EED604E OPTIMIZATION HOMEWORK 1

Dr. Öğr. Üyesi Mohammed ALKRUNZ

Istanbul Aydin University

Engineering Faculty, Electrical and Electronics Department

2024-2025 Fall Semester

Hüseyin KURT

PARAMETER ESTIMATION ALGORITHMS

1. Introduction

There are many algorithms for optimization of the parameter in the control system modeling. The most used algorithms are Recursive Least Square (RLS), Extended Least Square (ELS), Least Mean Square (LMS), Project Approximation (PA) and Stochastic Algorithm (SA). In this study, 5 algorithms was compared each other with an example. The simulation program used is MATLAB. In the RLS, the estimated parameters are updated recursively to minimize sum of squared differences between true and estimated values. ELS is extension of least square algorithm for nonlinear systems. PA projects a vector onto a constrained subspace to solve estimation problems.

Table 1: Compare the algorithms.

Algorithms	**Adaptivity**	**Complexity**	**Applications**
RLS	High	Moderate	Time-Varying Systems Real-Time System
ELS	Nonlinear	High	Nonlinear Systems
LMS	Simple	Low	Adaptive Filtering
PA	Constrained	Variable	Signal Processing, Optimization
SA	Stochastic	Low to Moderate	Machine Learning

Table shows comparing with these algorithms. LMS has low complexity but the adaptation is simple. However, RLS has high adaptation despite having moderate complexity.

2. Methodology

Optimization is an arrangement of the parameters for designing model according to measuring data from the physical system.

$$y(t) = a * y(t-1) + b * u(t-1) + c * e(t-1) + e(t) \tag{1}$$

Example model used in this study is shown in equation (1). It uses just 1 previous input and output values.

$$\varphi(t) = \begin{bmatrix} -y(t-1) \\ u(t-1) \\ e(t-1) \end{bmatrix} \tag{2}$$

$$\theta = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{3}$$

Equation (2) shows the regression vector and equation (3) is shows the parameters estimated.

$$y(t) = \varphi(t)'\theta + \ e(t) \tag{4}$$

Equation 4 is regression model. There is an error in this equation.

$$e(t) = y(t) - \varphi(t)' * \hat{\theta}(t-1) \tag{5}$$

The error indicates difference between true value (observed) and estimated value as shown in equation (5).

$$\hat{\theta}(t) = \hat{\theta}(t-1) + k(t) * e(t) \tag{6}$$

Equation (6) demonstrates parameter update. It has an updating cofactor (*k*). The algorithms mentioned above handle with this cofactor by means of different manner. The equation of these algorithms are following:

- Recursive Least Square:

$$k(t) = [p(t-1) - \frac{p(t-1)*\varphi(t)*\varphi(t)'*p(t-1)}{1+\varphi(t)'*p(t-1)*\varphi(t)}] * \varphi(t) \tag{7}$$

- Least Mean Square:

$$k(t) = \gamma * \varphi(t) \tag{8}$$

- Extended Least Square:

$$k(t) = \frac{P(t-1)*\varphi(t)}{1+\varphi(t)'*P(t-1)*\varphi(t)} \tag{9}$$

- Projection Approximation:

$$k(t) = \frac{\gamma*\varphi(t)}{\alpha+\varphi(t)'*\varphi(t)} \tag{10}$$

- Stochastic Algorithm:

$$k(t) = (\sum_i^t \varphi(t)' * \varphi(t))^{-1} * \varphi(t) \tag{11}$$

The algorithm are different each other and they are suitable according to requirements which are used subject.

3. Simulation and Results

Using the previous algorithm updating cofactor, the simulation is performed for each algorithm in MATLAB Script. In the equations, there are pre-defined parameters (P(0) and Theta(0)).

$$P(0) = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix} \tag{12}$$

$$\hat{\theta}(0) = [0\ 0\ 0]' \tag{13}$$

The used covariance matrix and beginning estimated value used in this study shows in equation (12) and (13). The step signal was used as a input signal (u(t)).

Table 2: Parameter initializing of the simulation.

```
N = 1000;
a_desired = 1;
b_desired = 0.5;
c_desired = -0.5;
u = ones(N, 1);
y = zeros(N, 1);
E = load("e.mat");
e = E(1).e;
theta_hat = zeros(3, 1);
theta_store = zeros(3, N);
P = 100 * eye(3);
phi = zeros(3, 1);
%Mathematical model of the system
for t = 2:N
    y(t) = -a_desired * y(t-1) + b_desired * u(t-1) + c_desired * e(t-
1) + e(t);
end
```

Table 3: Recursive Least Square algorithm code.

```
for t = 2:N
    phi = [-y(t-1); u(t-1); e(t-1)];
    error = y(t) - phi' * theta_hat;
    P = P - (P*phi*phi'*P)/(1 + phi'*P*phi);
    K = P*phi;
    theta_hat = theta_hat + K * error;
    theta_store(:, t) = theta_hat;
end
```

