M_0}$.

Step 1:

$$r(k) = \max\left\{\sigma r(k-1) + (1-\sigma)\hat{e}(k-1)^2, r_0\right\} \quad (46)$$

where $\hat{e}(k) = \boldsymbol{y}(k) - \boldsymbol{\phi}^T(k)\hat{\boldsymbol{\Theta}}(k-1)$

Step 2: Set $B(k) = 0$ and

$$A(k) = \begin{cases} 1 & \textit{if } \dfrac{\boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)}{r(k)} \geq \varepsilon_0 \\ 0 & \textit{otherwise} \end{cases} \quad (47)$$

Step 3: If $A(k) = 0$, set

$$B(k) = \begin{cases} 1 & \textit{if } r(k) \geq \max_{1 \leq i \leq k} r(k-i) \\ 0 & \textit{otherwise} \end{cases} \quad (48)$$

Set $\Delta(k) = A(k) + B(k)$

Step 4:

$$\hat{\boldsymbol{\Theta}}(k) = \hat{\boldsymbol{\Theta}}(k-1) + \Delta(k)\frac{C(k-1)\boldsymbol{\phi}(k)}{r(k) + \boldsymbol{\phi}^T(k)C(k-1)\boldsymbol{\phi}(k)}\hat{e}(k) \quad (49)$$

Step 5:

$$\boldsymbol{C}(k) = \boldsymbol{C}(k-1) - \Delta(k)\frac{\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)\boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)}{r(k) + \boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)\boldsymbol{\phi}(k)} \quad (50)$$

Step 6: Set $k = k+1$ and go to step 1

3.6 *RIV*

It can be shown that if the process does not meet the noise assumption made by the ARX model, the parameters are estimated biased and nonconsistent. This problem can be avoided using instrumental variable method.

The algorithm takes the form

$$\begin{aligned} \hat{e}(k) &= \boldsymbol{y}(k) - \boldsymbol{\phi}^T(k)\hat{\boldsymbol{\Theta}}(k-1) \\ \boldsymbol{L}(k) &= \frac{\boldsymbol{C}(k-1)\boldsymbol{z}(k)}{1 + \boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1)\boldsymbol{z}(k)} \\ \hat{\boldsymbol{\Theta}}(k) &= \hat{\boldsymbol{\Theta}}(k-1) + \boldsymbol{L}(k)\hat{e}(k) \\ \boldsymbol{C}(k) &= \boldsymbol{C}(k-1) - \boldsymbol{L}(k)\boldsymbol{\phi}^T(k)\boldsymbol{C}(k-1) \end{aligned} \quad (51)$$

where: $\boldsymbol{L}(k)$ denote gain matrix, $\boldsymbol{C}(k)$ is the covariance matrix of the estimated parameters, $\hat{\boldsymbol{\Theta}}(k)$ is the vector that contains the estimated parameters, $\boldsymbol{\phi}(k)$ is the data or regression vector, $\boldsymbol{z}(k)$ is instrumental variable

$$\hat{\boldsymbol{\Theta}}(k) = [a_1, \dots a_{na}, b_1, \dots, b_{nb}]^T \quad (52)$$

$$\begin{aligned} \boldsymbol{\phi}^T(k) = [&y(k-1), \dots, y(k-na), \\ &u(k-1), \dots, u(k-nb)] \end{aligned} \quad (53)$$

Choice of instrumental variable determines behaviour of the IV method in usage. Some common choices for generating instruments are proposed in (Söderström and Stoica, 1989).

