

# Online Identification Using RLS Algorithm and Kaczmarz's Projection Algorithm for a Bioreactor Process

S.Sundari<sup>1</sup>, Alamelu Nachiappan<sup>2</sup>

<sup>1</sup> Pondicherry Engineering College,  
Pondicherry, India.  
sunharsan@gmail.com

<sup>2</sup> Pondicherry Engineering College,  
Pondicherry, India.  
nalaam63@yahoo.com

**Abstract:** The control theory and automation technology is widely used in industries. Many control algorithms are based on the mathematical models of dynamic systems. Mathematical models and parameter estimation are basics for automatic control. A recursive least squares parameter estimation algorithms and Kaczmarz's projection algorithm is applied based on ARMAX model and OE models. The estimated parameters are compared with the true ones. The proposed method has been applied to identify the parameters of bioreactor process

**Keywords:** Biochemical reactor, RLS algorithm, system identification, Kaczmarz's projection algorithm, OE model, Mathematical modelling, ARMAX mode.

## 1. Introduction

Nowadays the use of batch reactors in the chemical, pharmaceutical, food and beverage industries is very common. According to some statistics, 50% of all production processes in industry are batch type processes. The term batch reactor is used for a variety of process operations, such as chemical reactions, product mixing, batch distillation, crystallization, solid dissolution, and polymerization. In some cases these reactors have specific role dependent, such as crystallizer or bio reactor. For optimization of cell mass growth and product formation continuous mode of operation of bioreactors are desirable. One of the major challenges in direct adaptive control is the existence of non minimum-phase zeros. Difficulties of identification for multivariable controlled autoregressive moving average (ARMA) systems are that there exists unknown noise terms in the information vector and the iterative identification can be used for the system with unknown terms in the information vector. By means of the hierarchical identification principle, those noise terms in the information vector are replaced with the estimated residuals and a least squares based iterative algorithm is proposed in [1], for multivariable controlled ARMA systems. In [2], identification problems of a class of nonlinear systems are considered. A stochastic gradient algorithm is developed by introducing a switching function and the finite impulse response model, the identification model of a nonlinear system is obtained. In [3], a two-stage recursive least squares (TS-RLS) algorithm is developed for pseudo-linear regressive models by combining the auxiliary model identification idea and the decomposition technique. The system is decomposed into two

subsystems with the system model parameters and the noise model parameters, and the parameters are identified for each subsystem. Compared with the auxiliary model based recursive generalized extended least squares algorithm, the TS-RLS algorithm has less computational burden. The identification problem of a class of linear-in-parameters is dealt in [4], for output error moving average systems. The difficulty of identification is that there exist some unknown variables in the information vector. By means of the auxiliary model identification idea, an auxiliary model based recursive least squares algorithm is developed for identifying the parameters. Modeling and parameter identification of multiple-input single output Wiener nonlinear systems is identified in [5], in which a multiple-input single output Wiener nonlinear model is identified and gradient-based iterative algorithm is derived for the proposed model. The proposed method has been applied to identify the parameters of a glutamate fermentation process. A hierarchical least-squares based iterative identification algorithm is derived in [6], for multivariable systems with moving average noises. A hierarchical identification principle and iterative identification principle is combined and a multivariable system is decomposed into two subsystems, one containing a parameter vector and the other containing a parameter matrix. This algorithm performs a hierarchical computational process at each iteration. The least-squares based iterative algorithm makes full use of all data at each iteration by which highly accurate parameter estimates are generated. The criterion functions of the recursive parameter estimation algorithms are explained in [7], for linear regressive models and pseudo-linear regressive models, including the equation error models and the output error models. The computation of the criterion functions has been implemented

by the recursive formulas. An iterative least squares algorithm is used to estimate the parameters of output error systems in [8] and uses the partitioned matrix inversion lemma to implement the proposed algorithm in order to enhance computational efficiencies.