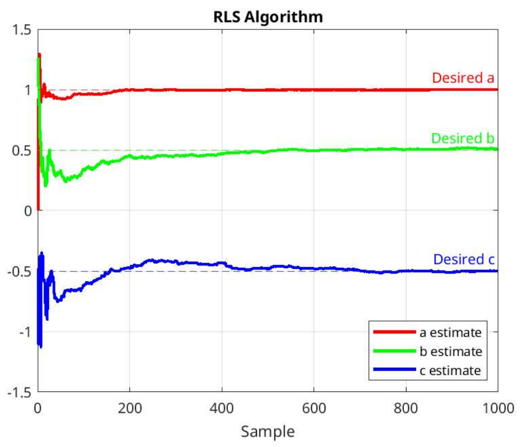


Figure 1: Recursive Least Square algorithm simulation graph.

Table 4: Least Mean Square algorithm code.

```
gamma_lms = 0.01;
for t = 2:N
    phi = [-y(t-1); u(t-1); e(t-1)];
    error = y(t) - phi' * theta_hat;
    K = (gamma_ lms*phi);
    theta_hat = theta_hat + K*error;
    theta_store(:, t) = theta_hat;
end
```

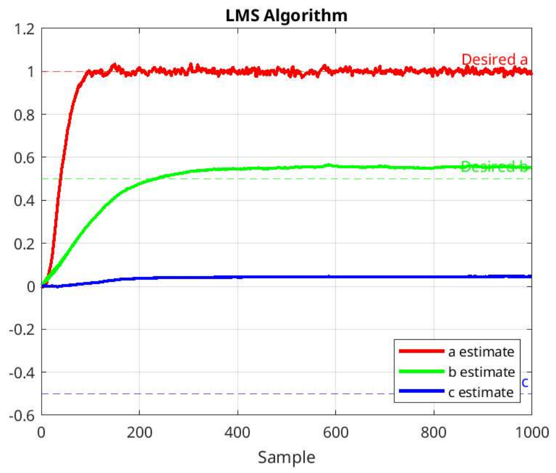


Figure 2: Least Mean Square algorithm simulation graph.

Table 5: Extended Least Square algorithm code.

```
for t = 2:N
    phi = [-y(t-1); u(t-1); e(t-1)];
    error = y(t) - phi' * theta_hat;
    M = (P*phi)/(1 + phi'*P*phi);
    P = P - (P*phi*phi'*P) / (1 + phi'*P*phi);
    theta_hat = theta_hat + M * error;
    theta_store(:, t) = theta_hat;
end
```

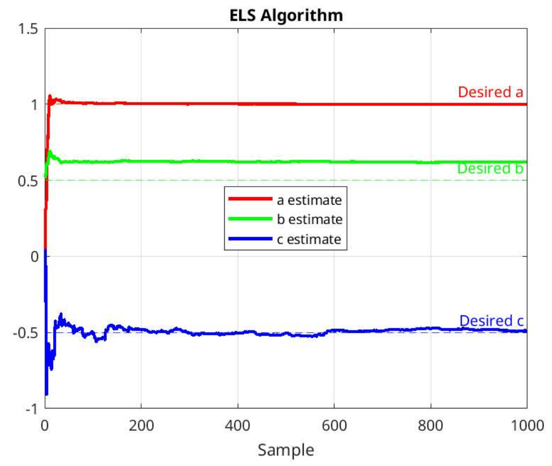


Figure 3: Extended Least Square algorithm simulation graph.

Table 6: Projection Approximation algorithm code.

```
for t = 2:N
    phi = [-y(t-1); u(t-1); e(t-1)];
    error = y(t) - phi' * theta_hat;
    K = (gamma_pa*phi)/(alpha_pa + phi'*phi);
    theta_hat = theta_hat + K*error;
    theta_store(:, t) = theta_hat;
end
```

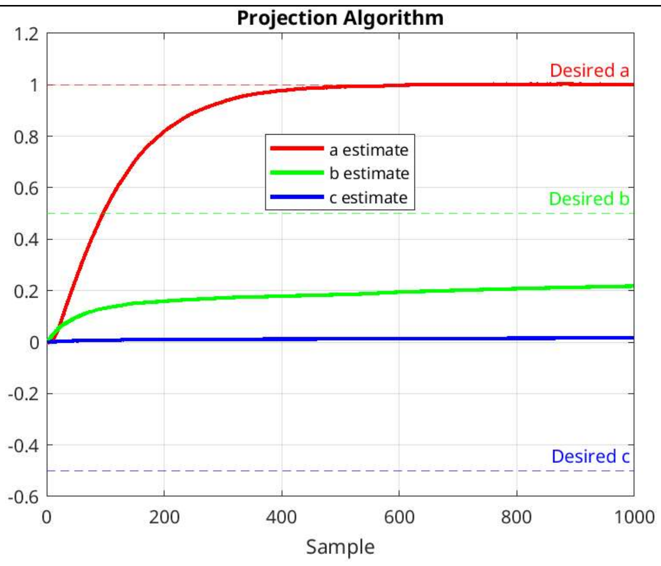


Figure 4: Projection Approximation algorithm simulation graph.

