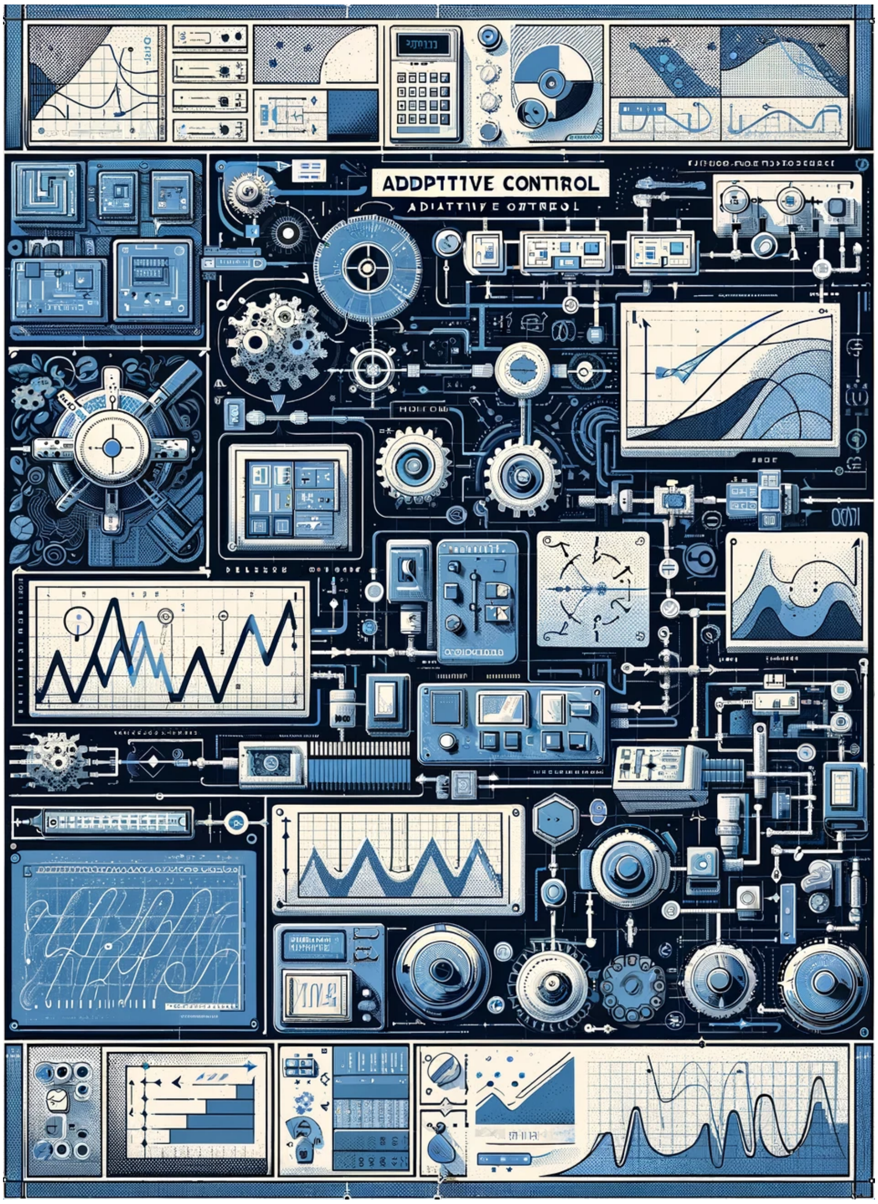
Simulation 1 Adaptive Control

Student Name: Amirhossein Ehsani
Student ID: 810602159
Instructor: Dr. Ayati



April 17, 2024

Contents

List of Figures

List of Tables

1 Offline System Identification

A mass, spring and damper system is shown as follows:

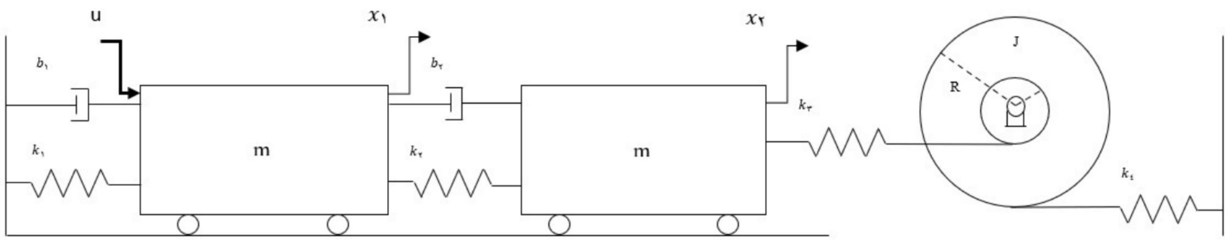


Figure 1: Mass, Spring and Damper System

All masses are equal to m.

Take the small pulley radius r and the large disk radius R.

take the parameters as bellow:(We can change the parameters in case of unstable response)

$$\begin{cases} k_1 = 0.6 \\ k_2 = 0.4 \\ k_3 = 0.4 \\ k_4 = 0.133 \\ b_1 = 0.6 \\ b_2 = 0.4 \end{cases}$$

1.1 State Space and Transfer Function Calculation

The dynamic equations of the system is as follows:

$$\begin{cases} m\ddot{x}_1 = -b_1\dot{x}_1 - k_1x_1 - b_2(\dot{x}_2 - \dot{x}_1) - k_2(x_2 - x_1) + u \\ m\ddot{x}_2 = b_2(\dot{x}_2 - \dot{x}_1) + k_2(x_2 - x_1) - k_3(x_3 - x_2) \\ J\frac{\ddot{x}_3}{r} = k_3(x_3 - x_2)r + k_4(x_3 - x_2)R \end{cases}$$

The state space form can be written as:

$$\begin{cases} \dot{x} = Ax + Bu \\ Y = Cx + Du \end{cases}$$

We will define the state variables as bellow:

$$X = \begin{bmatrix} x_1 \\ \dot{x}_1 \\ x_2 \\ \dot{x}_2 \\ x_3 \\ \dot{x}_3 \end{bmatrix}$$

The state spaces are obtained as bellow:

$$\begin{cases} A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{-k_1+k_2}{m} & \frac{-b_1+b_2}{m} & \frac{-k_2}{m} & \frac{-b_2}{m} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{-k_2}{m} & \frac{-b_2}{m} & \frac{k_2+k_3}{m} & \frac{b_2}{m} & \frac{-k_3}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-R^2k_4-k_3r^2}{J} & 0 & \frac{R^2k_4+k_3r^2}{J} & 0 \end{bmatrix} & B = \begin{bmatrix} 0 \\ \frac{1}{m} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} & D = 0 \end{cases}$$

The parameter given by the question could achieve stability in the step response of the system but the system was very slow and it took about 500 seconds for the steady state zero to reach zero. In order to get a stable step response with faster behavior, we choose the parameters as bellow:

$$\begin{cases} k_1 = 600 & k_2 = 400 \\ k_3 = 400 & k_4 = 133 \\ b_1 = 25 & b_2 = 20 \\ R = 0.2 & r = 0.1 \\ m = 1 & J = \frac{1}{2}mR^2 \end{cases}$$

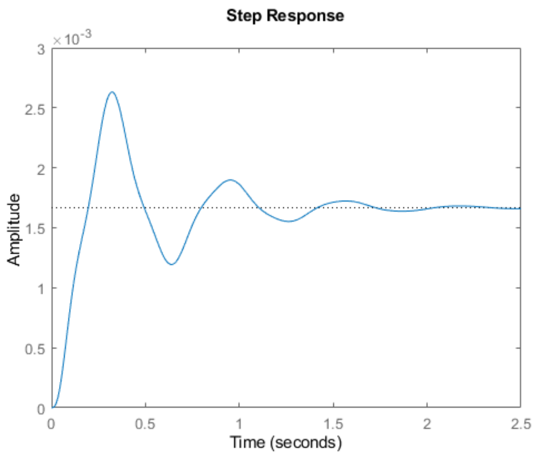


Figure 2: Step Response of the system

As we can see in figure above the step response of the system is stable. The continuous time transfer function of the system is:

$$G(s) = \frac{20s^3 + 400s^2 + 9320s + 1.864E05}{s^6 + 65s^5 + 2766s^4 + 7.029E04s^3 + 1.525E06s^2 + 1.025E07s + 1.118E08}$$

1.2 Discrete State Space and Transfer Function

In order to find the discrete transfer function and state space first we have to choose an appropriate sampling time. To find the sampling time we need to first find the highest frequency in the system which is also known as bandwidth frequency. The bandwidth frequency is where -3 dB from the steady frequency. The calculation process is shown as follows:

```
% Calculate the frequency response using Bode plot data
[mag, phase, w] = bode(G);

% Convert magnitude from absolute to dB
magdB = 20*log10(squeeze(mag));

% Find the -3 dB point, which is the bandwidth of the system
refdB = magdB(1);
minus3dB = refdB - 3;
bwIndex = find(magdB <= minus3dB, 1, 'first');

% Bandwidth frequency in rad/s and converting to Hz
bwFrequencyRad = w(bwIndex);
bwFrequencyHz = bwFrequencyRad / (2*pi);

% Calculating Ts using Shanon theoreom
Ts = 1/(10*bwFrequencyHz);
```

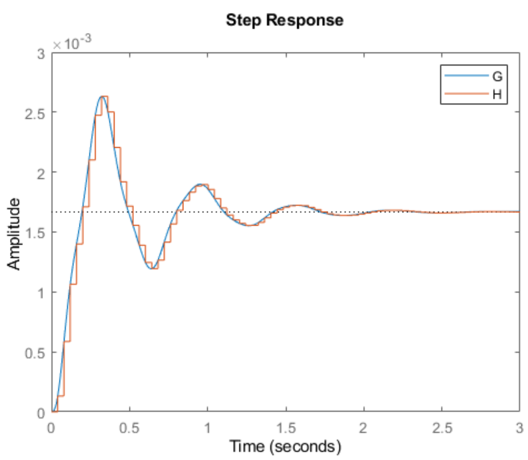


Figure 3: Step Response of the discrete system

As we can see in figure 2 above the chosen sampling time is good. The discrete transfer function and the discrete state space is obtained using c2d command in MATLAB as follows:

$$H(z) = \frac{0.0001308z^5 + 8.696E - 5z^4 - 0.0002903z^3 + 0.0003485z^2 - 5.855E - 5z - 2.354E - 5}{z^6 - 2.82z^5 + 3.957z^4 - 3.394z^3 + 1.796z^2 - 0.4957z + 0.07364}$$

1.3 Estimating System Using LS Method

first we need to write the difference equation of the system. and choose a model, in this part we will choose a model with same order of our system:

$$H(z) = \frac{b_1z^5 + b_2z^4 + b_3z^3 + b_4z^2 + b_5z + b_6}{z^6 + a_1z^5 + a_2z^4 + a_3z^3 + a_4z^2 + a_5z + a_6}$$

Now we divide both numerator and denumerator by z^6:

$$\rightarrow H(z) = \frac{b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4} + b_5z^{-5} + b_6z^{-6}}{1 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + a_4z^{-2} + a_5z^{-5} + a_6z^{-6}}$$

Now we can write the difference equations as bellow: (We will add a white noise with mean equal to zero and variance of 0.15 to our system)

```
% We will also creat a white noise for the system
var = 0.15;
e = sqrt(var)*randn(N,1);
e = e-mean(e);

% The first 6 signals must be created manually
y(1) = e(1);
y(2) = a1*y(1)+b1*u(1)+e(2);
y(3) = a1*y(2)+a2*y(1)+b1*u(2)+b2*u(1)+e(3);
y(4) = a1*y(3)+a2*y(2)+a3*y(1)+b1*u(3)+b2*u(2)+b3*u(1)+e(4);
y(5) = a1*y(4)+a2*y(3)+a3*y(2)+a4*y(1)+b1*u(4)+b2*u(3)+b3*u(2)+b4*u(1)+e
  (5);
y(6) = a1*y(5)+a2*y(4)+a3*y(3)+a4*y(2)+a5*y(1)+b1*u(5)+b2*u(4)+b3*u(3)+
  b4*u(2)+b5*u(1)+e(5);

for i = 7:N
  y(i) = a1*y(i-1)+a2*y(i-2)+a3*y(i-3)+a4*y(i-4)+a5*y(i-5)+a6*y(i-6)+...
    b1*u(i-1)+b2*u(i-2)+b3*u(i-3)+b4*u(i-4)+b5*u(i-5)+b6*u(i-6);
end
```

Now we know that LS algorithm is as follows:

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T * Y$$

Where the θ is the matrix of our parameters a_1 to a_6 and b_1 to b_6. And Φ is the matrix of our data. All this procedure is done as follows:

```
%% Estimation using LS method (the same order as the system)

Phi = [0 0 0 0 0 0 0 0 0 0 0 0;
y(1) 0 0 0 0 0 u(1) 0 0 0 0 0;
y(2) y(1) 0 0 0 0 u(2) u(1) 0 0 0 0;
y(3) y(2) y(1) 0 0 0 u(3) u(2) u(1) 0 0 0;
y(4) y(3) y(2) y(1) 0 0 u(4) u(3) u(2) u(1) 0 0;
y(5) y(4) y(3) y(2) y(1) 0 u(5) u(4) u(3) u(2) u(1) 0;
y(6:N-1) y(5:N-2) y(4:N-3) y(3:N-4) y(2:N-5) y(1:N-6) u(6:N-1) u(5:N-2)
 u(4:N-3) u(3:N-4) u(2:N-5) u(1:N-6)];

T_LS = inv(Phi'*Phi)*Phi'*y;
```

1.4 Results of The Model with Same Order (White Noise Input)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
Estimated Parameters	2.7661 -0.0009993	-3.6518 -0.0011111	2.6241 0.00099026	-0.89409 0.0023932	-0.061168 0.0023462	0.06857 -0.00017657

Table 1: Comparison of Real System and Least Square Estimation

As we can see most of the parameters are well estimated specially the a_i coefficients. Further in the next sections we will study the importance of PE order of the input signal in the estimation process.

1.5 Estimating the System Using Impulse Input

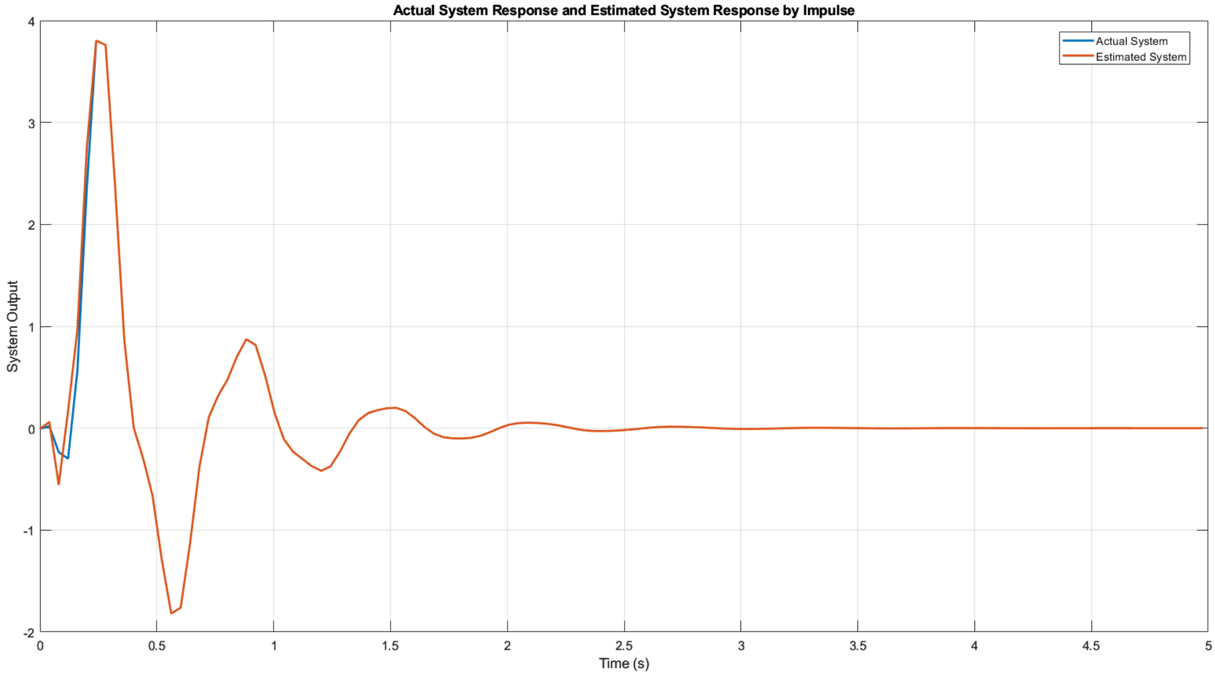


Figure 4: Step Response of the Real system vs. Estimated system (Impulse Input)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
Estimated Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.0406	-0.32115	0.46994	0.40109	0.40109	-5.8553e-05

Table 2: Comparison of Real System and Least Square Estimation (Impulse Input)

1.6 Estimating the System Using Step Input

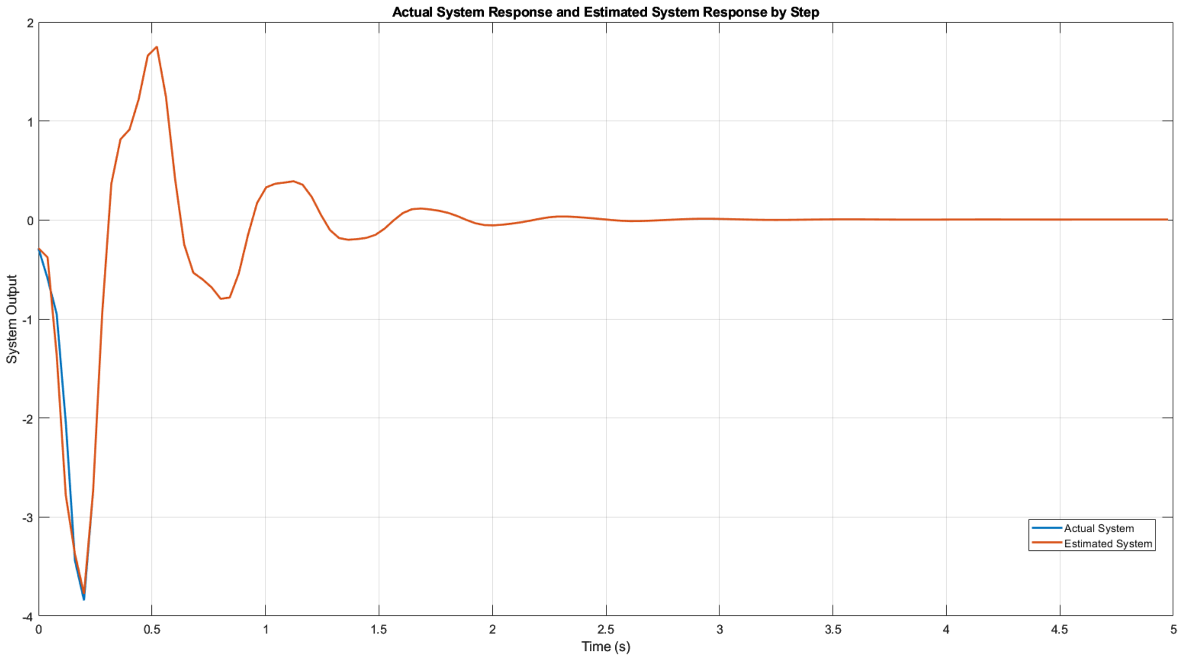


Figure 5: Step Response of the Real system vs. Estimated system (Step Input)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
Estimated Parameters	2.8201 0.21382	-3.957 -0.62536	3.3943 -0.32597	-1.7958 0.80281	0.49571 0.00034849	-0.073642 -0.06508