This paper is organized as follows. Section 2 describes the mathematical modeling of a reactor followed by state space model and operating conditions of a reactor. Section 3 describes the system identification procedure. Online identification of a bioreactor process is developed in section 4 using RLS algorithm and Kaczmarz's projection algorithm. Section 5 argues the simulation results of both the algorithms using Matlab coding technique. Finally some conclusions are given in Section 6, followed by references

## 2. MODEL OF BIOREACTOR

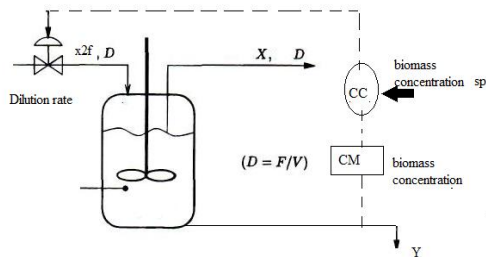


Figure 1 Schematic diagram of a continuous bioreactor

Table 1: parameters used for modeling of a bioreactor

The study is based on single biomass-single substrate process. The model equations are based on first principle. Given by Material Balance:

Rate of accumulation = inflow-outflow + generation - consumption.

The primary aim of a continuous bioreactor is to avoid wash out condition which ceases reaction. This may be done either by controlling cell mass or substrate concentrations. In order to maintain the reaction rate and product quality, both of them may be controlled with dilution rate and feed substrate concentration as manipulated variables. There are 4 number of control configurations possible which are as follows:

□ Dilution rate is used to control cell concentration and substrate concentration

□ Feed substrate concentration is used to control biomass or cell concentration and substrate concentration

The modeling equations of a bioreactor are given by equations 1&2

$$\frac{dx_1}{dt} = (\mu - D)x_1 \quad (1)$$

$$\frac{dx_2}{dt} = \frac{D(x_{2f} - x_2) - \mu x_1}{Y} \quad (2)$$

Where the state variables are  $x_1$ , the biomass concentration and  $x_2$ , the substrate concentration. The manipulated input is  $D$ , dilution rate and the disturbance input is  $x_{2f}$ , substrate feed concentration.

There are 2 possible solutions given by equations 3&4, which represents specific growth. They are Monod and substrate inhibition

$$\mu = \frac{\mu_{max} x_2}{K_m + x_2} \quad (3)$$

$$\mu = \frac{\mu_{max} x_2}{K_m + x_2 + K_{i1} x_2^2} \quad (4)$$

### 2.1 Dynamic behavior of a reactor

Table 1 shows the parameters to find the steady state conditions for the model shown by equations 1&2. The steady state dilution rate is  $D=0.3 \text{ hr}^{-1}$  and the feed substrate concentration is  $4.0 \text{ g/litre}$ .

Table 2 shows the operating conditions for a dilution rate of  $0.3 \text{ hr}^{-1}$ . Steady state condition 1 is a washout case since no reaction was occurred. Substrate concentration is the same as feed concentration.

Table 2: Operating conditions of bioreactor

S.No	Steady State	Biomass Concentration	Substrate Concentration	Stability
1	Equilibrium 1	$X_{1s}=0$	$X_{2s}=4.0$	stable
2	Equilibrium 2	$X_{1s}=0.995$	$X_{2s}=1.5122$	unstable
3	Equilibrium 3	$X_{1s}=1.53$	$X_{2s}=0.175$	stable

### 2.2 state space model of a reactor

The state space model matrices are

$$A = \begin{bmatrix} \mu - D_s & x_{1s} \mu_s \\ -\mu_s & -Dx - \frac{\mu_s x_{1s}}{Y} \end{bmatrix}$$

$$B = \frac{x_{1s}}{x_{2f} - x_{2s}}$$

$$C = [1 \quad 0]$$

$$D=0$$

$\mu_s$  represents the derivative of growth rate with respect to substrate concentration at steady state and given equation (5).

$$\mu_s = \frac{\partial \mu}{\partial x_{2s}} = \frac{\mu_{max} K_m}{(K_m + x_{2s})^2} \quad (5)$$