Typical choice of model independent instrumental variable is

$$\boldsymbol{z}(k) = [u(k-1), \dots, u(k-na-nb)]^T \quad (54)$$

and model dependent instrument is

$$\begin{aligned} \boldsymbol{z}(k) = [&y_u(k-1), \dots, y_u(k-n_a), \\ &u(k-1), \dots, u(k-n_b)]^T \end{aligned} \quad (55)$$

where $y_u(k-1)$ is generated by calculating following difference equation with current parameter estimates

$$\begin{aligned} y_u(k) = &\hat{b}_1(k)(k-1) + \dots + \hat{b}_{nb}(k)(k-nb) \\ &- \hat{a}_1(k)y_u(k-1) - \dots - \hat{a}_{na}(k)y_u(k-na) \end{aligned} \quad (56)$$

3.7 *RELS*

This method is used for parameter estimations of ARMAX model. Formally it takes the same form as RLS. However, the regression and parameter vector are different.

Parameter vector

$$\hat{\boldsymbol{\Theta}}(k) = [a_1, \dots a_{na}, b_1, \dots, b_{nb}, c_1, \dots, c_{nc}]^T \quad (57)$$

Regression vector

$$\begin{aligned} \boldsymbol{\phi}^T(k) = [&y(k-1), \dots, y(k-na), \\ &u(k-1), \dots, u(k-nb), \\ &\eta(k-1), \dots, \eta(k-nc)] \end{aligned} \quad (58)$$

or

$$\begin{aligned} \boldsymbol{\phi}^T(k) = [&y(k-1), \dots, y(k-na), \\ &u(k-1), \dots, u(k-nb), \\ &\hat{e}(k-1), \dots, \hat{e}(k-nc)] \end{aligned} \quad (59)$$

where $\eta(k)$ denotes the residual and $\hat{e}(k)$ is the prediction error.

It usually speeds up the convergence of the RELS algorithm if the residuals (*a posteriori*) rather than the prediction errors (*a priori*) are used.

3.8 *ERIV*

This method ensures improved accuracy and greater speed of convergence than RIV. The method is based on choice of instruments vector which has more elements than there are parameters in the model to be estimated. Derivation of this algorithm can be found in (Söderström and Stoica 1989). Instruments can be chosen according to (Branica *et. al.* 1996, Söderström and Stoica 1989).

The set of equations describe this algorithm

$$
\begin{aligned}
&\hat{\boldsymbol{\Theta}}(k)=\hat{\boldsymbol{\Theta}}(k-1)+\boldsymbol{L}(k)\left(\boldsymbol{v}(k)-\boldsymbol{\Phi}^T(k)\hat{\boldsymbol{\Theta}}(k-1)\right)\\
&\boldsymbol{L}(k)=\boldsymbol{P}(k-1)\boldsymbol{\Phi}(k)\left(\boldsymbol{\Lambda}(k)+\boldsymbol{\Phi}^T(k)\boldsymbol{P}(k-1)\boldsymbol{\Phi}(k)\right)^{-1}\\
&\boldsymbol{\Phi}(k)=\left[\boldsymbol{w}(k)\quad \boldsymbol{\phi}(k)\right]\\
&\boldsymbol{w}(k)=\boldsymbol{R}^T(k-1)\boldsymbol{z}(k)\\
&\Lambda(k)=\begin{bmatrix}-\boldsymbol{z}^T(k)\boldsymbol{z}(k) & 1\\ 1 & 0\end{bmatrix}\\
&\boldsymbol{v}(k)=\begin{bmatrix}\boldsymbol{z}^T(k)\boldsymbol{r}(k-1)\\ \boldsymbol{y}(k)\end{bmatrix}\\
&\boldsymbol{R}(k)=\boldsymbol{R}(k-1)+\boldsymbol{z}(k)\boldsymbol{\phi}^T(k)\\
&\boldsymbol{r}(k)=\boldsymbol{r}(k-1)+\boldsymbol{z}(k)\boldsymbol{y}(k)\\
&\boldsymbol{P}(k)=\boldsymbol{P}(k-1)-\boldsymbol{L}(k)\boldsymbol{\Phi}^T(k)\boldsymbol{P}(k-1)
\end{aligned}
\tag{60}
$$

3.9 *RPEM*

The recursive prediction error method (RPEM) allows the online identification of all linear model structure. Since all model structure except ARX are nonlinearly parameterized, no exact recursive algorithm can exist; rather some approximations must be made (Moore and Boel 1986, Moore and Weiss 1979, Söderström and Stoica 1989). In fact, the RPEM can be seen as a nonlinear least squares Gauss-Newton method.