Table 3: Comparison of Real System and Least Square Estimation (Step Input)

1.7 Estimating the System Using Sinusoidal Input

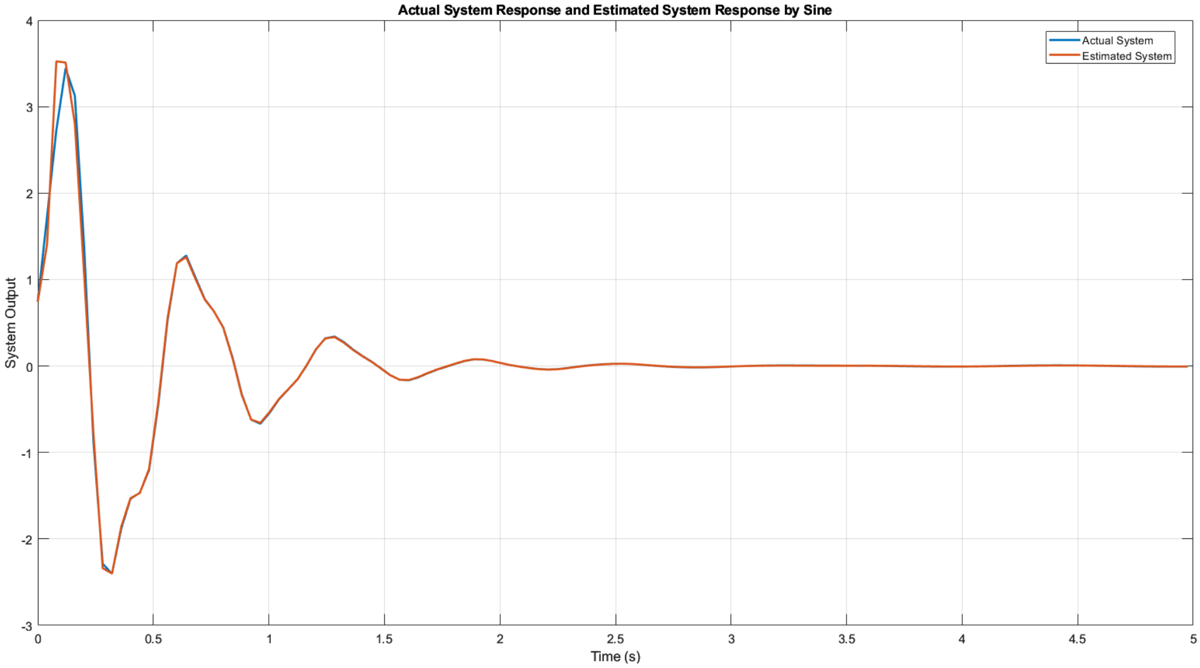


Figure 6: Step Response of the Real system vs. Estimated system (Sinusoidal Input)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
Estimated Parameters	2.389 2.3158	-2.661 -2.7878	1.484 -2.2561	-0.12057 2.2324	-0.36514 2.4907	0.11718 -1.9462

Table 4: Comparison of Real System and Least Square Estimation (Sinusoidal Input)

1.8 Estimating the System Using Sum of Two Sinusoidal Input

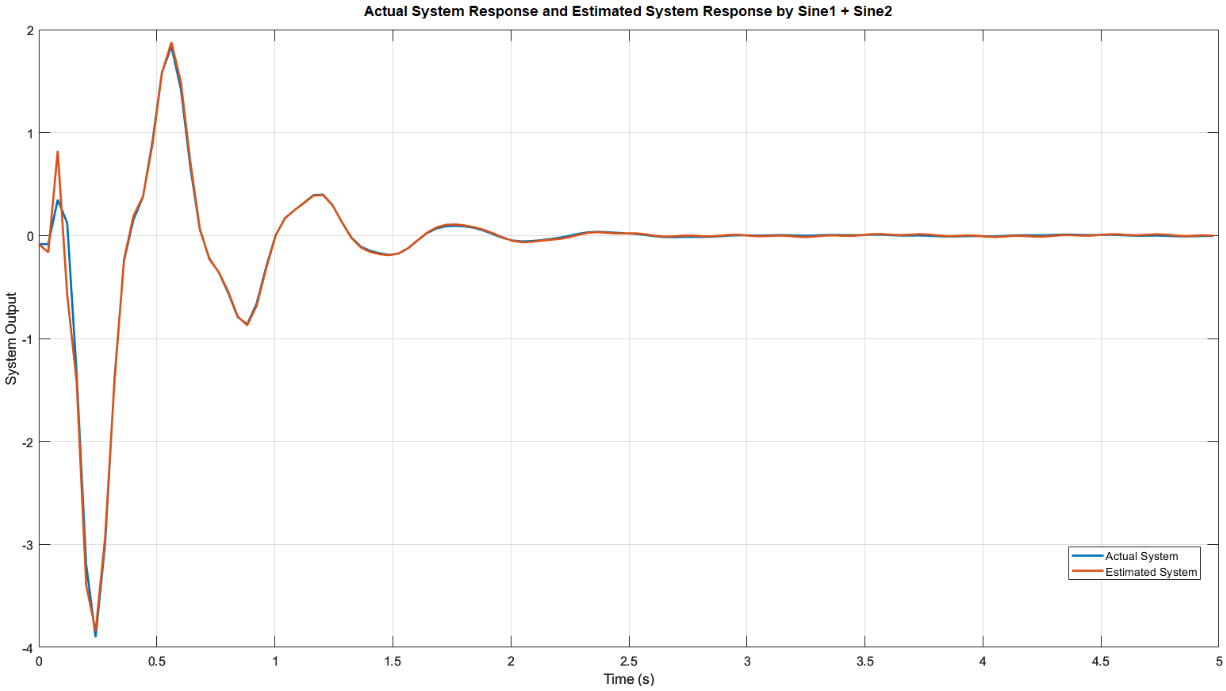


Figure 7: Step Response of the Real system vs. Estimated system (Sum of Two Sinusoidal Input)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
Estimated Parameters	3.7593	-7.0306	8.1437	-6.152	2.8626	-0.66462
	0.24266	-0.94582	1.5788	-1.616	1.0289	-0.29963

Table 5: Comparison of Real System and Least Square Estimation (Sum of Two Sinusoidal Input)

1.9 Estimating the System Using Ramp Input

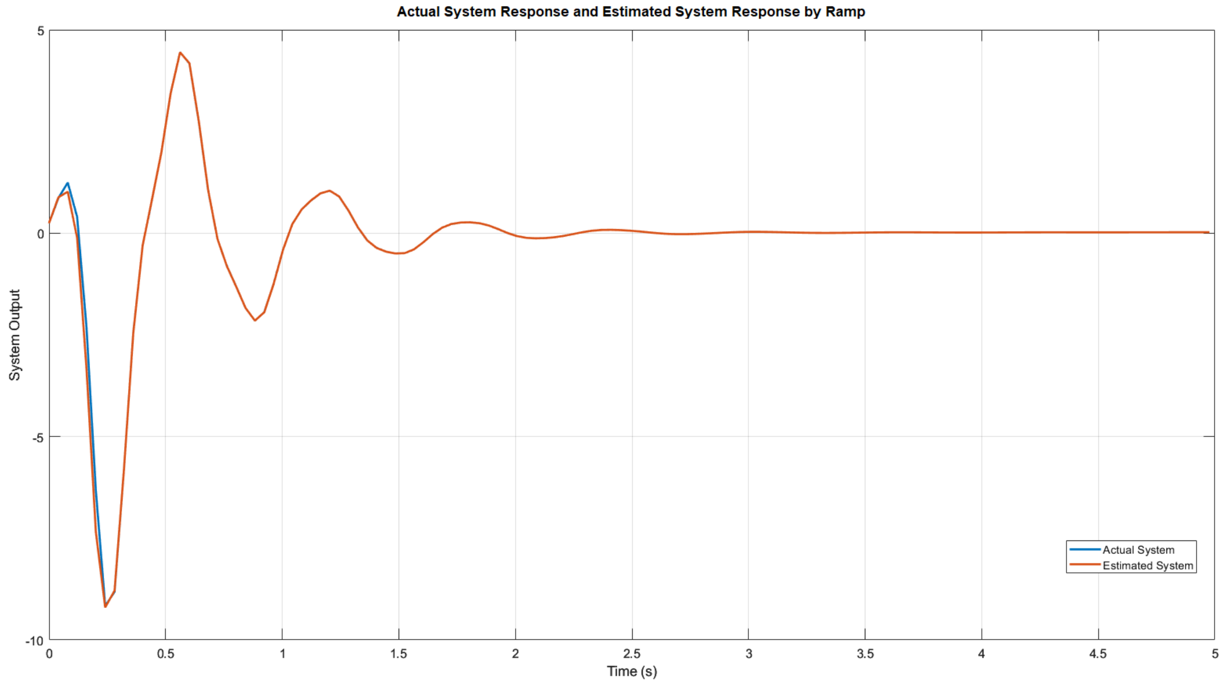


Figure 8: Step Response of the Real system vs. Estimated system (Ramp Input)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
Estimated Parameters	2.8833 -5.6856	-4.1587 -0.96978	3.7003 -5.6206	-2.0691 12.772	0.63837 23.627	-0.10678 -24.122

Table 6: Comparison of Real System and Least Square Estimation (Ramp Input)

1.10 Estimating the System Using White Noise Input

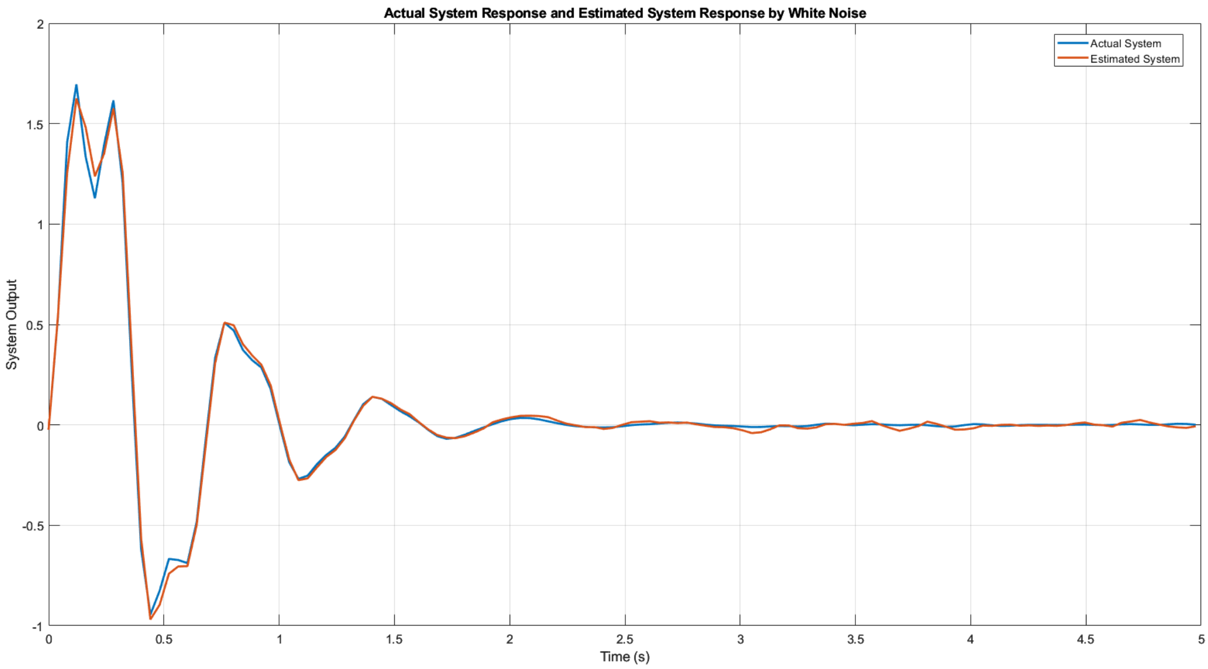


Figure 9: Step Response of the Real system vs. Estimated system (White Noise Input)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
Estimated Parameters	2.5488 0.0044072	-3.4191 0.0047721	3.0749 0.0050897	-1.8164 0.0024229	0.67687 0.00076009	-0.19988 -0.00069022

Table 7: Comparison of Real System and Least Square Estimation (White Noise Input)

1.11 Estimating the System Using Colored Noise Input

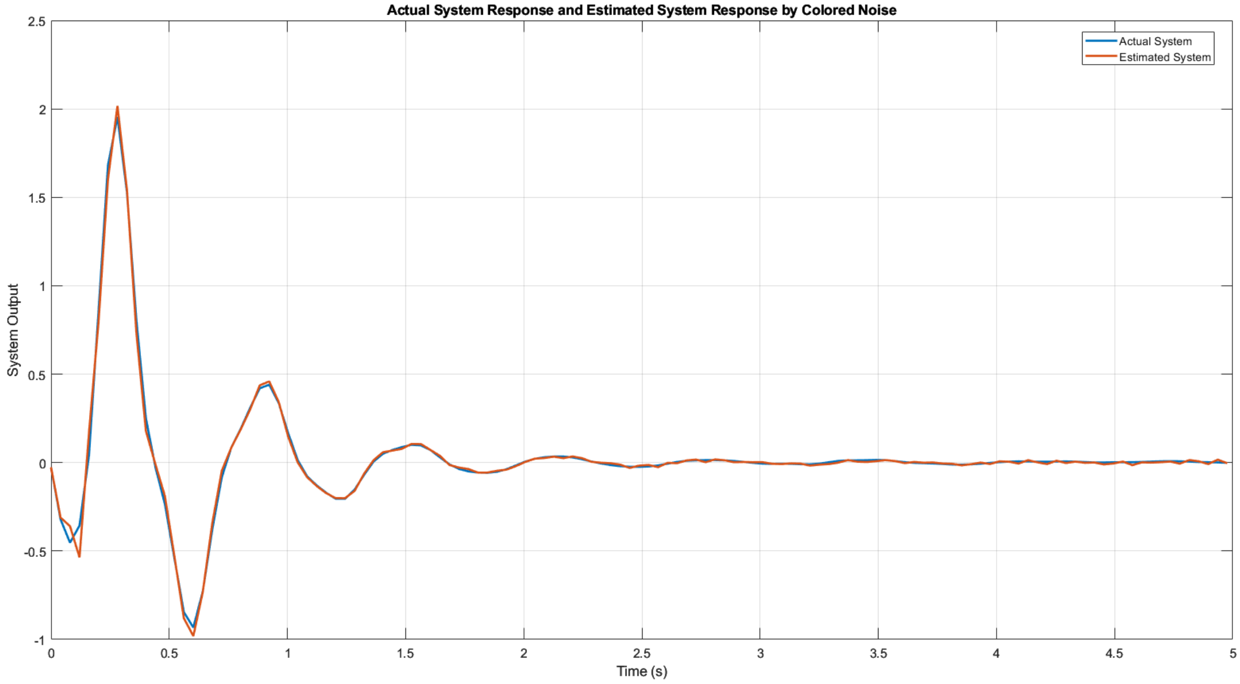


Figure 10: Step Response of the Real system vs. Estimated system (Colored Noise Input)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
Estimated Parameters	2.4495 -0.00017149	-2.7269 -0.0044982	1.5427 0.0021907	-0.33526 -0.00072226	-0.075859 -0.0019799	-0.012987 0.0001263

Table 8: Comparison of Real System and Least Square Estimation (Colored Noise Input)

1.12 Conclusion on the Effect of the PE Order of the Input Signal

As observed, most inputs provide good estimates for the coefficients of $y(t - i)$. However, the estimation of $u(t - i)$ is less reliable for signals with a lower Predominant Excitation (PE) order.

Upon closer examination, it appears that white noise yields a slightly better estimate than colored noise. This difference, albeit small, is because the Least Squares (LS) method is optimized for white noise. For colored noise, methods like Generalized Least Squares (GLS) or Best Linear Unbiased Estimator (BLUE) are more appropriate.

In summary, while LS is generally effective for system identification, its accuracy decreases with lower PE order signals and colored noise. In such cases, switching to methods like GLS or BLUE may provide more accurate estimates.

1.13 Studying Effect of The Model's Order

In this section we will estimate the system using a model with order different than the real model.

1.13.1 Under Parameter

In order to estimate the system we will choose a model as bellow:

$$y(t) = a_1y(t-1)+a_2y(t-2)+a_3y(t-3)+a_4y(t-4)+b_1u(t-1)+b_2u(t-2)+b_3u(t-3)+b_4u(t-4)+e(t)$$

This model has 8 parameter with order 4. The Φ matrix is created as follows:

```
%% Under Parameter order 4
Phi = [0 0 0 0 0 0 0 0;
y(1) 0 0 0 u(1) 0 0 0;
y(2) y(1) 0 0 u(2) u(1) 0 0;
y(3) y(2) y(1) 0 u(3) u(2) u(1) 0;
y(4:N-1) y(3:N-2) y(2:N-3) y(1:N-4) u(4:N-1) u(3:N-2) u(2:N-3) u(1:N-4)];
```

And the RLS algorithm is performed as bellow:

```
  T_ls = inv(Phi'*Phi)*Phi'*y;
  y_LS = zeros(N,1);
  y_LS(1) = e(1);
  y_LS(2) = T_ls(1)*y_LS(1)+T_ls(5)*u(1)+e(2);
  y_LS(3) = T_ls(1)*y_LS(2)+T_ls(2)*y_LS(1)+T_ls(5)*u(2)+T_ls(6)*u(1)+e(3)
   ;
  y_LS(4) = T_ls(1)*y_LS(3)+T_ls(2)*y_LS(2)+T_ls(3)*y_LS(1)+T_ls(5)*u(3)+
   T_ls(6)*u(2)+T_ls(7)*u(1)+e(4);
  for i = 5:N
  y_LS(i) = T_ls(1)*y_LS(i-1)+T_ls(2)*y_LS(i-2)+T_ls(3)*y_LS(i-3)+T_ls(4)*
   y_LS(i-4)+...
  T_ls(5)*u(i-1)+T_ls(6)*u(i-2)+T_ls(7)*u(i-3)+T_ls(8)*u(i-4)+e(i);
  end

  E = y-y_LS';
  JJ=0.5*(E'*E);
```

We can plot the output of the under parameter system and compare it with the original system's output as shown in figure 11. Also we will choose white noise input to eliminate any concerns about the PE order of the input.