### 2.3 Stable operating point

The following initial condition is used for simulation

$$X(0) = \begin{bmatrix} 1.53 \\ 0.175 \end{bmatrix} \text{ .The state space model}$$

for the corresponding to stable operating point is

$$A = \begin{bmatrix} 0 & 0.9056 \\ -0.7500 & -2.564 \end{bmatrix}$$

$$B = \begin{bmatrix} -1.5301 \\ 3.8255 \end{bmatrix}$$

$$C = [1 \quad 0]$$

$$D=0$$

Eigen values are determined for the above matrix and its values are  $-0.3, -2.264 \text{ hr}^{-1}$ , so the system is stable. The transfer function relating the dilution rate to the biomass concentration is determined using Matlab and is given by equation 6.

$$Gp(s) = \frac{-1.5302s - 0.4590}{s^2 + 2.564s + 0.6792} \quad (6)$$

The discretized transfer function can be written as

$$Gp(z) = \frac{-0.1327z^{-1} + 0.1368z^{-2}}{1 - 1.768z^{-1} + 0.7738z^{-2}} \quad (6A)$$

## 3 System identification

System identification is an approach to obtain a mathematical model that reproduces for desired purposes and with enough exactitude, the dynamic characteristics of the process under study based on the observed (measured) variables of the process output signal or controlled variable  $y(t)$ , input signal or control variable  $u(t)$ , and in some cases disturbances  $v(t)$ . Figure 1 shows the block diagram of system identification

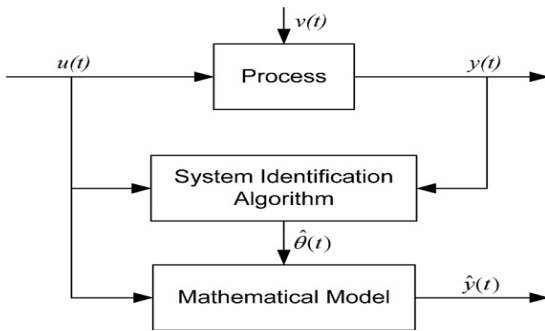


Figure 2: Block diagram of system identification

System identification is an integral part of any control design and deals with the problem of building reliable mathematical models of dynamic processes based on observed input–output data. The identification results determine the achieved control quality

### 3.1 Modelling principles

Identification of systems can be classified into three main groups, white-box, and grey-box and black-box identification

1. White-box modelling denotes a modelling technique where the model is fully derived based on the physical, chemical or biological properties of the system. The main advantage of white-box models is that they are not data dependent.
2. Grey-box modelling represents modelling a system based on a priori knowledge of the system. In this some physical information is used in the modeling process together with system identification techniques. This type of identification often results in models with high accuracy.
3. Black-box modelling is the alternative modelling technique where little a priori knowledge about the system is used in the modelling process and all the description of the system is strictly empirical and solely based on the collected data. One advantage of such methods is that many times the complexity of the models is reduced, but physical interpretation of the system is not available in the model.

The ARMAX model structure is

$$Y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b - 1) + C_1 u(t-1) + \dots + C_{n_c} u(t-n_c) + e(t) \quad (7)$$

The difference equation can be written as

$$A(q)y(t) = B(q)u(t-n_k) + C(q)e(t) \quad (8)$$

Where  $y(t)$  is output at time  $t$

$n_a$  is the number of Number of poles.

$n_b$  - Number of zeroes plus 1.

$n_c$  - Number of C coefficients.

$n_k$  - Number of input samples that occur before the input affects the output, also called the dead time in the system. For discrete systems with no dead time, there is a minimum 1–sample delay because the output depends on the previous input and  $n_k=1$

$y(t-1) \dots y(t-n_a)$  - Previous outputs on which the current output depends.

$u(t-1) \dots u(t-n_k-n_b+1)$  - Previous and delayed inputs on which the current output depends.

$e(t)$  - White-noise disturbance value.