The Gauss-Newton technique is based on the approximation of the Hessian by the gradients. Thus, the RPEM requires the calculation of the gradient $\boldsymbol{\psi}(k)$ of the model output with respect to its parameters:

$$\boldsymbol{\psi}^T(k)=\frac{\partial\,\hat{y}(k)}{\partial\,\boldsymbol{\Theta}(k)}=\left[\frac{\partial\,\hat{y}(k)}{\partial\,\Theta_1(k)}\quad\frac{\partial\,\hat{y}(k)}{\partial\,\Theta_2(k)}\quad\cdots\quad\frac{\partial\,\hat{y}(k)}{\partial\,\Theta_n(k)}\right]\tag{61}$$

RPEM algorithm takes the form

$$
\begin{aligned}
&\hat{e}(k)=y(k)-\boldsymbol{\phi}^T(k)\hat{\boldsymbol{\Theta}}(k-1)\\
&\boldsymbol{L}(k)=\frac{\boldsymbol{P}(k-1)\boldsymbol{\psi}(k)}{1+\boldsymbol{\psi}^T(k)\boldsymbol{P}(k-1)\boldsymbol{\psi}(k)}\\
&\hat{\boldsymbol{\Theta}}(k)=\hat{\boldsymbol{\Theta}}(k-1)+\boldsymbol{L}(k)\hat{e}(k)\\
&\boldsymbol{P}(k)=\boldsymbol{P}(k-1)-\boldsymbol{L}(k)\boldsymbol{\psi}^T(k)\boldsymbol{P}(k-1)
\end{aligned}
\tag{62}
$$

where $\boldsymbol{P}(k)$ denotes covariance matrix.

The model structure will influence the way in which the quantities $\hat{e}(k)$ and $\boldsymbol{\psi}(k)$ in the algorithm are computed from data and the previously computed parameter estimate.

4 RECURSIVE IDENTIFICATION ALGORITHMS LIBRARY (RIA)

The Recursive Identification Algorithm Library is designed for recursive parameter estimation of linear dynamics model ARX, ARMAX, OE using recursive identification methods: Least Square Method (RLS), Recursive Leaky Incremental Estimation (RLIE), Damped Least Squares (DLS), Adaptive Control with Selective Memory (ACSM), Instrumental Variable Method (RIV), Extended Least Square Method (RELS), Prediction Error Method (RPEM) and Extended Instrumental Variable Method (ERIV).