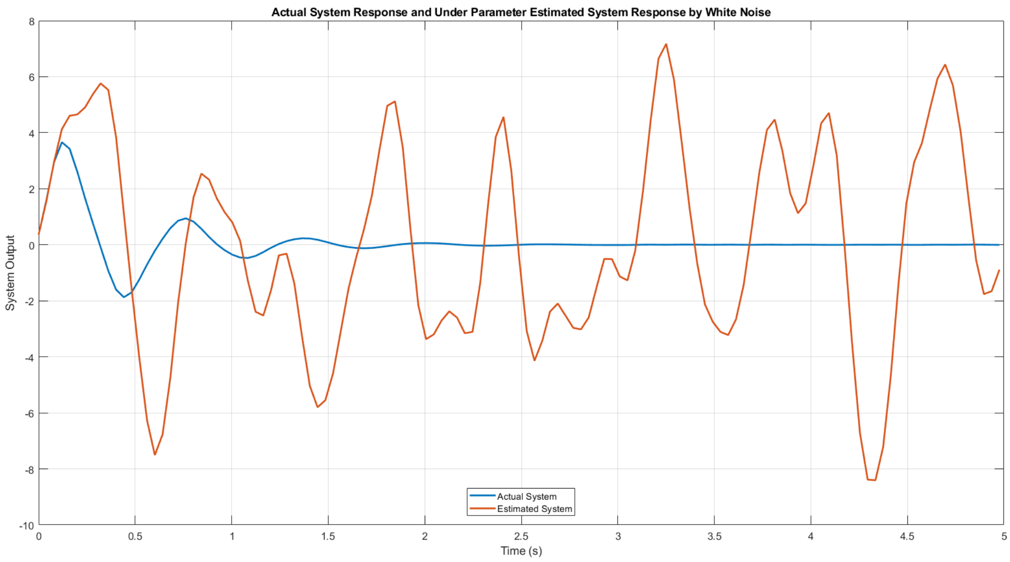


Figure 11: Step Response of the Real system vs. Under Parameter Estimated system

As it is demonstrated in the figure the estimation is not a good estimation.

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4
Estimated Parameters	2.9285 0.0082349	-4.2913 -0.0013074	3.9083 -0.0029454	-2.2609 -0.00070003

Table 9: Under Parameter System's Least Square Estimation (White Noise Input)

1.13.2 Over Parameter

In order to estimate the system we will choose a model as bellow:

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + a_3 y(t-3) + a_4 y(t-4) + a_5 y(t-5) + a_6 y(t-6) + a_7 y(t-7)$$

$$+ b_1 u(t-1) + b_2 u(t-2) + b_3 u(t-3) + b_4 u(t-4) + b_5 u(t-5) + b_6 u(t-6) + b_7 u(t-7) + e(t)$$

This model has 14 parameter with order 7. The Φ matrix is created as follows:

```
%% Over parameter order 7

Phi = [0 0 0 0 0 0 0 0 0 0 0 0 0 0;
y(1) 0 0 0 0 0 0 u(1) 0 0 0 0 0 0;
y(2) y(1) 0 0 0 0 0 u(2) u(1) 0 0 0 0 0;
y(3) y(2) y(1) 0 0 0 0 u(3) u(2) u(1) 0 0 0 0;
y(4) y(3) y(2) y(1) 0 0 0 u(4) u(3) u(2) u(1) 0 0 0;
y(5) y(4) y(3) y(2) y(1) 0 0 u(5) u(4) u(3) u(2) u(1) 0 0;
y(6) y(5) y(4) y(3) y(2) y(1) 0 u(6) u(5) u(4) u(3) u(2) u(1) 0;
y(7:N-1) y(6:N-2) y(5:N-3) y(4:N-4) y(3:N-5) y(2:N-6) y(1:N-7) u(7:N-1) u
    (6:N-2) u(5:N-3) u(4:N-4) u(3:N-5) u(2:N-6) u(1:N-7)];
```

And the RLS algorithm is performed as bellow:

```
T_LS = inv(Phi'*Phi)*Phi'*y;
y_LS = zeros(N,1);
y_LS(1) = e(1);
y_LS(2) = T_LS(1)*y(1)+T_LS(8)*u(1)+e(2);
y_LS(3) = T_LS(1)*y(2)+T_LS(2)*y(1)+T_LS(8)*u(2)+T_LS(9)*u(1)+e(3);
y_LS(4) = T_LS(1)*y(3)+T_LS(2)*y(2)+T_LS(3)*y(1)+T_LS(8)*u(3)+T_LS(9)*u(2)
    +T_LS(10)*u(1)+e(4);
y_LS(5) = T_LS(1)*y(4)+T_LS(2)*y(3)+T_LS(3)*y(2)+T_LS(4)*y(1)+T_LS(8)*u(4)
    +T_LS(9)*u(3)+T_LS(10)*u(2)+T_LS(11)*u(1)+e(5);
y_LS(6) = T_LS(1)*y(5)+T_LS(2)*y(4)+T_LS(3)*y(3)+T_LS(4)*y(2)+T_LS(5)*y(1)
    +T_LS(8)*u(5)+T_LS(9)*u(4)+T_LS(10)*u(3)+T_LS(11)*u(2)+T_LS(12)*u(1)+e
    (6);
y_LS(7) = T_LS(1)*y(6)+T_LS(2)*y(5)+T_LS(3)*y(4)+T_LS(4)*y(3)+T_LS(5)*y(2)
    +T_LS(6)*y(1)+T_LS(8)*u(6)+T_LS(9)*u(5)+T_LS(10)*u(4)+T_LS(11)*u(3)+
    T_LS(12)*u(2)+T_LS(13)*u(1)+e(7);
for i = 8:N
y_LS(i) = T_LS(1)*y(i-1)+T_LS(2)*y(i-2)+T_LS(3)*y(i-3)+T_LS(4)*y(i-4)+T_LS
    (5)*y(i-5)+T_LS(6)*y(i-6)+T_LS(7)*y(i-7)+...
```

```
+T_LS(8)*u(i-1)+T_LS(9)*u(i-2)+T_LS(10)*u(i-3)+T_LS(11)*u(i-4)+T_LS(12)*u(
    i-5)+T_LS(13)*u(i-6)+T_LS(14)*u(i-7)+e(i);
end
```

We can plot the output of the under parameter system and compare it with the original system's output as shown in figure 11. Also we will choose white noise input to eliminate any concerns about the PE order of the input.

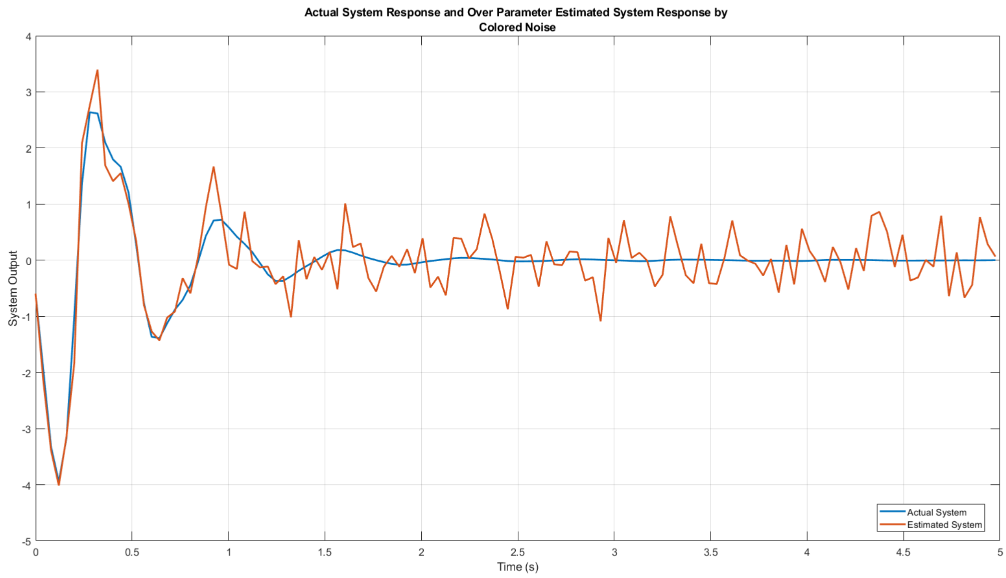


Figure 12: Step Response of the Real system vs. Over Parameter Estimated system

As it is demonstrated in the figure the estimation is not a very good estimation. But it is better than under parameter model.

Name	a_1	a_2	a_3	a_4	a_5	a_6
	a_7					
	b_1	b_2	b_3	b_4	b_5	b_6
	b_7					
Estimated Parameters	2.8693	-4.4898	4.734	-3.6234	2.0438	-0.8631
	0.18118					
	0.18118	-0.0030243	0.002078	0.0023935	-0.0022553	1.7191e-05
	0.00047849					

Table 10: Over Parameter System's Least Square Estimation (White Noise Input)

2 Online System Identification

In this part the given system is the same system as the previous section. The dynamic equations of the system is as follows:

$$\begin{cases} m\ddot{x}_1 = -b_1\dot{x}_1 - k_1x_1 - b_2(\dot{x}_2 - \dot{x}_1) - k_2(x_2 - x_1) + u \\ m\ddot{x}_2 = b_2(\dot{x}_2 - \dot{x}_1) + k_2(x_2 - x_1) - k_3(x_3 - x_2) \\ J\frac{\ddot{x}_3}{r} = k_3(x_3 - x_2)r + k_4(x_3 - x_2)R \end{cases}$$

The state space form can be written as:

$$\begin{cases} \dot{x} = Ax + Bu \\ Y = Cx + Du \end{cases}$$

The state spaces are obtained as bellow:

$$\begin{cases} A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{-k_1+k_2}{m} & \frac{-b_1+b_2}{m} & \frac{-k_2}{m} & \frac{-b_2}{m} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{-k_2}{m} & \frac{-b_2}{m} & \frac{k_2+k_3}{m} & \frac{b_2}{m} & \frac{-k_3}{m} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-R^2k_4-k_3r^2}{J} & 0 & \frac{R^2k_4+k_3r^2}{J} & 0 \end{bmatrix} & B = \begin{bmatrix} 0 \\ \frac{1}{m} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} & D = 0 \end{cases}$$

$$\rightarrow H(z) = \frac{b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4} + b_5z^{-5} + b_6z^{-6}}{1 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + a_4z^{-2} + a_5z^{-5} + a_6z^{-6}}$$

2.1 RLS Estimation Method (No Noise)

2.1.1 Impulse Input (P=10000, $\theta_0 = 0$)

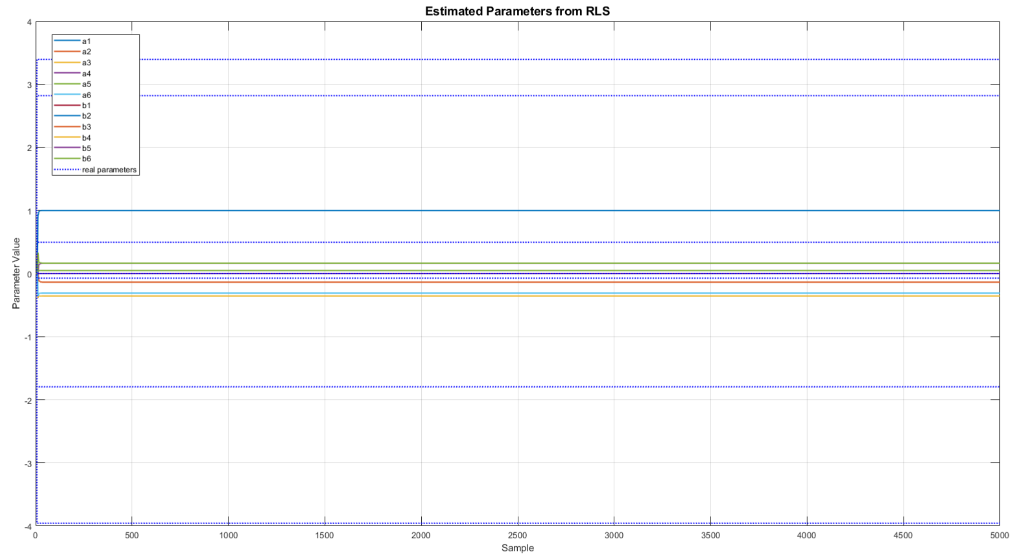


Figure 13: Estimated Parameters Using Impulse Input (P=10000, $\theta_0 = 0$)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(RLS)	1.1767 0	-0.34079 0	-0.42189 0	0.35211 0	0.090123 0	-0.29559 0.06268

Table 11: Real System and RLS Estimation with Impulse Input (P=10000, $\theta_0 = 0$)

As we can see the estimation is very bad and the reason is that impulse input is not a good input for estimating a system with such parameters.

2.1.2 Step Input (P=10000, $\theta_0 = 0$)

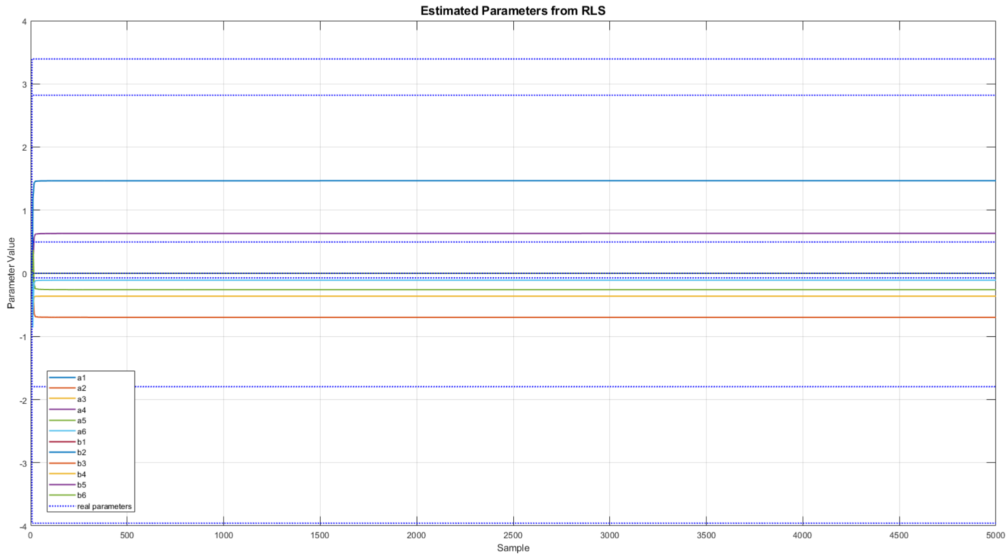


Figure 14: Estimated Parameters Using Step Input (P=10000, $\theta_0 = 0$)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(RLS)	1.466 0.00026908	-0.69779 0.00026908	-0.36243 0.00026908	0.63175 0.00026908	-0.26017 0.00026908	-0.1088 0.00026908

Table 12: Real System and RLS Estimation with Step Input (P=10000, $\theta_0 = 0$)

The estimation is bad and the reason is that step input is not a good input for estimating a system with such parameters. However it's better than impulse input but not quite satisfactory.

2.1.3 Sinusoidal Input (P=10000, $\theta_0 = 0$)

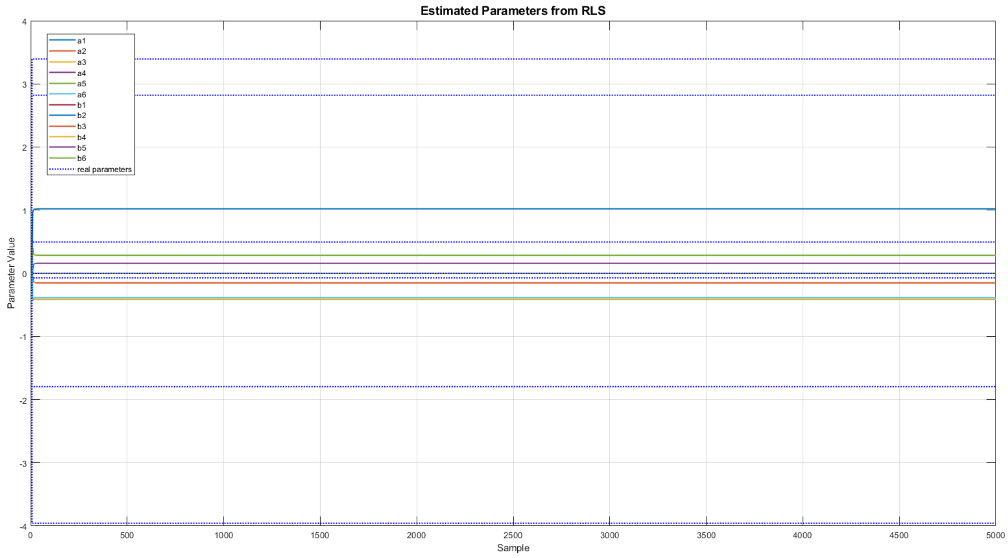


Figure 15: Estimated Parameters Using Sinusoidal Input (P=10000, $\theta_0 = 0$)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(RLS)	1.0213 -6.7785e-14	-0.15161 -6.7785e-14	-0.41397 -6.7785e-14	0.15862 -6.7785e-14	0.28484 -6.7785e-14	-0.38664 -6.7785e-14

Table 13: Real System and RLS Estimation with Sinusoidal Input (P=10000, $\theta_0 = 0$)

The estimation is still bad and the reason is that sinusoidal input is not a good again very good because of lack in PE order for estimating a system with such parameters. However it's better than impulse and step input but not quite satisfactory.

2.1.4 White Noise Input (P=10000, $\theta_0 = 0$)

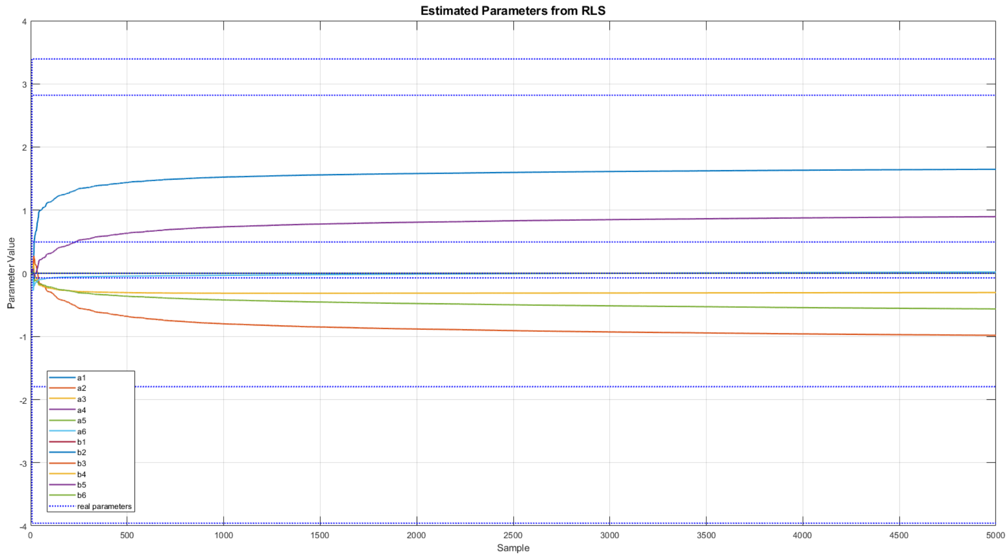


Figure 16: Estimated Parameters Using White Noise Input (P=10000, $\theta_0 = 0$)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(RLS)	1.6471 0.00013201	-0.98213 0.00024163	-0.30496 0.00014514	0.8967 3.4952e-05	-0.56423 0.00065568	0.020619 0.00046033

Table 14: Real System and RLS Estimation with White Noise Input (P=10000, $\theta_0 = 0$)

The estimation is acceptable and the reason is that White Noise input is good PE order for estimating a system with such parameters is enough. However it's better when the system has a noise which will be explained in the next section.