The parameters  $n_a$ ,  $n_b$ , and  $n_c$  are the orders of the ARMAX model, and  $n_k$  is the delay.  $q$  is the delay operator. Specifically,

$$A(q) = 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}$$

$$B(q) = b_1 + b_2 q^{-1} + \dots + b_{n_b} q^{-n_b}$$

$$C(q) = 1 + c_1 q^{-1} + \dots + c_{n_c} q^{-n_c}$$

$$\text{From 6A, } A(q) = 1 - 1.768q^{-1} + 0.7738q^{-2}$$

$B(q) = -0.1327q^{-1} + 0.1368q^{-2}$  and  $C(q)$  is chosen arbitrarily and is given by

$$C(q) = 1 - 1.768q^{-1} + 0.7738q^{-2}$$

The recursive estimation functions include RPEM, RPLR, RARMAX, RARX, ROE, and RBJ. RPEM is the general Recursive Prediction Error algorithm for arbitrary multiple-input-single-output models. RPLR is the general Recursive Pseudo Linear Regression method for the same family of models. RARX is a more efficient version of RPEM (and RPLR) for the ARX-case. ROE, RARMAX and RBJ are more efficient versions of RPEM for the OE, ARMAX, and BJ cases. To implement these algorithms the following adaptation principle are used.

- i) Kalman filter approach: The true parameters are supposed to vary like a random walk with incremental covariance matrix  $P$
- ii) Forgetting factor approach: Old measurements are discounted exponentially. The base of the decay is the forgetting factor  $\lambda$ .
- iii) Gradient method: The update step is taken as a gradient step of length  $\gamma$ .
- iv) Normalized gradient method: The Gradient methods are also known as LMS (least mean squares) for the ARX case.

### 4. On line model identification

In many practical cases such as adaptive control, or where the system may change from day today, the model parameters are estimated online. Other motivations for online identification are given by including optimal control with model following, using matched filters, failure prediction etc. Because the identification is online, it must also be reasonably automatic. The algorithm must pick a suitable model form (number of dead times, order of the difference equation etc.), guess initial parameters and then calculate residuals and measures of fit. As time passes, number of data points that are continually collected will be more and better models can be built. The problem with using the offline identification scheme is that the data matrix,  $X$  in equation  $y_N = X_N \theta$  will grow and grow and more input/output rows have to be added to it. Eventually this matrix will grow too large to store or manipulate in our controller. There are two obvious solutions to this problem:

1. A sliding window can be used where only the last 50 input/output data pairs can be retained.
2. A recursive scheme may be used to achieve the same result but without wasting the old data.

#### 4.1 Recursive least-squares

As more data is collected, the current estimate of the model parameters can be updated. At every sample time, as the parameters are updated, estimation procedure is now no longer off-line, but now online. One approach to take advantage of the new data pair would be to add another row to the  $X$  matrix in Equation  $y_N = X_N \theta$  as each new data pair is collected. Then a new  $\theta$  estimate can be obtained using Equation 9.

$$\theta = (X_N^T X_N)^{-1} X_N^T Y_N \quad (9)$$

With the new augmented  $\mathbf{X}$  matrix. This equation would be solved every sample time. But  $\mathbf{C}$  matrix grows as each new data point is collected and so matrix inversion is required. The solution is to use a recursive formula for the estimation of the new  $\theta_{k+1}$  given the previous  $\theta_k$ .

#### 4.2 A recursive algorithm for least-squares

Suppose we have an estimate of the model parameters at time  $t = N$ ,  $\theta_N$ , perhaps calculated offline, and we subsequently collect a new data pair  $(u_{N+1}, y_{N+1})$ . To find the new improved parameters,  $\theta_{N+1}$ ,  $y_N = \mathbf{X}_N \theta$  is augmented by adding a new row and is given by the equation 10

$[y_N, y_{N-1}, \dots, y_{N-m+1}; u_{N+1}, u_N, \dots, u_{N-m+1}]$  (10)  
to the old matrix  $\mathbf{X}_N$ . So a new system is obtained and is given by equation 11

$$\varphi_{N+1}^T = \begin{bmatrix} y_N \\ \vdots \\ y_{N+1} \end{bmatrix} = \begin{bmatrix} X_N \\ \vdots \\ \varphi_{N+1}^T \end{bmatrix} \theta_N \quad (11)$$