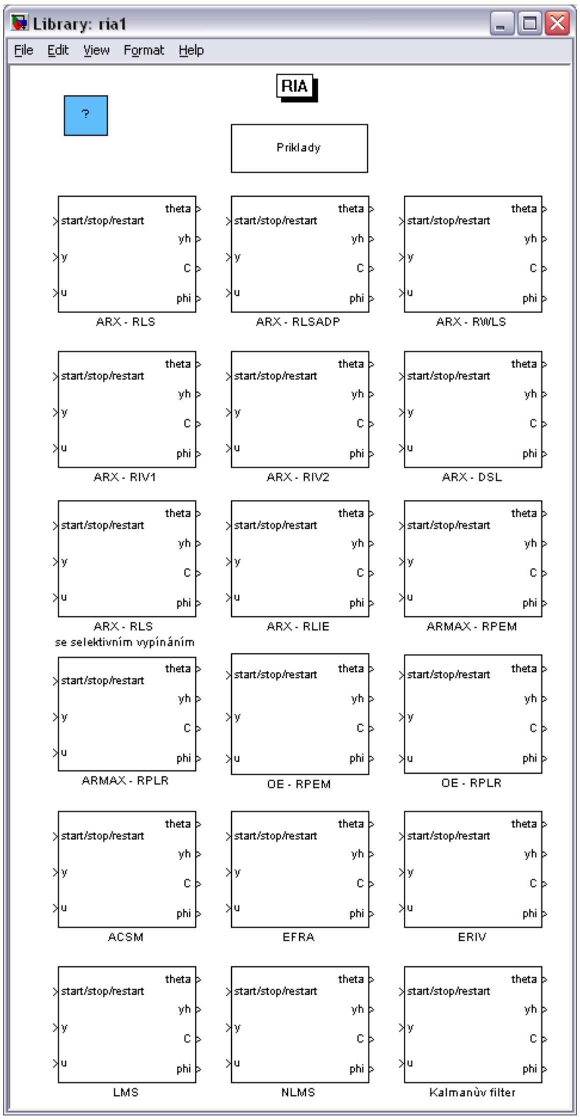

Fig. 1. Recursive Identification Algorithms Library

The Recursive Identification Algorithm Library is depicted in Figure1. The Library consists of 18 user-defined blocks and is designed for MATLAB&SIMULINK environment. Each block is realized as an s-function.

Each block is masked by user-defined dialog. Several necessary input parameters should be input through this dialog. These are: type of forgetting factor and its value, degrees of polynomials, sampling period, initial values of parameter estimate, covariance matrix and data vector, etc. Each block also contains the help describes the meaning of each parameter, inputs and outputs and used recursive identification algorithms. Example of input dialog is shown in Figure 2.

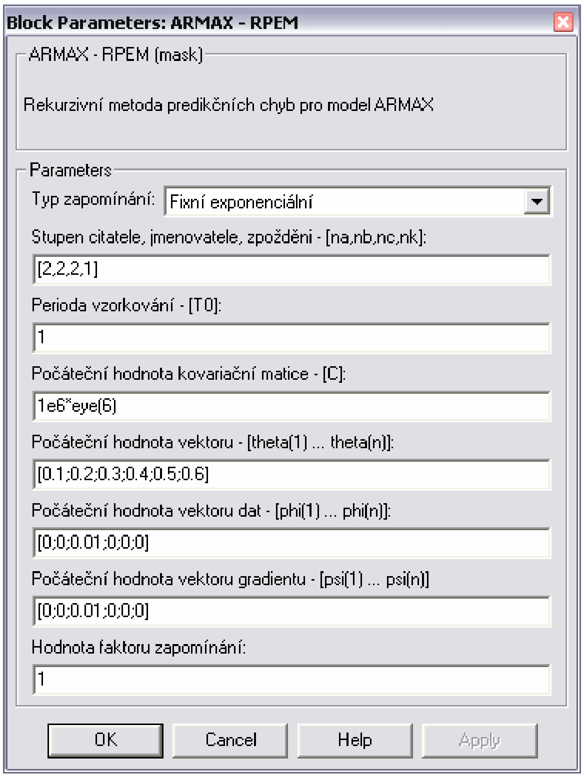

Fig. 2. Input dialog of the identification block

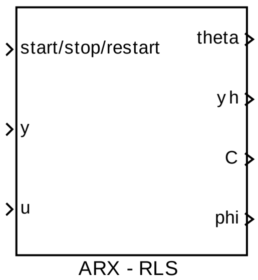

Fig. 3. Inputs/outputs of the identification block

Input/output data from object under identification process are inputs to the identification block. Another input (start/stop/restart) is used for control the identification algorithm. This input provides possibility of start, stop and restart the identification algorithm in selected instant of time. Outputs of the block are estimate of parameter vector, one-step prediction of output of model, covariance matrix and data vector. The inputs and outputs of the block are shown in Figure 3.Example of application of the identification block in the model is illustrated in Figure 4.