2.2 Conclusion on PE order of the input signal

The type of input signal used in system identification is key because it affects how well we can understand and model a system's behavior. Different input signals provide different kinds of information by exciting the system in various ways.

- White Noise Inputs: White noise is typically the best choice for system identification because it contains all frequencies. This ensures that every part of the system's behavior across the entire frequency range is excited, leading to a thorough and accurate model.

- Sine Wave Inputs: Sine waves are useful for examining specific frequencies within a system by changing the sine wave's frequency during tests. This method can be very informative but might take longer since each frequency is tested one at a time.

- Step Inputs: Step inputs are easy to use and good for seeing how the system reacts to a sudden change. This helps in understanding the system's overall stability and response time. However, they don't provide much information about how the system behaves at different frequencies.

- Impulse Inputs: Impulse inputs send a quick, sharp burst, theoretically exciting all frequencies at once. This can be useful for a quick overview of the system, but practical issues often make it less effective for detailed analysis compared to white noise.

To sum up things the white noise input is the best signal to estimate the system because it contains all frequencies.

2.3 Studying The Effect of Initial Values (P and θ_0) on RLS Algorithm Performance Under White Noise Input

2.3.1 $\theta_0 = 10$ and P=10000

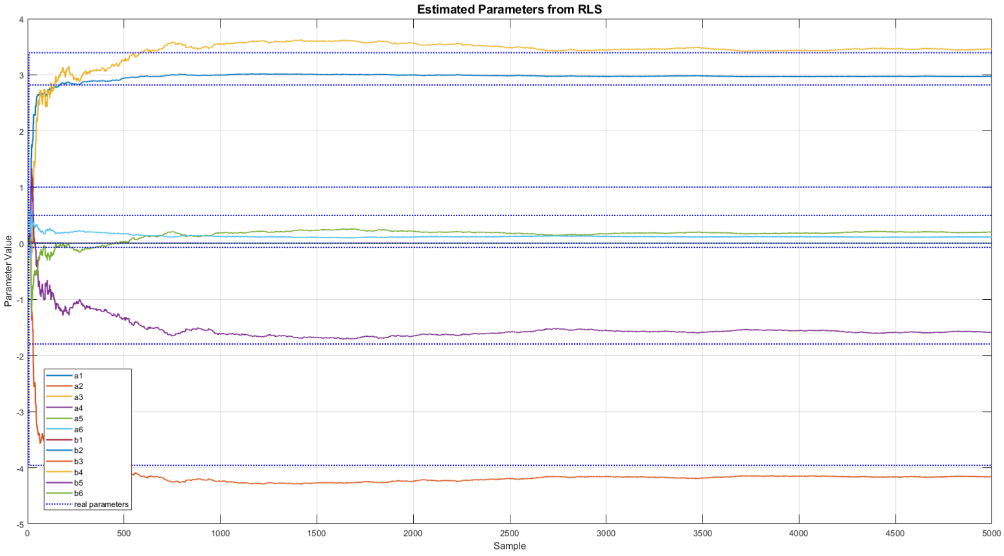


Figure 17: Estimated Parameters Using White Noise Input (P=10000, $\theta_0 = 10$) with white noise

Name	a_1 / b_1	a_2 / b_2	a_3 / b_3	a_4 / b_4	a_5 / b_5	a_6 / b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
(RLS)	2.9725	-4.1618	3.4615	-1.5876	0.19868	0.10966
	0.00039837	-7.8617e-05	-0.00015987	0.00050002	0.00037666	-0.00019399

Table 15: Real System and RLS Estimation with White Noise Input (P=10000, $\theta_0 = 10$) with white noise

2.3.2 $\theta_0 = 100$ and P=10000

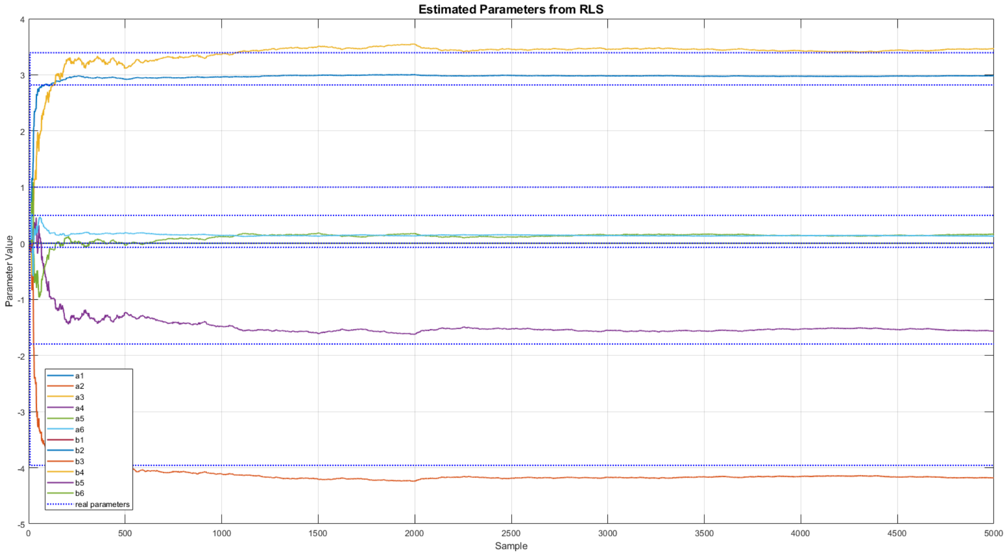


Figure 18: Estimated Parameters Using White Noise Input (P=10000, $\theta_0 = 100$) with white noise

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(RLS)	2.9829 0.00015946	-4.1817 -0.00026394	3.4686 -0.00044268	-1.568 0.00073887	0.16453 2.3794e-05	0.12677 -0.00011449

Table 16: Real System and RLS Estimation with White Noise Input (P=10000, $\theta_0 = 100$) with white noise

2.3.3 $\theta_0 = 1000$ and P=10000

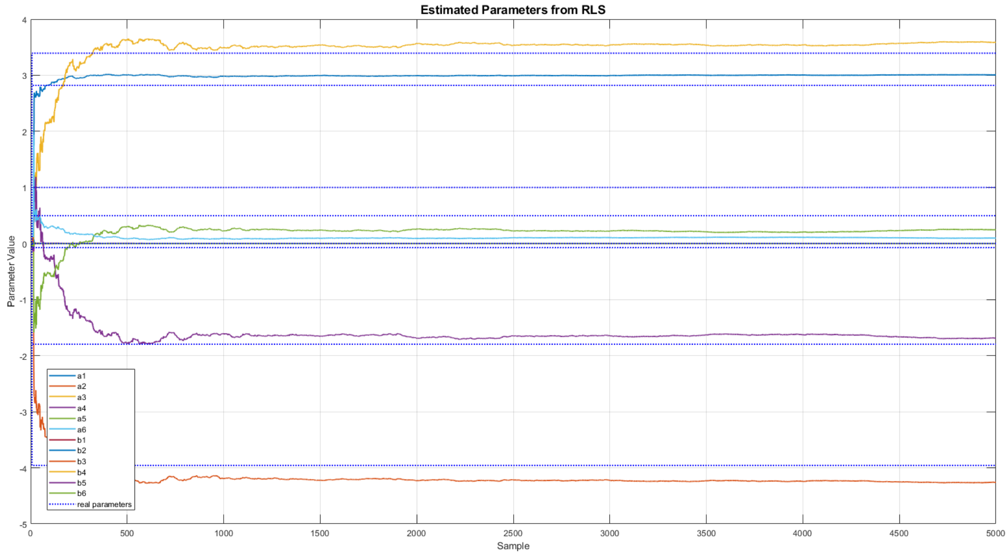


Figure 19: Estimated Parameters Using White Noise Input (P=10000, θ_0 = 1000) with white noise

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(RLS)	3.0078 0.00040306	-4.2592 -5.6456e-05	3.5877 -0.00044279	-1.6865 -0.00020792	0.24624 0.00014501	0.096725 -0.00031749

Table 17: Real System and RLS Estimation with White Noise Input (P=10000, $\theta_0 = 1000$) with white noise

2.3.4 $\theta_0 = 0$ and P=1000

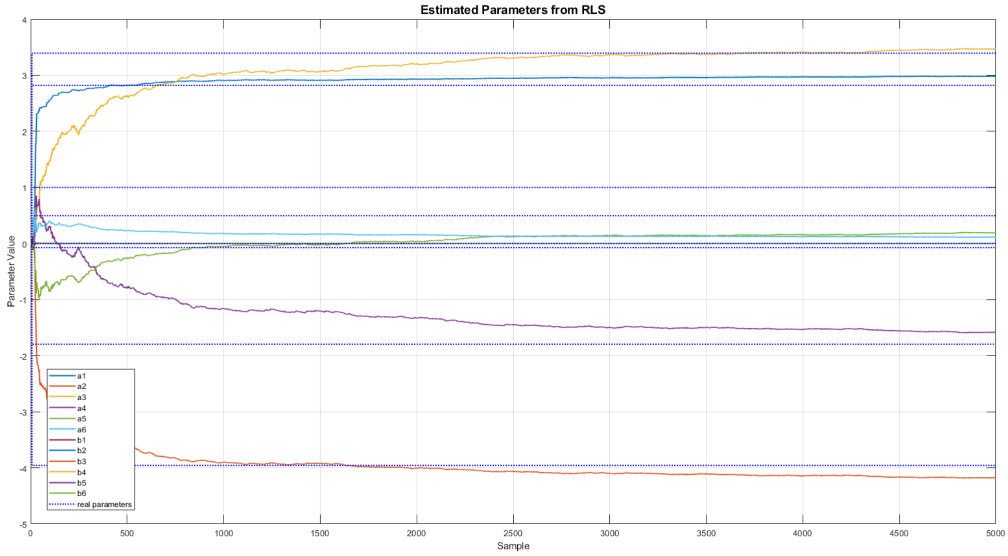


Figure 20: Estimated Parameters Using White Noise Input (P=1000, $\theta_0 = 0$) with white noise

Name	a_1 / b_1	a_2 / b_2	a_3 / b_3	a_4 / b_4	a_5 / b_5	a_6 / b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
(RLS)	2.9813	-4.1784	3.4693	-1.5854	0.1927	0.11339
	0.00019286	6.3694e-06	-0.0003309	0.00049888	0.0001567	-0.0002422

Table 18: Real System and RLS Estimation with White Noise Input (P=1000, $\theta_0 = 0$) with white noise

2.3.5 $\theta_0 = 0$ and P=100

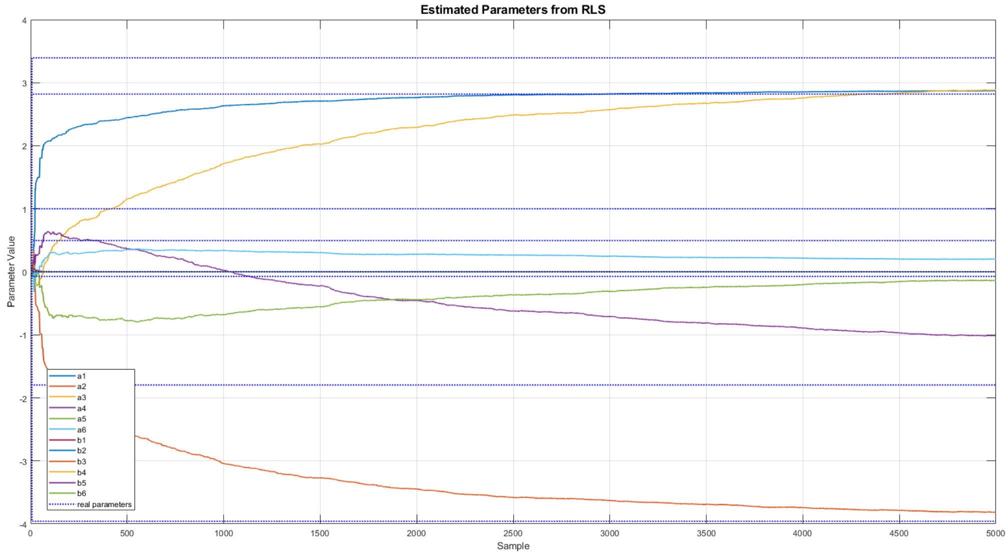


Figure 21: Estimated Parameters Using White Noise Input (P=100, $\theta_0 = 0$) with white noise

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(RLS)	2.9813 0.00019286	-4.1784 6.3694e-06	3.4693 -0.0003309	-1.5854 0.00049888	0.1927 0.0001567	0.11339 -0.0002422

Table 19: Real System and RLS Estimation with White Noise Input (P=100, $\theta_0 = 0$) with white noise

2.4 Conclusion on The Effect of The Initial Values P and θ_0 in RLS Algorithm

Upon careful examination of Figures 17 to 21, it becomes apparent that altering the value of theta0 yields negligible variations in the estimation outcomes. The plots visually depict that such alterations exert minimal influence on the results. However, from a theoretical standpoint, the selection of theta0 holds significance, with a prevailing consensus favoring the adoption of zero as the initial value. Subsequently, parameters tend to adapt iteratively, culminating in refined estimations.

Conversely, the parameter P assumes paramount importance within the algorithmic framework. It serves as a metric delineating the algorithm's perception of its own accuracy. Notably, selecting a larger value for P signifies the algorithm's acknowledgment of substantial error, prompting corrective measures. This phenomenon is vividly illustrated by the observation that a higher P value, exemplified by P=10000, facilitates superior estimation outcomes, whereas smaller P values, such as P=100 as depicted in Figure 21, engender sluggish convergence and delayed attainment of optimal parameter values.

Thus, through empirical observation and theoretical underpinnings, it becomes evident that while the manipulation of theta0 evinces minimal impact, judicious selection of the parameter P profoundly influences the algorithm's convergence trajectory and the fidelity of its estimations.

2.5 ELS Algorithm In Presence of Colored Noise

This Method is performed as follows:

```
T_ELS = inv(phi_ELS'*phi_ELS)*phi_ELS'*y;
E_ELS = y-phi_ELS*T_ELS;

Ts = 0.04;
tf = 200;
t = 0:Ts:tf; t = t';
y(5001)=y(5000);
% The first 6 signals must be created manually
y_ELS(1) = e(1);
y_ELS(2) = T_ELS(1)*y(1)+T_ELS(7)*u(1)+e(2);
y_ELS(3) = T_ELS(1)*y(2)+T_ELS(2)*y(1)+T_ELS(7)*u(2)+T_ELS(8)*u(1)+e(3);
y_ELS(4) = T_ELS(1)*y(3)+T_ELS(2)*y(2)+T_ELS(3)*y(1)+T_ELS(7)*u(3)+T_ELS(8)*u(2)+T_ELS(9)*u(1)+e(4);
y_ELS(5) = T_ELS(1)*y(4)+T_ELS(2)*y(3)+T_ELS(3)*y(2)+T_ELS(4)*y(1)+T_ELS(7)*u(4)+T_ELS(8)*u(3)+T_ELS(9)*u(2)+T_ELS(10)*u(1)+e(5);
y_ELS(6) = T_ELS(1)*y(5)+T_ELS(2)*y(4)+T_ELS(3)*y(3)+T_ELS(4)*y(2)+T_ELS(5)*y(1)+T_ELS(7)*u(5)+T_ELS(8)*u(4)+T_ELS(9)*u(3)+T_ELS(10)*u(2)+T_ELS(11)*u(1)+e(5);

for i = 7:N
y_ELS(i) = T_ELS(1)*y(i-1)+T_ELS(2)*y(i-2)+T_ELS(3)*y(i-3)+T_ELS(4)*y(i-4)+T_ELS(5)*y(i-5)+T_ELS(6)*y(i-6)+...
T_ELS(7)*u(i-1)+T_ELS(8)*u(i-2)+T_ELS(9)*u(i-3)+T_ELS(10)*u(i-4)+T_ELS(11)*u(i-5)+T_ELS(12)*u(i-6);
end

```

The results of this method are presented on next page.

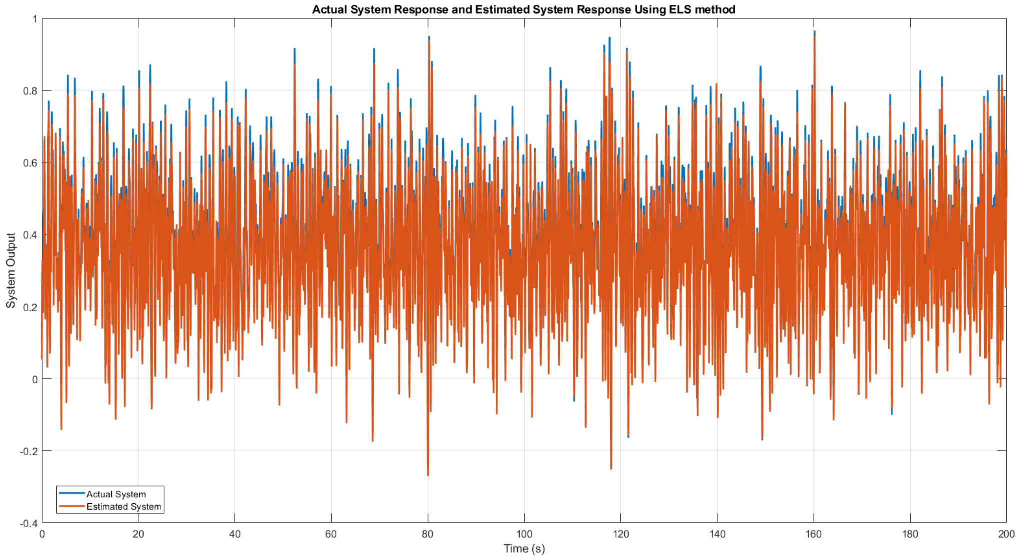


Figure 22: Estimated Parameters Using White Noise Input (ELS method) with colored noise

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(ELS)	2.4333 -0.00010988	-2.5965 6.7925e-05	1.2033 -0.00036586	0.34361 0.00035396	-0.761 0.00052431	0.29707 -9.7983e-05

Table 20: Real System and ELS Estimation with White Noise Input (ELS method) with colored noise

The data presented in Figure 22 and Table 20 unequivocally demonstrate the superior performance of the ELS method, especially in the presence of colored noise, particularly in estimating parameters labeled 'b'. Unlike other methods, ELS method proves robust in such scenarios, navigating through spectral complexities effectively. This resilience makes it a valuable asset in accurate parameter estimation, especially in noisy environments.