Or  $y_{N+1} = X_{N+1} \theta_{N+1}$

So  $\theta_{N+1}$  is computed and is given by the equation 12

$$\varphi_{N+1} = (X_N^T X_{N+1})^{-1} X_{N+1}^T y_{N+1} \quad (12)$$

$$\varphi_{N+1} = \left( \begin{bmatrix} X_N^T \\ \vdots \\ \varphi_{N+1}^T \end{bmatrix} \right)^{-1} \begin{bmatrix} y_N \\ \vdots \\ y_{N+1} \end{bmatrix} = (X_N^T X_N + \varphi_{N+1} \varphi_{N+1}^T)^{-1} \begin{bmatrix} y_N \\ \vdots \\ y_{N+1} \end{bmatrix} \quad (13)$$

Using Matrix inversion lemma the above equation can be written as

$$\left( X_N^T X_N + \varphi_{N+1} \varphi_{N+1}^T \right)^{-1} = \left( X_N^T X_N \right)^{-1} - \frac{\left( X_N^T X_N \right)^{-1} \varphi_{N+1} \varphi_{N+1}^T \left( X_N^T X_N \right)^{-1}}{1 + \varphi_{N+1}^T \left( X_N^T X_N \right)^{-1} \varphi_{N+1}} \quad (14)$$

And the equation 13 can be rewritten as

$$\theta_{N+1} = \left( X_N^T X_N \right)^{-1} - \frac{\left( X_N^T X_N \right)^{-1} \varphi_{N+1} \varphi_{N+1}^T \left( X_N^T X_N \right)^{-1}}{1 + \varphi_{N+1}^T \left( X_N^T X_N \right)^{-1} \varphi_{N+1}} \left( X_N^T y_N + \varphi_{N+1}^T y_{N+1} \right) \quad (15)$$

A new parameter vector in terms of the old can be developed and is given by equation 15

$$\theta_{N+1} = \theta_N - \frac{\left( X_N^T X_N \right)^{-1} \varphi_{N+1} \varphi_{N+1}^T \theta_N}{1 + \varphi_{N+1}^T \left( X_N^T X_N \right)^{-1} \varphi_{N+1}} + \left( X_N^T X_N \right)^{-1} \varphi_{N+1}^T y_{N+1} - \frac{\left( X_N^T X_N \right)^{-1} \varphi_{N+1} \varphi_{N+1}^T \left( X_N^T X_N \right)^{-1} \varphi_{N+1}^T y_{N+1}}{1 + \varphi_{N+1}^T \left( X_N^T X_N \right)^{-1} \varphi_{N+1}} \quad (15)$$

$$= \theta_N + \frac{\left( X_N^T X_N \right)^{-1} \varphi_{N+1}^T}{1 + \varphi_{N+1}^T \left( X_N^T X_N \right)^{-1} \varphi_{N+1}} (y_{N+1} - \varphi_{N+1} \theta_N) \quad (16)$$

By defining the covariance matrix  $P$  as

$$P_N = \left( X_N^T X_N \right)^{-1} \quad (16)$$

And the parameter updated equation is

$$\theta_{N+1} = \theta_N + K_{N+1} (y_{N+1} - \varphi_{N+1}^T \theta_N) \quad (17)$$

With the gain  $K$  in equation and the new covariance given by equation (18).

$$K_{N+1} = \frac{P_N \varphi_{N+1}}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} \quad (18)$$

$$P_{N+1} = P_N \left[ I - \frac{\varphi_{N+1} \varphi_{N+1}^T P_N}{1 + \varphi_{N+1}^T P_N \varphi_{N+1}} \right] \quad (19)$$

$$\theta_{N+1}$$

updated equation 17 is nothing but that the new value of is the old value  $\theta_N$  with an added correction term which is a recursive form. The correction term is proportional to the observed error.