Table 7: Stochastic algorithm code.

```
alpha_pa = 0.1;
gamma_pa = 0.01;
P = 0;
for t = 2:N
    phi = [-y(t-1); u(t-1); e(t-1)];
    error = y(t) - phi' * theta_hat;
    P = (P + phi'*phi)^-1;
    K = P*phi;
    theta_hat = theta_hat + K*error;
    theta_store(:, t) = theta_hat;
end
```

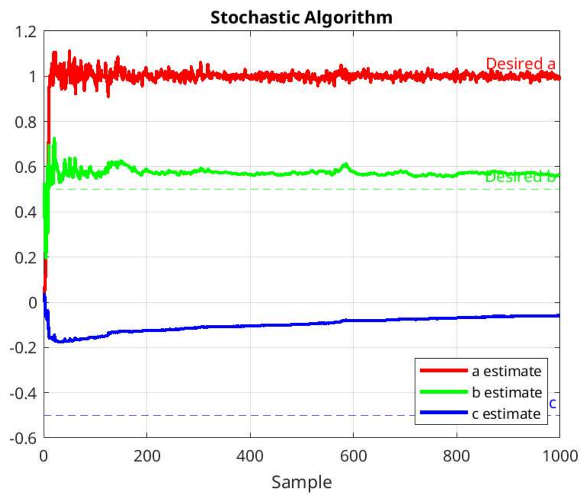


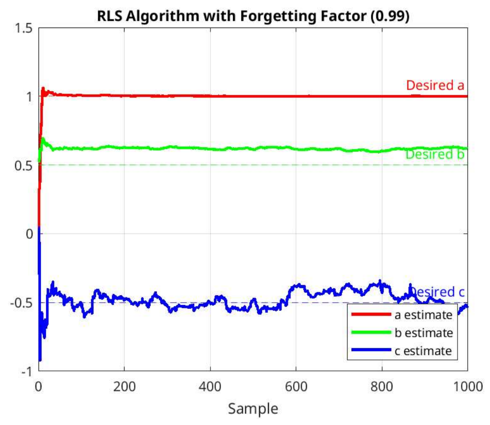
Figure 5: Stochastic algorithm simulation graph.

$$k(t) = \frac{1}{l} * [p(t-1) - \frac{p(t-1) * \varphi(t) * \varphi(t)' * p(t-1)}{1 + \varphi(t)' * p(t-1) * \varphi(t)}] * \varphi(t) \quad (14)$$

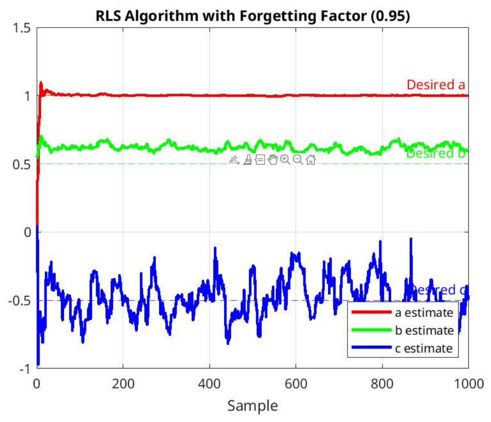
In the algorithm with training such as machine learning, the low forgetting factor in model cause memorizing the model. Therefore, the model can behave misleadingly and cannot adapted to new conditions. As seen in equation (14) decreasing the factor leads to increasing oscillation in error.

Table 8: RLS algorithm code with the forgetting factor.

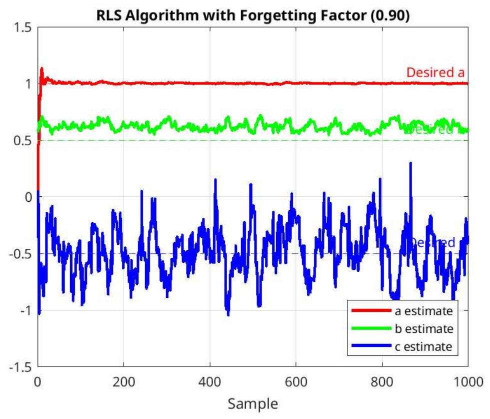
```
forgetting_factor = 0.99;
for t = 2:N
    phi = [-y(t-1); u(t-1); e(t-1)];
    error = y(t) - phi' * theta_hat;
    P = (1 / forgetting_factor) * (P - (P*phi*phi'*P)/(1 + 
phi'*P*phi));
    K = P*phi;
    theta_hat = theta_hat + K * error;
    theta_store(:,t) = theta_hat;
end
```