2.6 Under Parameter and Over Parameter effect on estimation

When we explore how well the Recursive Least Squares (RLS) and Least Squares (LS) algorithms work in figuring out the parameters of a system, we find that they're quite similar in how they operate. So, when it comes to estimating parameters in models that have too few or too many parameters compared to what's actually needed, the difference in results between these two algorithms is pretty small.

As we've discussed earlier, when the model doesn't have enough parameters (we call this "under-parameterized"), the estimates it gives us aren't very good. On the other hand, if we use a model with too many parameters (which we call "over-parameterized"), the estimates are a bit better, but they still don't match up perfectly with the true number of parameters in the original model.

2.7 RLS Estimation Under White Noise

2.7.1 Impulse Input (P=10000, $\theta_0 = 0$) with white noise

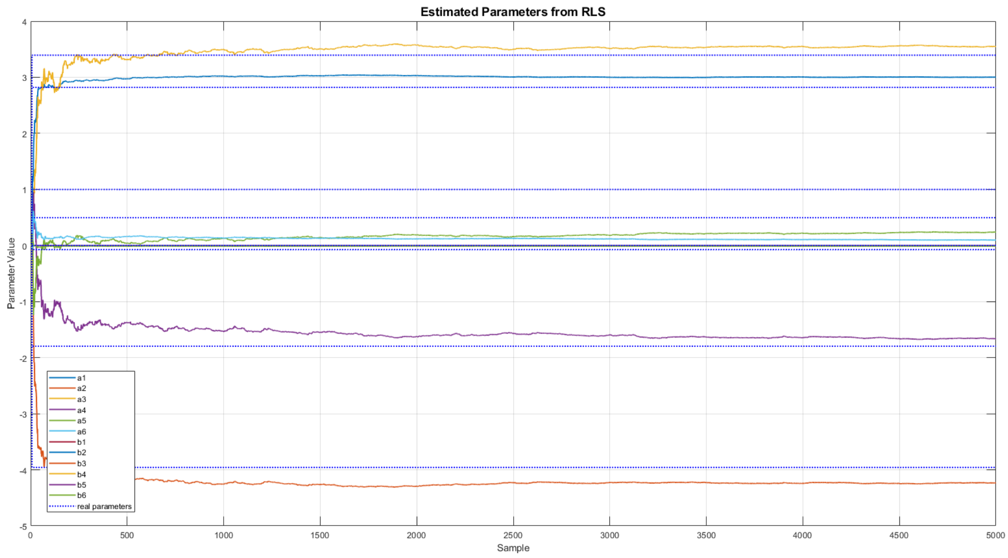


Figure 23: Estimated Parameters Using Impulse Input (P=10000, $\theta_0 = 0$) with white noise

Name	a_1 / b_1	a_2 / b_2	a_3 / b_3	a_4 / b_4	a_5 / b_5	a_6 / b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
(RLS)	3.0014	-4.2312	3.5464	-1.6591	0.23787	0.097274
	0	0	0	0	0	-0.01134

Table 21: Real System and RLS Estimation with Impulse Input (P=10000, $\theta_0 = 0$) with white noise

2.7.2 Step Input (P=10000, $\theta_0 = 0$) with white noise

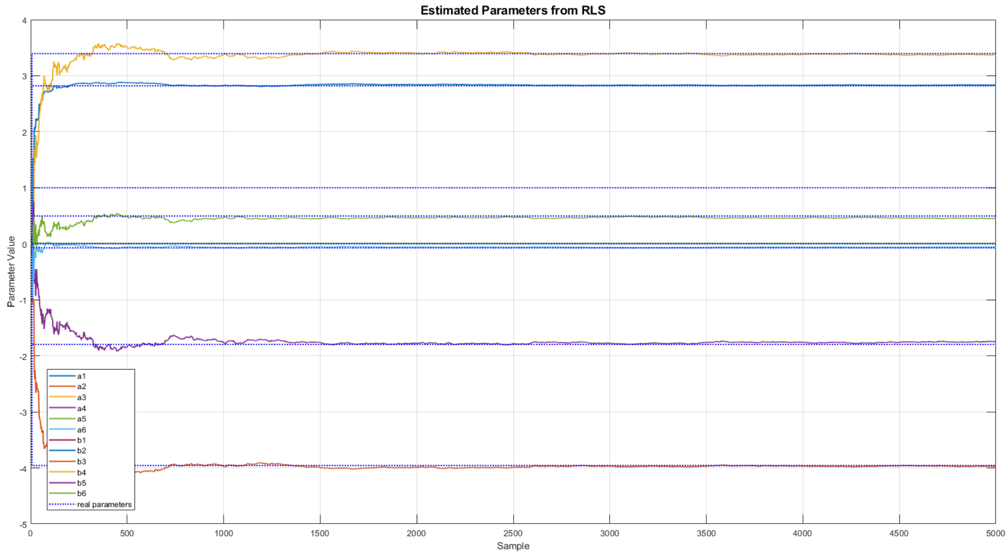


Figure 24: Estimated Parameters Using Step Input (P=10000, $\theta_0 = 0$) with white noise

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(RLS)	2.8332 0.0058388	-3.9678 0.0058388	3.3692 0.0058388	-1.7438 0.0058388	0.4515 0.0058388	-0.05805 0.0058388

Table 22: Real System and RLS Estimation with Step Input (P=10000, $\theta_0 = 0$) with white noise

2.7.3 Sinusoidal Input (P=10000, $\theta_0 = 0$) with white noise

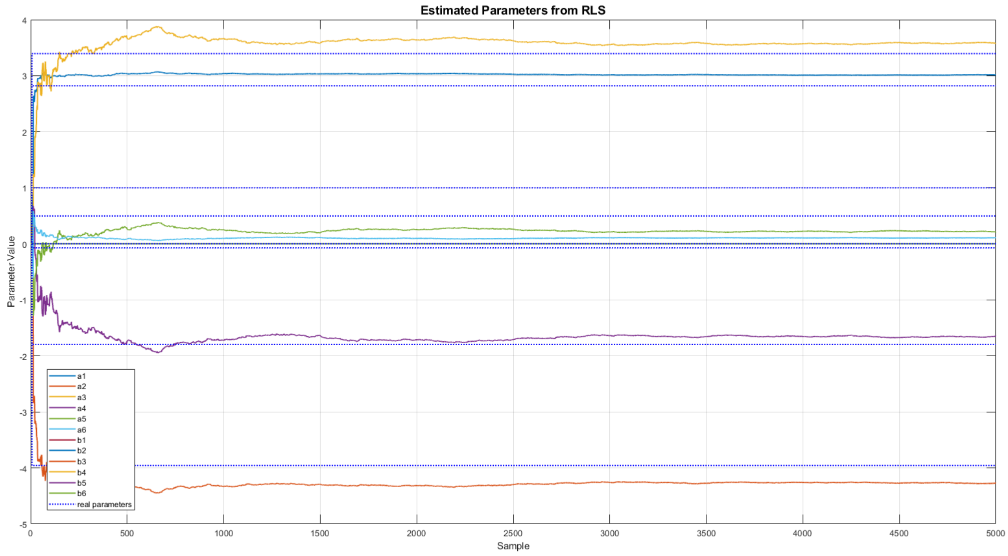


Figure 25: Estimated Parameters Using Sinusoidal Input (P=10000, $\theta_0 = 0$) with white noise

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
(RLS)	3.0149	-4.2717	3.5798	-1.651	0.21194	0.10865
	-1.6212e-10	-1.6212e-10	-1.6212e-10	-1.6212e-10	-1.6212e-10	-1.6212e-10

Table 23: Real System and RLS Estimation with Sinusoidal Input (P=10000, $\theta_0 = 0$) with white noise

2.7.4 White Noise Input (P=10000, $\theta_0 = 0$) with white noise

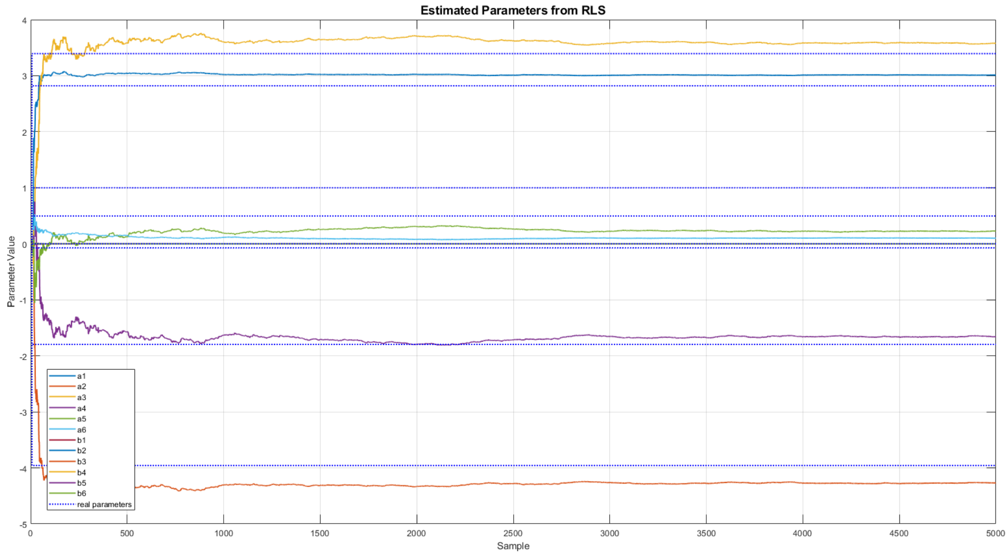


Figure 26: Estimated Parameters Using White Noise Input (P=10000, $\theta_0 = 0$) with white noise

Name	a_1	a_2	a_3	a_4	a_5	a_6
	b_1	b_2	b_3	b_4	b_5	b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
(RLS)	3.0114	-4.2663	3.5766	-1.6563	0.22655	0.10066
	8.4281e-05	3.2067e-05	-0.00063354	0.00038876	0.00063213	-0.00021179

Table 24: Real System and RLS Estimation with White Noise Input (P=10000, $\theta_0 = 0$) with white noise

2.8 RLS Estimation Under Colored Noise

2.8.1 Impulse Input (P=10000, $\theta_0 = 0$) with colored noise

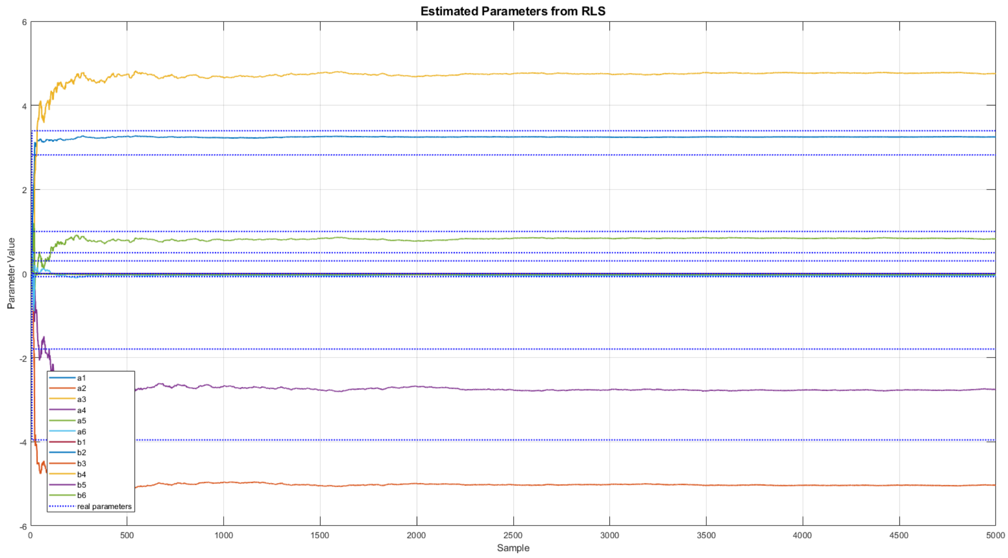


Figure 27: Estimated Parameters Using Impulse Input (P=10000, $\theta_0 = 0$) with colored noise

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
(RLS)	3.2487	-5.0326	4.7565	-2.7539	0.82405	-0.048146
	0	0	0	0	0	-0.024751

Table 25: Real System and RLS Estimation with Impulse Input (P=10000, $\theta_0 = 0$) with colored noise

2.8.2 Step Input (P=10000, $\theta_0 = 0$) with colored noise

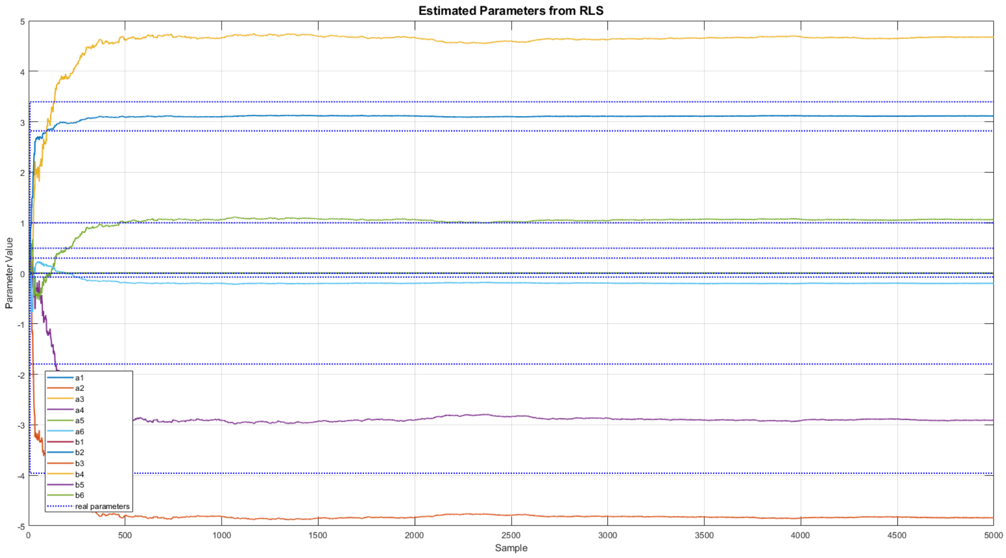


Figure 28: Estimated Parameters Using Step Input (P=10000, $\theta_0 = 0$) with colored noise

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(RLS)	3.1142 0.0063432	-4.8388 0.0063432	4.6751 0.0063432	-2.9092 0.0063432	1.0634 0.0063432	-0.1993 0.0063432

Table 26: Real System and RLS Estimation with Step Input (P=10000, $\theta_0 = 0$) with colored noise

2.8.3 Sinusoidal Input (P=10000, $\theta_0 = 0$) with colored noise

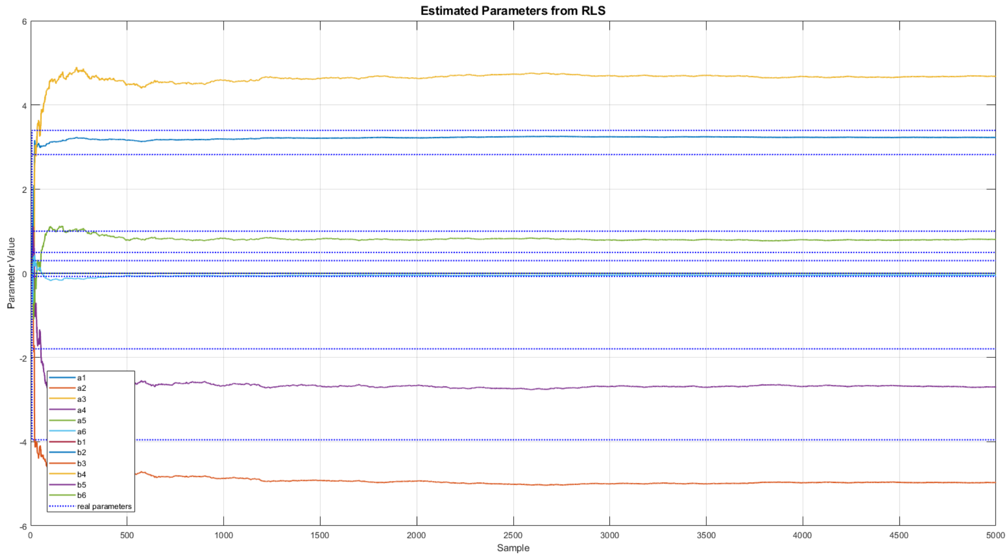


Figure 29: Estimated Parameters Using Sinusoidal Input (P=10000, $\theta_0 = 0$) with colored noise

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(RLS)	3.2266 -1.4858e-10	-4.9691 -1.4858e-10	4.6771 -1.4858e-10	-2.6981 -1.4858e-10	0.80478 -1.4858e-10	-0.046561 -1.4858e-10

Table 27: Real System and RLS Estimation with Sinusoidal Input (P=10000, $\theta_0 = 0$) with colored noise

2.8.4 White Noise Input (P=10000, $\theta_0 = 0$) with colored noise

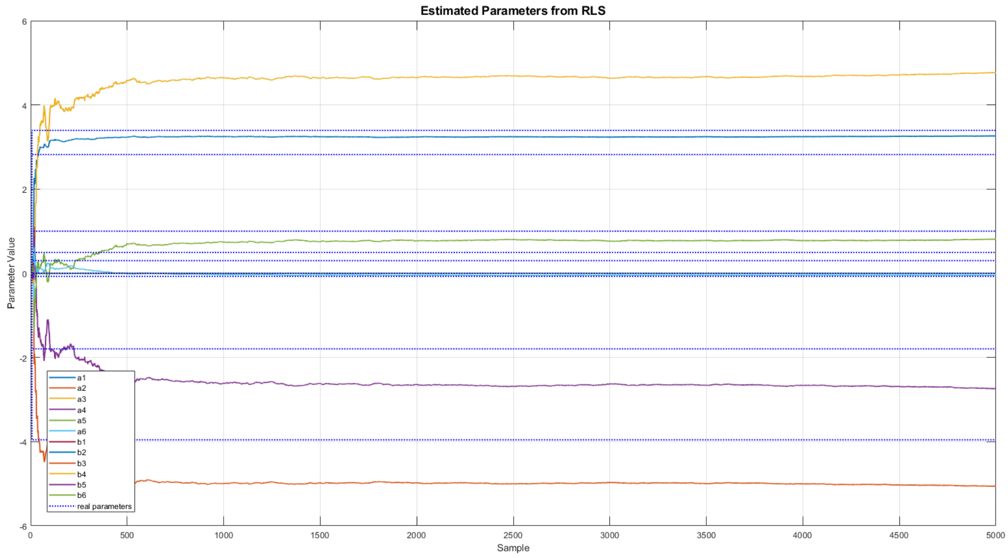


Figure 30: Estimated Parameters Using White Noise Input (P=10000, $\theta_0 = 0$) with colored noise

Name	a_1	a_2	a_3	a_4	a_5	a_6
	b_1	b_2	b_3	b_4	b_5	b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
(RLS)	3.262	-5.0599	4.7708	-2.7434	0.81304	-0.048108
	-0.00029961	6.136e-05	-0.00022867	0.00025331	0.00024638	-0.00042875

Table 28: Real System and RLS Estimation with White Noise Input (P=10000, $\theta_0 = 0$) with colored noise

2.9 Conclusion On The Effect of White and Colored Noise on The RLS Algorithm

The data from Figures 17 to 24 help us understand how different types of noise affect estimation. When there's white noise, our estimations tend to be better than when there's no noise at all. This matches what we've seen before: the more complex our input signal, the better our estimations tend to be. So, it's not surprising that white noise gives us the best results.