#### 4.3 Recursive least-squares estimation Algorithm

Initialise parameter vector,  $\theta_0$ , (say random values), and set co-variance to a large value,  $P_0 = 10^6 I$ , to reflect the initial uncertainty in the trial guesses

1. At sample  $N$ , collect a new input/output data pair  $(y_{N+1}$  and  $u_{N+1})$
2. Form the new  $\varphi_{N+1}$  vector by inserting  $u_{N+1}$ .
3. Evaluate the new gain matrix  $K_{N+1}$ , given in equation 18
4. Update the parameter vector  $\theta_N$  given in equation 17
5. Update the covariance matrix  $P_{N+1}$ , given in equation 19 which is required for the next iteration.
6. Wait out the remainder of one sample time  $T$ , increment sample counter,  $N \leftarrow N + 1$ , then go back to step 1.

#### 4.4 Kaczmarz's projection algorithm

The standard recursive least squares (RLS) algorithm requires two update equations, one to update gain, and one for the covariance matrix. A simplified algorithm can be implemented at the expense of the quality of parameter estimates. That method is known as Kaczmarz's projection algorithm, in which the input/output relation is assumed as  $y_k = \varphi_k \theta_k$  (20)

Where  $\theta$  is the parameter vector and  $\varphi$  is the row data vector and the correction to the estimated parameter vector is

$$\hat{\theta} = \hat{\theta}_{k-1} + \alpha \varphi_k^T \quad (21)$$

Where  $\alpha$  is chosen such that  $y_k = \varphi_k \hat{\theta}_k$  which gives the full updating formula given by equation (22).

$$\hat{\theta}_k = \varphi_{k-1} + \frac{\varphi_k^T}{\varphi_k \varphi_k^T} (y_k - \varphi_k \hat{\theta}_{k-1}) \quad (22)$$

This update scheme is sometimes modified to avoid potential problems when the parameter vector equals zero. Data is first generated using Kaczmarz's algorithm for identification in Matlab from an ARX plant, and then the data is processed pseudo-online to obtain the estimated parameters. The input/output data and parameters from this data are shown in Fig 6. The estimated parameters do not converge quickly to the true parameters like a full recursive least-squares algorithm.

#### 5. Simulation results

To illustrate some of the schemes mentioned above, an ARMAX model for a bioreactor process is chosen and some input-output data is generated as shown in figure 3. An Output-Error model is built for the plotted data. A second order model with one delay is used and a forgetting factor of  $\lambda = 0.98$  is applied. Figure 4 shows an output of OE model. Estimated parameters (solid lines) converge quickly to the true ones (dotted lines). A second order ARMAX model is simulated using Matlab program by the RPLR approach (i.e. ELS) with Kalman filter adaptation by assuming a parameter variance of 0.001. The simulated results are shown in figure 5. Estimated

parameters tries to converge quickly to the true ones at the 30<sup>th</sup> instant itself.