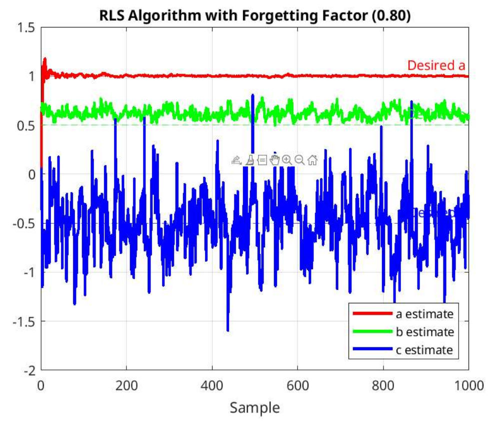
(a)



(b)



(c)



(d)

Figure 6: RLS with forgetting factor.

The figure 6 demonstrates the RLS algorithm simulation graphs with forgetting factor respectively 0.99, 0.95, 0.90 and 0.80. As the graphs indicate, decreasing the forgetting factor leads to inadequate estimation achievement.

- **RLS Closed-Loop Feedback**

The simulation performed above uses input unit step function. The closed loop input simulation shows in this section.

$$u(t) = -0.2 * y(t) \tag{15}$$

*Table 9: Closed-Loop feedback code with 0.2*y(t).*

```
for t = 2:N
    u(t) = -0.2 * y(t);
    y(t) = -a_desired * y(t-1) + b_desired * u(t-1) + c_desired * e(t-
1) + e(t);
end
```

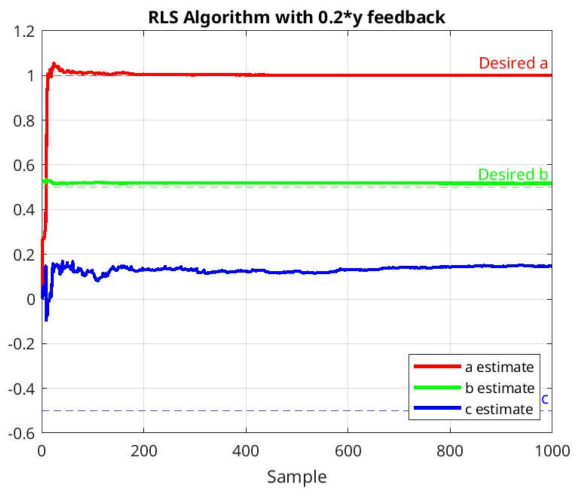


*Figure 7: RLS algorithm with 0.2*y(t) closed loop feedback.*

The closed loop feedback using 0.2*y(t) input signal provides the better result for parameter a and b, but parameter c cannot reach to true value.

$$u(t) = -0.32 * y(t-1) \tag{16}$$

*Table 10: Closed-Loop feedback code with 0.32*y(t-1).*

```
for t = 2:N
    u(t)  = -0.32 * y(t-1);
    y(t)  = -a_desired * y(t-1) + b_desired * u(t-1) + c_desired * e(t-
1) + e(t);
end
```

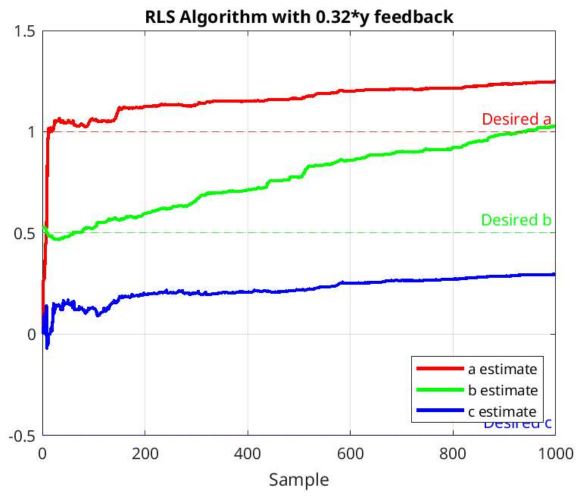


*Figure 8: RLS algorithm with 0.32*y(t-1) closed loop feedback.*

The equation (16) demonstrates the closed loop feedback including previous output value. The control give rise to deviate the estimation values from true values.

4. Conclusion

The modelling physical system is very important to control it. Many algorithms are developed according to requirements and where it uses. Mostly uses are RLS, ELS, LMS, PA and SA. In this study, these algorithms are compared in MATLAB simulation program by using an example. For this example, RLS gives the better result. Closed loop feedback control with current output is better than closed loop feedback using previous output value.