On the other hand, when there's colored noise, our estimations get worse compared to when we have white noise. Once again, it's clear that white noise is the best type of noise for getting accurate estimations.

2.10 λ-RLS Slow Changes in Parameters

In this section we will create a small change in the parameters and see how theλ-RLS algorithm will perform. There is one important thing to mention which is we cannot change any parameters more than 4 percent because the system will be unstable and the estimation process will no longer work properly. In order to prevent such thing and also see the working result we will change some of exclusive parameters which are b_3 and b_4. which will change as follows:

```
if t>=500  && t<700
x = x+0.002;
z= z+0.001;
b4=b4+z;
b3=b3+x;
end
```

This means when we reach the time 500 these parameters will increase by their own step until we get to time 700.

On next parts we will study the performance of the algorithm with 3 different λ values and we will plot the result and bring them on table to see how they operate. The three λ values are :

$$\begin{cases} \lambda = 1 \text{ Which is Normal RLS} \\ \lambda = 0.99 \\ \lambda = 0.96 \end{cases}$$

2.10.1 Slow Change in Parameters λ-RLS ($\lambda = 1$)

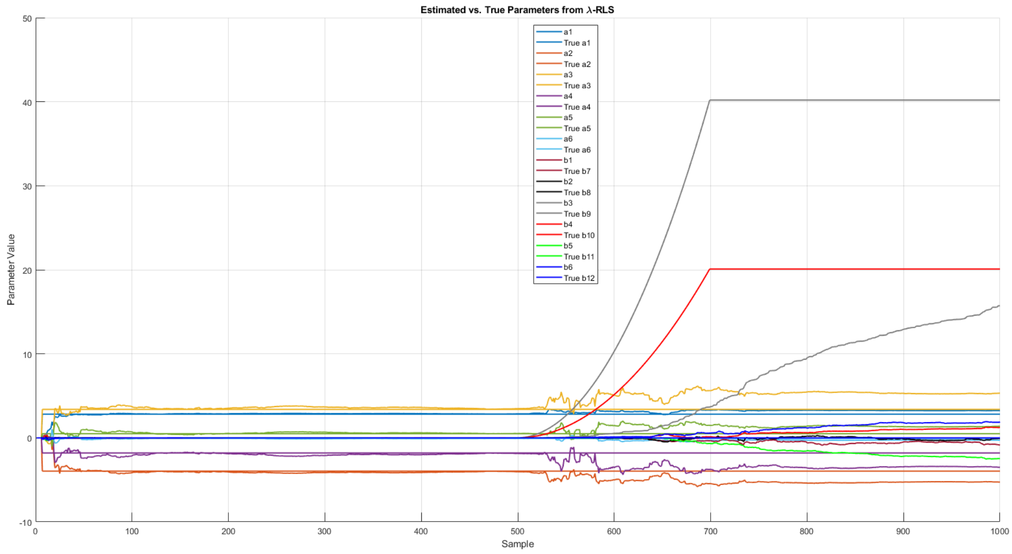


Figure 31: Estimated Parameters Using λ-RLS (P=10000, $\theta_0 = 0$,$\lambda = 1$)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(λ-RLS)	3.2438 -0.85949	-5.2566 -0.084822	5.3169 15.732	-3.497 1.2425	1.385 -2.4839	-0.2792 1.8853

Table 29: Real System and λRLS Estimation (P=10000, $\theta_0 = 0$) ,λ=1

2.10.2 Slow Change in Parameters λ-RLS ($\lambda = 0.99$)

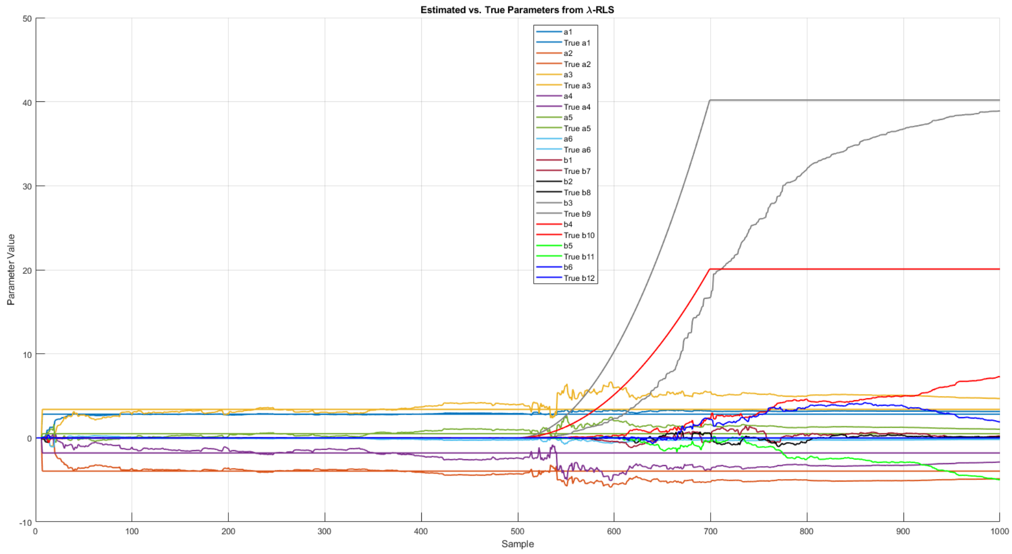


Figure 32: Estimated Parameters Using λ-RLS (P=10000, $\theta_0 = 0$,$\lambda = 0.99$)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(λ-RLS)	3.1352 0.2172	-4.8698 0.13073	4.6791 38.911	-2.8749 7.2943	1.0252 -4.9699	-0.18144 1.9058

Table 30: Real System and λRLS Estimation (P=10000, $\theta_0 = 0$) ,λ=0.99

2.10.3 Slow Change in Parameters λ-RLS ($\lambda = 0.96$)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(λ-RLS)	2.8066 -0.0039585	-3.9247 0.0013004	3.3565 40.194	-1.7705 20.654	0.48735 0.51068	-0.073853 0.1561

Table 31: Real System and λRLS Estimation (P=10000, $\theta_0 = 0$) ,λ=0.96

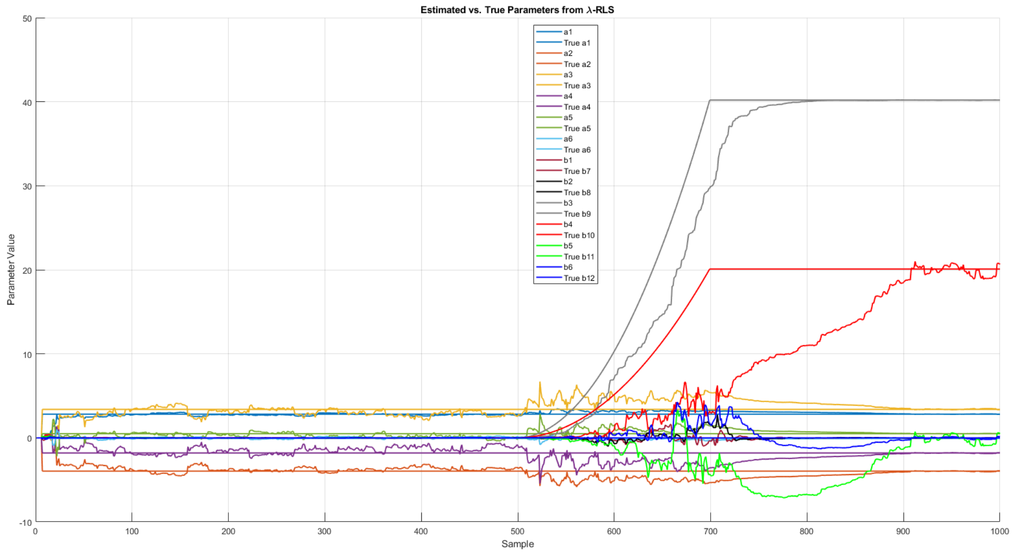


Figure 33: Estimated Parameters Using λ-RLS (P=10000, $\theta_0 = 0$,$\lambda = 0.96$)

2.11 Covariance Resetting RLS Slow Changes in Parameters

In this section we will create a small change in the parameters and see how theλ-RLS algorithm will perform. There is one important thing to mention which is we cannot change any parameters more than 4 percent because the system will be unstable and the estimation process will no longer work properly. In order to prevent such thing and also see the working result we will change some of exclusive parameters which are b_3 and b_4. which will change as follows:

```
if t>=500  && t<700
x = x+0.002;
z= z+0.001;
b4=b4+z;
b3=b3+x;
end
```

This means when we reach the time 500 these parameters will increase by their own step until we get to time 700.

On the next parts we will study how the number of the times that we reset the covariance matrix will affect the estimation quality. We will study the 1, 2, 4, 5 times of covariance resetting.

2.11.1 Slow Change in Parameters P-Resetting-RLS (One Time)

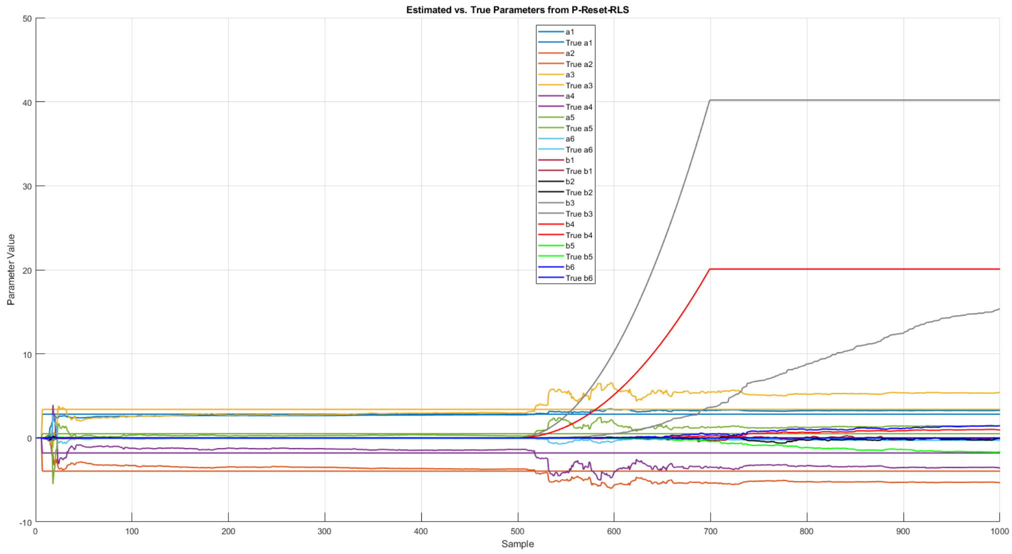


Figure 34: Estimated Parameters Using P-Resetting-RLS (P=10000, θ_0 = 0,One Time P-Resetting)

Name	a_1 / b_1	a_2 / b_2	a_3 / b_3	a_4 / b_4	a_5 / b_5	a_6 / b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
(P-Reset-RLS)	3.2639	-5.3194	5.4161	-3.5654	1.4404	-0.29871
	-0.023612	-0.095918	15.343	0.93889	-1.6775	1.4389

Table 32: Real System and P-Resetting-RLS Estimation (P=10000, θ_0 = 0) ,One Time P-Resetting

2.11.2 Slow Change in Parameters P-Resetting-RLS (Two Time)

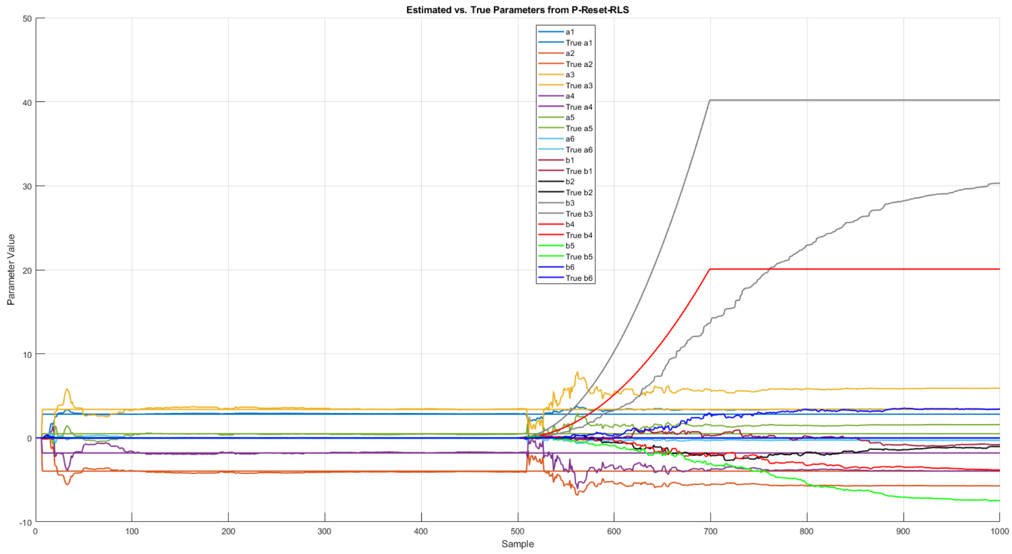


Figure 35: Estimated Parameters Using P-Resetting-RLS (P=10000, θ_0 = 0,Two Time P-Resetting)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(P-Reset-RLS)	3.4076 -0.80288	-5.7141 -1.0258	5.9059 30.304	-3.9556 -3.8184	1.5728 -7.4824	-0.31311 3.4271

Table 33: Real System and P-Resetting-RLS Estimation (P=10000, θ_0 = 0) ,Two Time P-Resetting

2.11.3 Slow Change in Parameters P-Resetting-RLS (Four Time)

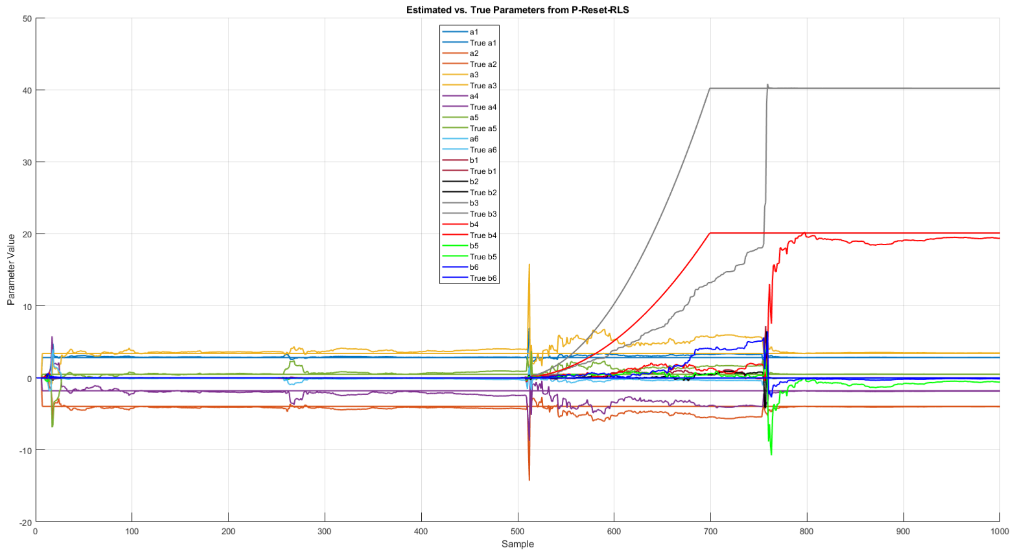


Figure 36: Estimated Parameters Using P-Resetting-RLS (P=10000, θ_0 = 0,Four Time P-Resetting)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(P-Reset-RLS)	2.8384 0.0021617	-4.0032 -0.00082107	3.4516 40.206	-1.8375 19.371	0.51229 -0.58901	-0.074993 -0.13514

Table 34: Real System and P-Resetting-RLS Estimation (P=10000, θ_0 = 0) ,Four Time P-Resetting

2.11.4 Slow Change in Parameters P-Resetting-RLS (Five Time)

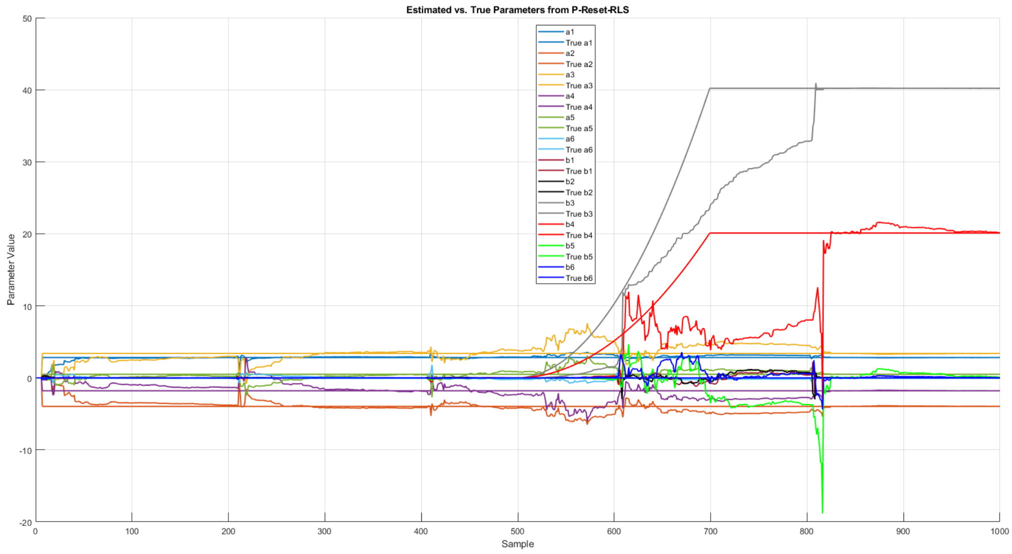


Figure 37: Estimated Parameters Using P-Resetting-RLS (P=10000, θ_0 = 0,Five Time P-Resetting)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(P-Reset-RLS)	2.8187 -0.0095977	-3.9543 0.01389	3.3916 40.198	-1.7948 20.142	0.49589 0.072858	-0.073914 0.024919

Table 35: Real System and P-Resetting-RLS Estimation (P=10000, θ_0 = 0) ,Five Time P-Resetting

2.12 Conclusion on the Comparison Between Covariance Resetting and λ RLS Method Where Parameters Change Slowly

Upon reviewing the plots presented in Figures 25 to 31, a discernible pattern emerges concerning the performance of the Landa RLS method. It becomes evident that reducing the value of landa leads to expedited estimation processes. However, a caveat surfaces: such expedited estimations often yield oscillatory parameters, a less desirable outcome. Hence, the pursuit of an optimal landa value becomes imperative. This optimal value should strike a balance, facilitating swift algorithmic performance while mitigating oscillations in parameter estimation.