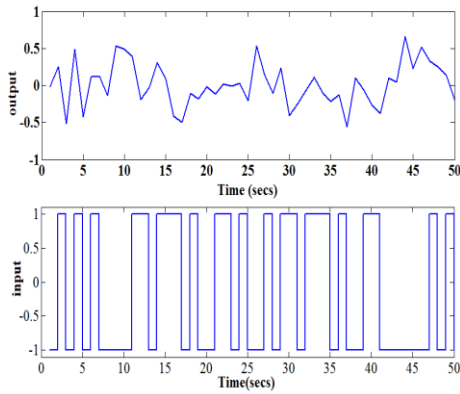


Figure3:Analysis of input/output

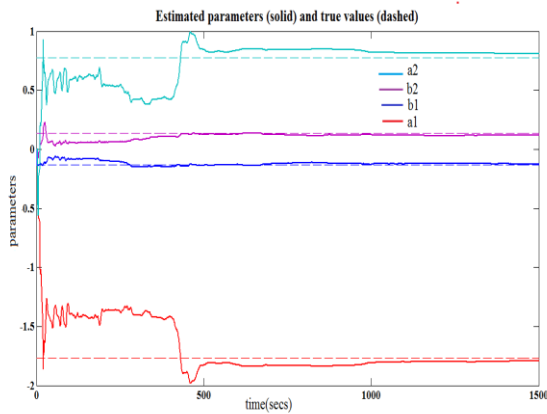


Figure 4: Output Error Model Estimation Using ROE

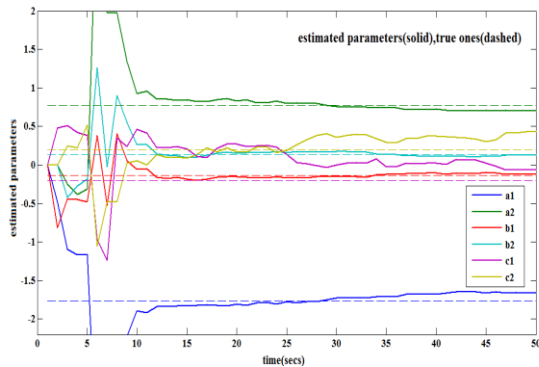


Figure 5: output of ARMAX Model Estimation using RPLRE approach

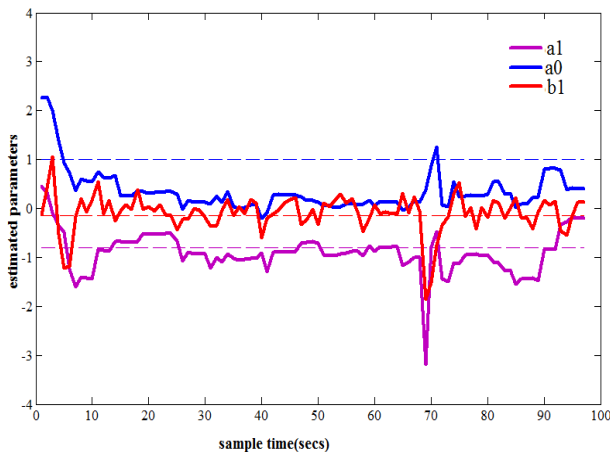


Figure 6: The performance of a simplified RLS algorithm, Kaczmarz's algorithm (estimated parameters (solid lines) and true parameters (dashed)).

## 6. Conclusion

Convergences of estimated parameters with the true ones are investigated with various model structures for a bioreactor process using Matlab. Output Error Model Estimation using ROE gave good results compared to the ARMAX model using RPLRE approach. Kaczmarz's algorithm does not converge to the true coordinates.

## References

1. Bo Bao, Yingqin Xua, Jie Sheng, Ruifeng Ding , “ Least squares based iterative parameter estimation algorithm for multivariable controlled ARMA system modelling with finite measurement data” Mathematical and Computer Modelling 53 (2011) 1664–1669
2. Jing Chen , Xianling Lu , Rui Ding , “Parameter identification of systems with preload nonlinearities based on the finite impulse response model and negative gradient search”, Applied Mathematics and Computation 219 (2012) 2498–2505
3. Rui Ding, Honghong Duan, “TS-RLS algorithm for pseudo-linear regressive models” 2012 American Control Conference Fairmont Queen Elizabeth, Montréal, Canada, June 27-June 29, 2012.
4. Cheng Wang, Tao Tang, “Recursive least squares estimation algorithm applied to a class of linear-in-parameters output error moving average systems”
5. Lincheng Zhou, Xiangli Li, Feng Pan, “Gradient-based iterative identification for MISO Wiener nonlinear systems: Application to a glutamate fermentation process”, Applied Mathematics Letters 26 (2013) 886–892.
6. Heqiang Han, Li Xie, Feng Ding Xinggao Liu , “Hierarchical least-squares based iterative identification for multivariable systems with moving average noises” Mathematical and Computer Modeling 51 (2010) 1213–1220
7. Junxia Ma, Rui Ding, “Recursive computational formulas of the least square criterion functions for scalar system identification” Applied Mathematical Modelling 38 (2014) 1–11.
8. Feng Ding, “Decomposition based fast least squares algorithm for output error systems” Signal Processing 93(2013)1235–1242.