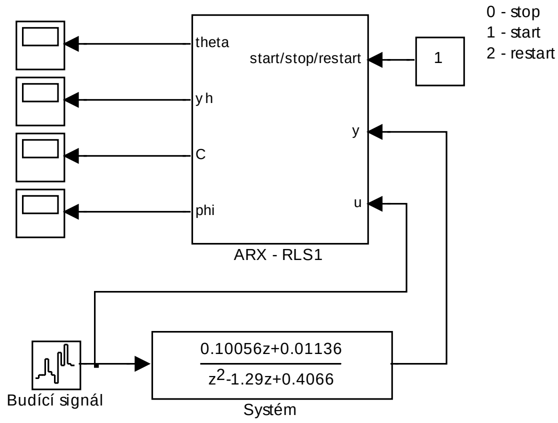

Fig. 4. Example of application of identification block

5 CONCLUSION

The Recursive Identification Algorithm Library is designed for recursive parameter estimation of linear dynamics model ARX, ARMAX, OE using recursive identification methods. The library can be used e.g. in identification part of self-tuning controller or in educational process when it is possible to demonstrate the properties and behaviour of the recursive identification algorithms and forgetting factors under various conditions.

ACKNOWLEDGMENTS

This work was supported by grant No. MSM 7088352101 of the Ministry of Education of the Czech Republic.

6 REFERENCES

Bobál, V., J. Böhm, J. Fessl and J. Macháèek (2005). *Digital Self-tuning Controllers: Algorithms, Implementation and Applications.* Springer-Verlag, London.

Branica, I., N. Peric and I. Petrovic (1996). Comparison of Several Recursive Identification Methods. *Automatika*,Vol. **37**, No. 3-4, pp. 99-104.

Cirka, L. and M. Fikar (1998). *Identification tool for Simulink.* Department of Process Control, FCT STU, Technical Report KAMF9803.

Corriou, J. P. (2004). *Process Control : Theory and Applications*. Springer-Verlag, London.

Cunha, J. B., P.B. Oliviera, J.P. Coelho (2004). Recursive Parameter Estimation Algorithms. In *controlo 2004: 6th Portuguese Conference on Automatic Control*. pp. 109-114.

Garnier, H. (2006). *Continuous-Time System Identification Toolbox*. [online]. http://www.iris.cran.uhp-nancy.fr/contsid/

Hill, J.H. and B.E. Ydstie (2004). Adaptive control with selective memory. *International Journal of Adaptive Control and Signal Processing*, Vol. **18**, No 7, pp. 571-587.

Kulhavý, R. (1987). Restricted exponential forgetting in real time identification. *Automatica*, Vol. **23**, pp. 586-600.

Kulhavý, R. and M.B. Zarrop (1993). On a general concept of forgetting. *International Journal of Control,* **58**, 4: pp. 905–924.

Ljung, L. (1987). *System identification – theory for user.* Prentice-Hall, Englewood Cliffs, N.J.

Ljung, L. (2004). System Identification Toolbox: For use with MATLAB. Prentice-Hall, Englewood Cliffs, N.J.

Lambert, E. P. (1987). Process control applications of long-range prediction. DPhill thesis, University of Oxford.

Moore, J. B. and R. K. Boel (1986). Asymptotically Optimum Recursive Prediction Error Methods in Adaptive Estimation and Control. *Automatica*, Vol. **22**, No. 2, pp. 237-240.

Moore, J. B., Weiss, H. (1979). Recursive Prediction Error Methods for Adaptive Estimation. *IEEE Trans. on Systems, Man, and Cybernetics.* Vol. **SMC-9**, No. 4, pp. 197-205.

Nelles, O. (2001). *Nonlinear system identification. S*pringer-Verlag*.* Berlin.

Söderström, T. and P. Stoica (1989). *System Identification.* Prentice Hall, University Press, Cambridge, UK.

Wellstead, P.E. and M.B. Zarrop (1991). *Self-Tuning System – Control and Signal Processing,* John Wiley & Sons Ltd. Chichester.Able, B.C. (1956). Nucleic acid content of microscope. *Nature* **135**, 7-9

Zhou, Q. and W.R. Cluett (1996). Recursive Identification of Time-varying Systems via Incremental Estimation. Automatica, Vol. **32**, No. 10, pp 1427-1431.