Transitioning to the discussion on the P Resetting RLS algorithm, an observation emerges: resetting the covariance matrix repeatedly expedites the attainment of true parameters. Yet, this expedited pace comes with a downside—it risks impeding convergence to the parameter and potentially necessitating a reset of the algorithm.

In summation, a comparative analysis reveals the superiority of the Landa RLS method over the P Resetting algorithm. The former exhibits superior convergence speed and precision, notwithstanding its oscillatory tendencies, thus emerging as the preferred choice for estimation tasks.

2.13 λ-RLS Fast Changes in Parameters

In this section we will study the performance of the λ-RLS method where the parameters change instantly in a specific time. Again as mentioned in previous sections we cannot change any parameter we want because the system will be unstable and the estimation cannot operate properly. Hence we will change the parameters b_3 and b_4 which will change as follows:

```
if t==500
b4=b4+1;
end
if t==300
b3=b3+2;
end
```

This means that at the times 300 and 500 each parameter will face a sudden change. On next parts we will study the performance of the algorithm with 3 different λ values and we will plot the result and bring them on table to see how they operate. The three λ values are :

$$\begin{cases} \lambda = 1 \text{ Which is Normal RLS} \\ \lambda = 0.99 \\ \lambda = 0.96 \end{cases}$$

2.13.1 Fast Change in Parameters λ-RLS ($\lambda = 1$)

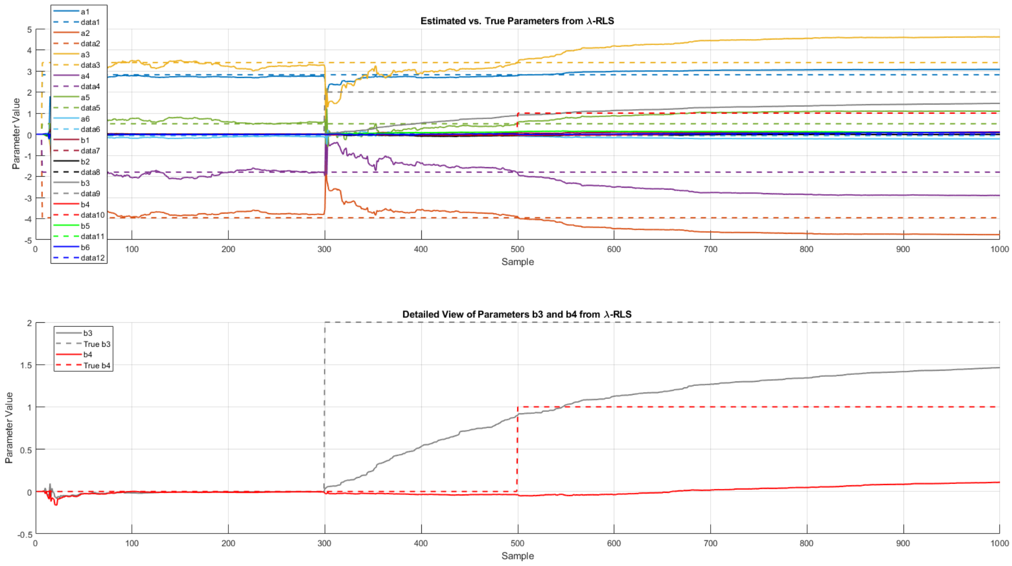


Figure 38: Estimated Parameters Using λ-RLS (P=10000, $\theta_0 = 0$,$\lambda = 1$)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(λ-RLS)	3.0717 -2.9001	-4.7569 1.0939	4.6107 -2.9001	-2.9001 1.0939	1.0939 -2.9001	-0.2188 1.0939

Table 36: Real System and λRLS Estimation (P=10000, $\theta_0 = 0$) ,λ=1

2.13.2 Fast Change in Parameters λ-RLS ($\lambda = 0.99$)

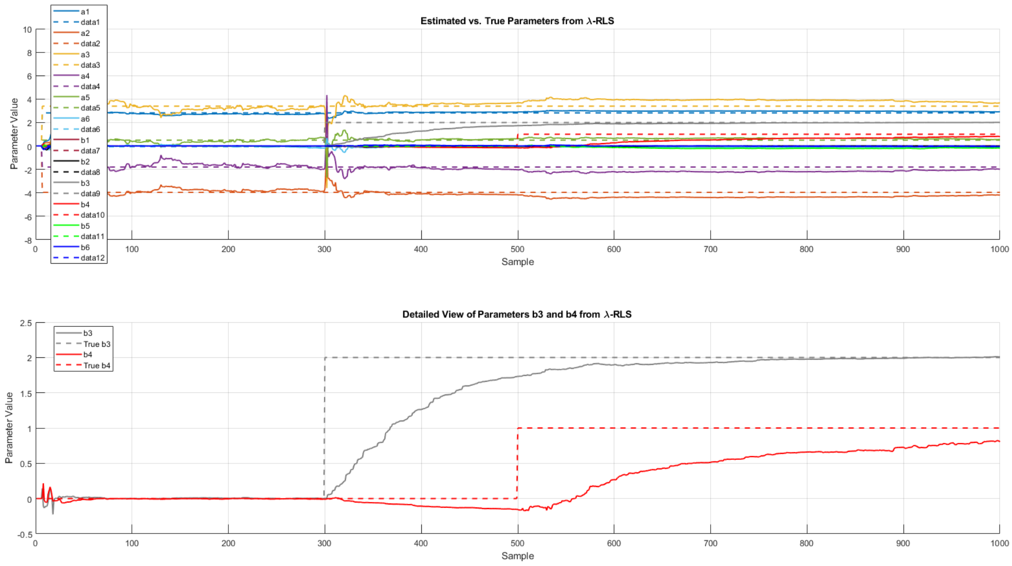


Figure 39: Estimated Parameters Using λ-RLS (P=10000, $\theta_0 = 0$,$\lambda = 0.99$)

Name	a_1	a_2	a_3	a_4	a_5	a_6
	b_1	b_2	b_3	b_4	b_5	b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
(λ-RLS)	2.9188	-4.1963	3.678	-1.9868	0.55903	-0.072324
	-1.9868	0.55903	-1.9868	0.55903	-1.9868	0.55903

Table 37: Real System and λRLS Estimation (P=10000, $\theta_0 = 0$) ,λ=0.99

2.13.3 Fast Change in Parameters λ-RLS ($\lambda = 0.96$)

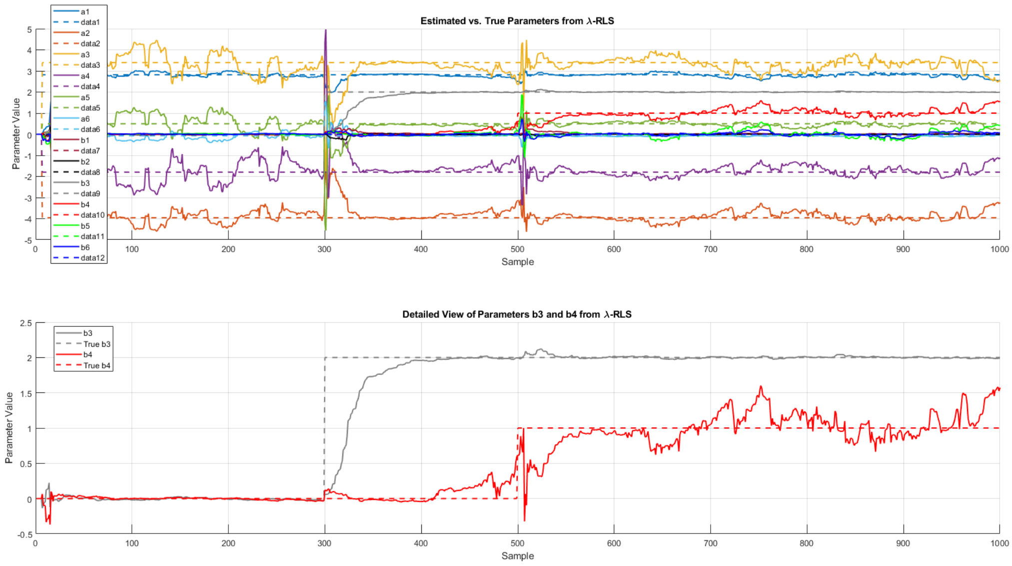


Figure 40: Estimated Parameters Using λ-RLS (P=10000, $\theta_0 = 0$,$\lambda = 0.96$)

Name	a_1	a_2	a_3	a_4	a_5	a_6
	b_1	b_2	b_3	b_4	b_5	b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
(λ-RLS)	2.539	-3.2353	2.482	-1.1165	0.21611	-0.042791
	-1.1165	0.21611	-1.1165	0.21611	-1.1165	0.21611

Table 38: Real System and λRLS Estimation (P=10000, $\theta_0 = 0$) ,λ=0.96

2.14 Covariance Resetting RLS Slow Changes in Parameters

In this section we will study the performance of the P-Resetting-RLS method where the parameters change instantly in a specific time. Again as mentioned in previous sections we cannot change any parameter we want because the system will be unstable and the estimation cannot operate properly. Hence we will change the parameters b_3 and b_4 which will change as follows:

```
if t==500
b4=b4+1;
end
if t==300
b3=b3+2;
end
```

This means that at the times 300 and 500 each parameter will face a sudden change. On next parts we will study the performance of the algorithm by resetting the covariance matrix one, two, four, and five times.

2.14.1 Fast Change in Parameters P-Resetting-RLS (One Time)

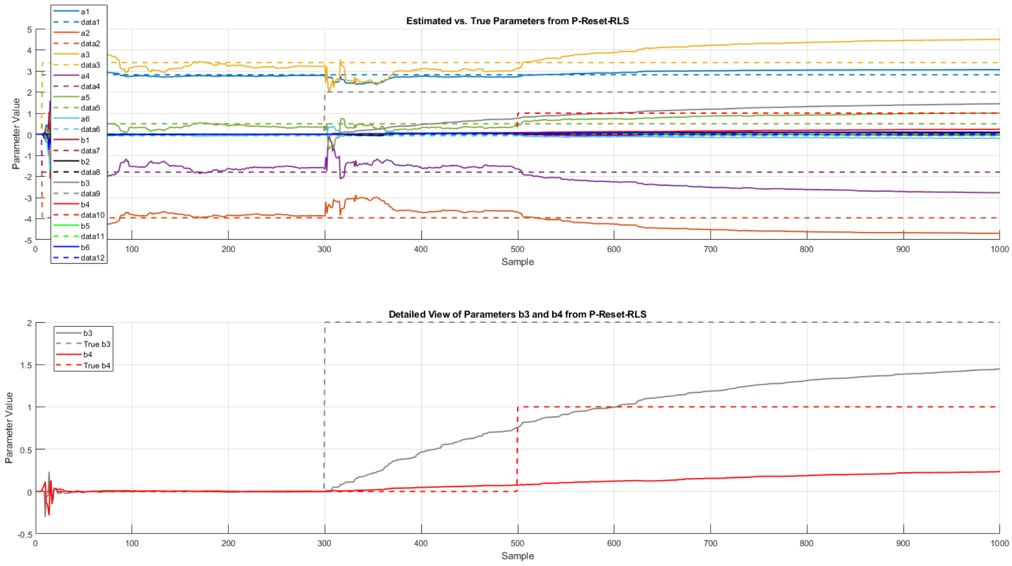


Figure 41: Estimated Parameters Using P-Resetting-RLS (P=10000, $\theta_0 = 0$,One Time P-Resetting)

Name	a_1 / b_1	a_2 / b_2	a_3 / b_3	a_4 / b_4	a_5 / b_5	a_6 / b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
(P-Reset-RLS)	3.0588	-4.6965	4.4929	-2.7688	1.0043	-0.18906
	-2.7688	1.0043	-2.7688	1.0043	-2.7688	1.0043

Table 39: Real System and P-Resetting-RLS Estimation (P=10000, $\theta_0 = 0$) ,One Time P-Resetting

2.14.2 Fast Change in Parameters P-Resetting-RLS (Two Time)

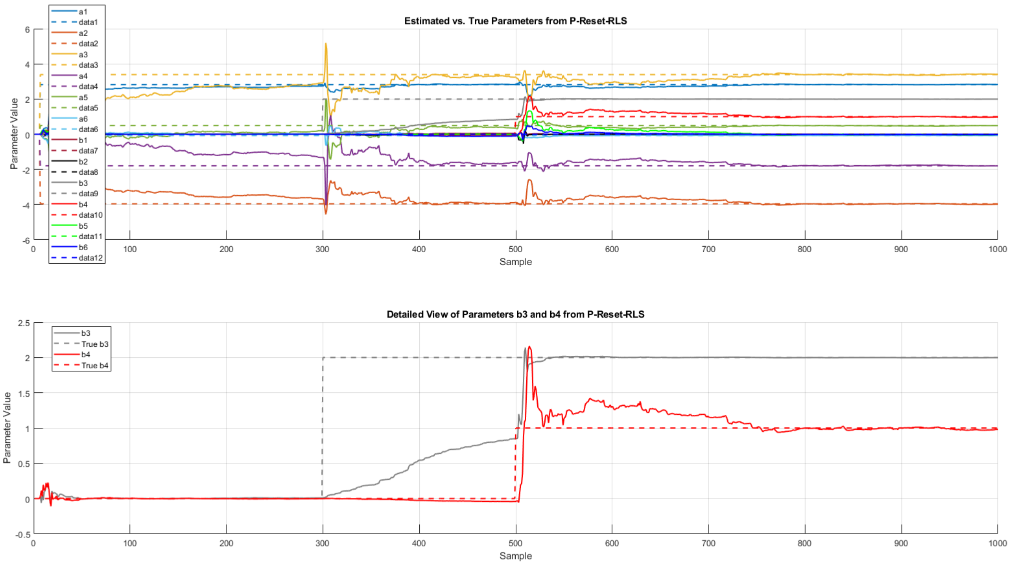


Figure 42: Estimated Parameters Using P-Resetting-RLS (P=10000, θ_0 = 0,Two Time P-Resetting)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
(P-Reset-RLS)	2.8306	-3.9802	3.4114	-1.8009	0.49031	-0.069373
	-1.8009	0.49031	-1.8009	0.49031	-1.8009	0.49031

Table 40: Real System and P-Resetting-RLS Estimation (P=10000, θ_0 = 0) ,Two Time P-Resetting

2.14.3 Fast Change in Parameters P-Resetting-RLS (Four Time)

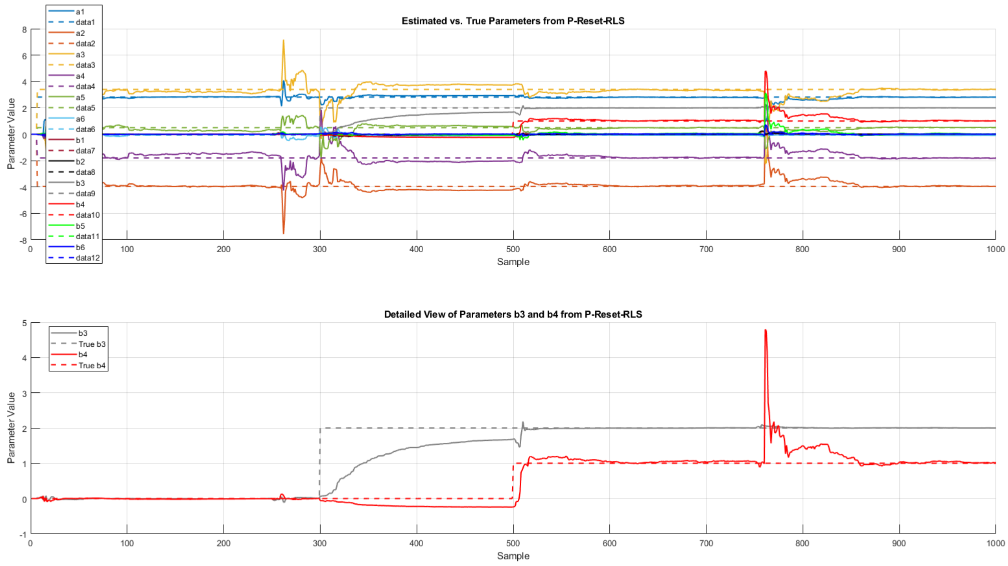


Figure 43: Estimated Parameters Using P-Resetting-RLS (P=10000, θ_0 = 0,Four Time P-Resetting)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201 0.00013078	-3.957 8.6965e-05	3.3943 -0.00029035	-1.7958 0.00034849	0.49571 0.00034849	-0.073642 -5.8554e-05
(P-Reset-RLS)	2.813 -1.8242	-3.9556 0.51996	3.4042 -1.8242	-1.8242 0.51996	0.51996 -1.8242	-0.084963 0.51996

Table 41: Real System and P-Resetting-RLS Estimation (P=10000, θ_0 = 0) ,Four Time P-Resetting

2.14.4 Fast Change in Parameters P-Resetting-RLS (Five Time)

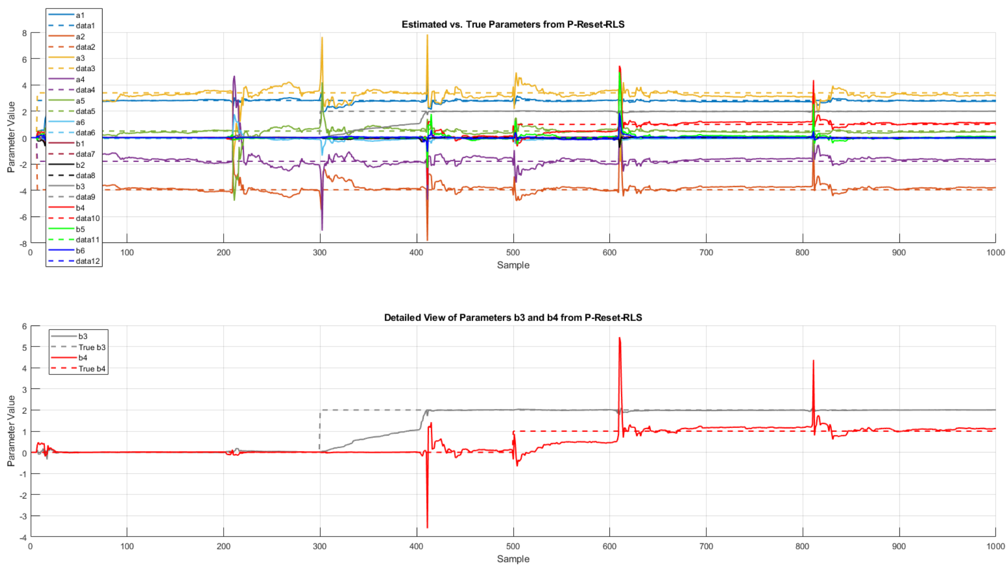


Figure 44: Estimated Parameters Using P-Resetting-RLS (P=10000, θ_0 = 0,Five Time P-Resetting)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4	a_5 b_5	a_6 b_6
Real Parameters	2.8201	-3.957	3.3943	-1.7958	0.49571	-0.073642
	0.00013078	8.6965e-05	-0.00029035	0.00034849	0.00034849	-5.8554e-05
(P-Reset-RLS)	2.7547	-3.7928	3.1899	-1.6479	0.43965	-0.066787
	-1.6479	0.43965	-1.6479	0.43965	-1.6479	0.43965

Table 42: Real System and P-Resetting-RLS Estimation (P=10000, θ_0 = 0) ,Five Time P-Resetting

2.15 Conclusion on the Comparison Between Covariance Resetting and λ RLS Method Where Parameters Change Fast

Upon analyzing Figures 38 to 44, similar conclusions emerge regarding slow changes. However, when individually assessing methods, such as P-Resetting-RLS and landa-RLS, distinct observations surface. Specifically, in scenarios with rapid parameter shifts, P-Resetting-RLS proves more effective. This highlights the method's superior adaptability to dynamic changes compared to landa-RLS. Such insights emphasize the importance of method selection tailored to the specific dynamics of parameter variations, thereby optimizing the accuracy of estimation processes in dynamic environments.

2.16 Estimating System Using RLS, LMS, PA, SA Methods with White Noise Input at Presence of White Noise (P=10000, $\theta_0 = 0$)

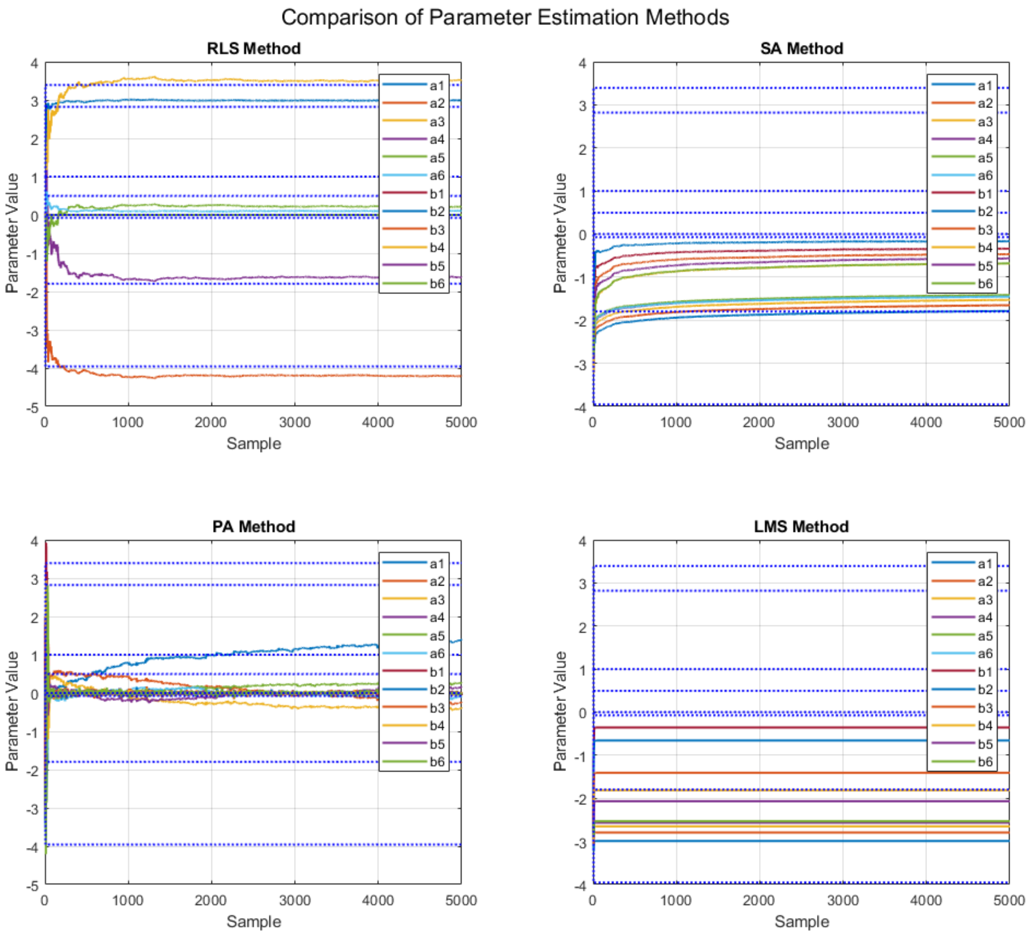


Figure 45: Comparison of RLS, LMS, PA, SA Methods

	True Parameters	RLS	SA	PA	LMS
a_1	2.8201	2.9924	-1.7858	1.3704	-2.9912
a_2	-3.957	-4.2094	-1.652	-0.26287	-2.7951
a_3	3.3943	3.517	-1.5287	-0.40958	-2.6516
a_4	-1.7958	-1.6261	-1.4433	0.13818	-2.5652
a_5	0.49571	0.21071	-1.4155	0.25049	-2.5289
a_6	-0.073642	0.10805	-1.452	-0.11391	-2.5333
b_1	0.00013078	7.0333e-05	-0.3392	-0.056908	-0.35488
b_2	8.695e-05	-0.0003443	-0.16794	-0.045114	-0.65799
b_3	-0.00029035	-0.00042474	-0.46785	-0.013006	-1.4072
b_4	0.00034849	0.00081258	-0.68135	0.0040294	-1.8135
b_5	0.00034849	0.00021985	-0.65859	0.095331	-2.0679
b_6	-5.8554e-05	-0.00064284	-0.68002	-0.011632	-2.532
RMSD	NaN	0.14476	3.4074	2.6086	4.5603

Table 43: Comparison of Parameter Estimation Methods and Real System Parameters

Looking at the table and the graph provided, here's a simple conclusion:

The table shows various methods used to estimate parameters labeled as a_1 to a_6 and b_1 to b_6, along with an error measurement called RMSD. The true values of the parameters are listed for comparison.

From the table, it's clear that the RLS method has the smallest RMSD value, indicating it has the closest estimates to the true parameters. This suggests that the RLS method is the most accurate among the ones listed. On the other hand, the LMS method shows the highest RMSD value, which means it's less accurate.

The graph provides a visual comparison of the estimation quality. In a good estimation, the estimated values would closely follow the dotted lines that represent the true parameter values. The RLS method shows estimates that quickly converge to the true values, which aligns with its low RMSD score. In contrast, the LMS method's lines do not settle close to the true values, reflecting its higher RMSD score and thus lower estimation quality.

The SA and PA methods fall between RLS and LMS in terms of accuracy. The SA method's lines diverge significantly from the true values, while the PA method's lines are closer but still not as accurate as the RLS.

In summary, the RLS method gives the best parameter estimates, followed by PA, with LMS being the least accurate according to the given data.

3 System Estimation Under Feedback Effect

In this section we will study a system under the effect of a controller with a simple proportional controller. The system is defined as follows:

$$\rightarrow H(z) = \frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}$$

We will add a white noise to the system with the variance of 0.07. and the parameters are defined as bellow:

```
a1 = 0.25;
a2 = -0.02;
a3 = 0.01;
a4 = 0.02;

b1 = 0.6;
b2 = 0.8;
b3 = 0.5;
b4 = 0.4;
```

We will simply design add a proportional control with a gain of $K = 0.2$. The controller is as follows:

```
% Proportional Controller
K=0.2;
H = feedback(G_d,K);
```

Now we can start estimation process.

3.1 RLS Method Stable System (Feedback)

The step response of the system is shown as bellow which demonstrates the stability of the system.

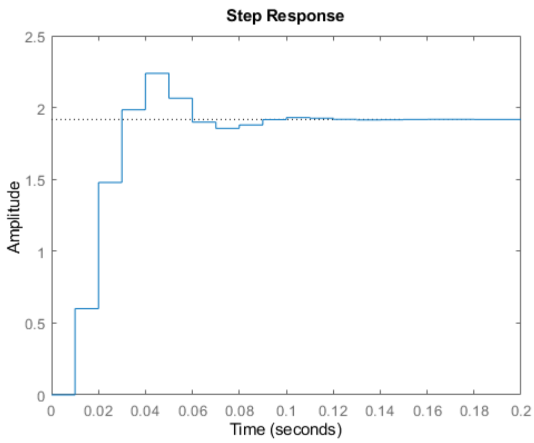


Figure 46: Step Response of the feedback system

As it is illustrated in the figure above the system reaches stability at a good time. Now we can perform the RLS method as bellow:

```
phi = zeros(N,8);
phi(2,:) = [y(1) 0 0 0 u(1) 0 0 0];
phi(3,:) = [y(2) y(1) 0 0 u(2) u(1) 0 0];
phi(4,:) = [y(3) y(2) y(1) 0 u(3) u(2) u(1) 0];
for i=5:N
phi(i,:) = [y(i-1) y(i-2) y(i-3) y(i-4) u(i-1) u(i-2) u(i-3) u(i-4)];
end

phi = phi';
Teta_RLS=q4*ones(8,N);
Teta_RLS(:,1:4)= 0;
K = zeros(8,N);
K(:,1:4)= 0;
for t=5:N
p_inv = inv(P(:,:,t-1))+phi(:,t)*phi(:,t)';
P(:,:,t) = inv(p_inv);
K(:,t)=P(:,:,t)*phi(:,t);
Teta_RLS(:,t)=Teta_RLS(:,t-1)+K(:,t)*(y(t)-phi(:,t)'*Teta_RLS(:,t-1));
end
```

Now we will give the system white noise input and estimate the system. with P=10000 and $\theta_0 = 0$.

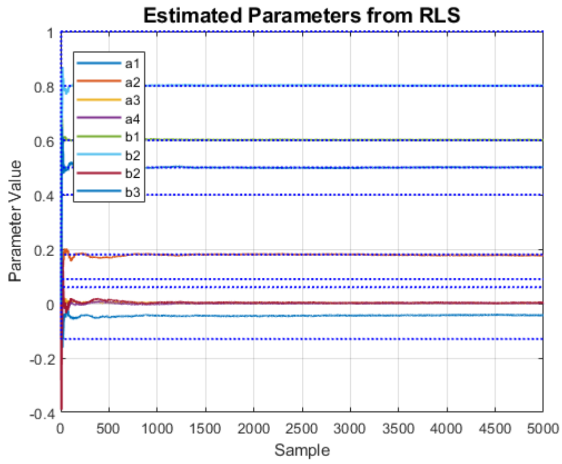


Figure 47: Estimated parameters of the feedback system using RLS

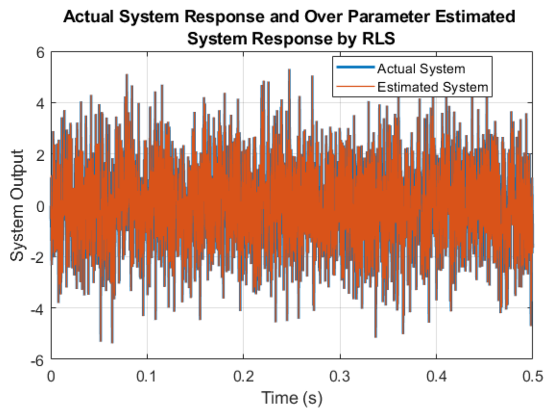


Figure 48: Estimated Response of the feedback system using RLS

Name	a_1	a_2	a_3	a_4
	b_1	b_2	b_3	b_4
Real Parameters	-0.13	0.18	0.09	0.06
	0.6	0.8	0.5	0.4
(RLS)	-0.040121	0.17855	0.00040546	0.0010903
	0.59919	0.79876	0.00057578	0.5003

Table 44: Real feedback System and RLS Estimation (P=10000, $\theta_0 = 0$)

As we can see the RLS algorithm could do better on open loop systems the estimation of a system under feed back is not very good.

3.2 KF Method Stable System (Feedback)

RLS method did not give us a good estimation hence we will perform KF method as follows:

```
%% Estimation Using KF

phi = zeros(N,8);
phi(2,:) = [y(1) 0 0 0 u(1) 0 0 0];
phi(3,:) = [y(2) y(1) 0 0 u(2) u(1) 0 0];
phi(4,:) = [y(3) y(2) y(1) 0 u(3) u(2) u(1) 0];
for i=5:N
phi(i,:) = [y(i-1) y(i-2) y(i-3) y(i-4) u(i-1) u(i-2) u(i-3) u(i-4)];
end

phi = phi';

p_kf(:,:,1)=q3*eye(8);
p_kf(:,:,2)=p_kf(:,:,1);
p_kf(:,:,3)=p_kf(:,:,1);
p_kf(:,:,4)=p_kf(:,:,1);

Teta_kf(:,1)=q4*[1;1;1;1;1;1;1;1];
Teta_kf(:,2)=Teta_kf(:,1);
Teta_kf(:,3)=Teta_kf(:,1);
Teta_kf(:,4)=Teta_kf(:,1);

K_KF(:,1)=zeros(8,1); K_KF(:,2)=zeros(8,1); K_KF(:,3)=zeros(8,1); K_KF
  (:,4)=zeros(8,1);

for t=5:N
K_KF(:,t)=p_kf(:,:,t-1)*phi(:,t)*inv(1+phi(:,t)'*p_kf(:,:,t-1)*phi(:,t))
  ;
p_kf(:,:,t)=p_kf(:,:,t-1)-p_kf(:,:,t-1)*phi(:,t)*inv(1+phi(:,t)'*p_kf
  (:,:,t-1)*phi(:,t))*phi(:,t)'*p_kf(:,:,t-1)+0.05;
epsilon(t)=y(t)-phi(:,t)'*Teta_kf(:,t-1);
Teta_kf(:,t)=Teta_kf(:,t-1)+K_KF(:,t)*epsilon(t);
end
```

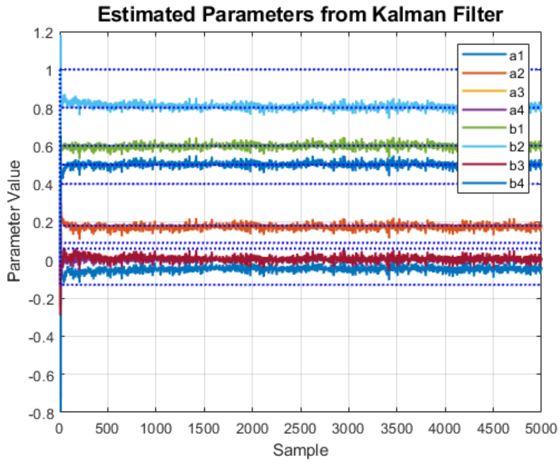


Figure 49: Estimated parameters of the feedback system using KF

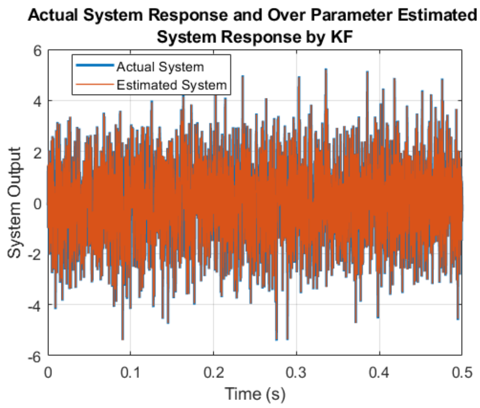


Figure 50: Estimated Response of the feedback system using KF

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4
Real Parameters	-0.13 0.6	0.18 0.8	0.09 0.5	0.06 0.4
(KF)	-0.04883 0.59672	0.17312 0.80107	0.0014067 0.0030818	-0.000887 0.49849

Table 45: Real feedback System and RLS Estimation (P=10000, $\theta_0 = 0$)

As we can see the KF algorithm has a better estimation specially in b parameters

3.3 Unstable System Estimation

The RLS and KF algorithm cannot estimate the unstable system the K matrix in both algorithm will reach NaN in MATLAB after some time and the algorithm stops working.

4 System Identification Using Kalman Filter Method

In this section we will study a system which is defined as follows:

$$\rightarrow H(z) = \frac{b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}$$

We will add a white noise to the system with the variance of 0.07. and the parameters are defined as bellow:

```
a1 = 0.25;
a2 = -0.02;
a3 = 0.01;
a4 = 0.02;

b1 = 0.6;
b2 = 0.8;
b3 = 0.5;
b4 = 0.4;
```

As we know from previous parts and figure 46 the system is stable and there is no need to concern about the quality of estimation and we expect a good estimation form this algorithm. Also the code of KF method is mentioned before.

4.1 KF Method No Change in Parameters

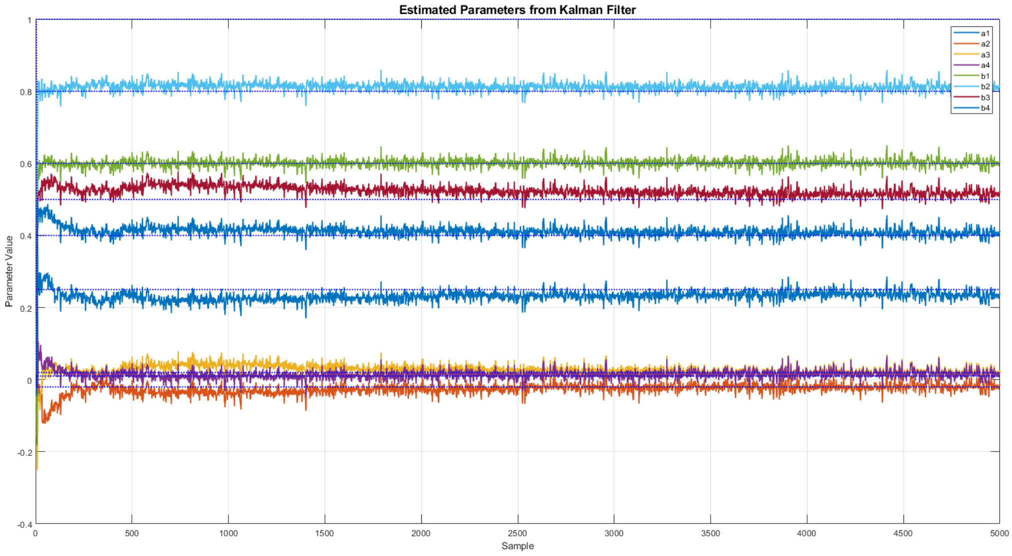


Figure 51: Estimated parameters vs. real data using Kalman Filter method

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4
Real Parameters	0.25 0.6	-0.02 0.8	0.01 0.5	0.02 0.4
(KF)	0.23721 0.60434	-0.015108 0.81488	0.023864 0.51933	0.01864 0.40945

Table 46: Real System and KF Estimation

4.2 RLS Method No Change in Parameters

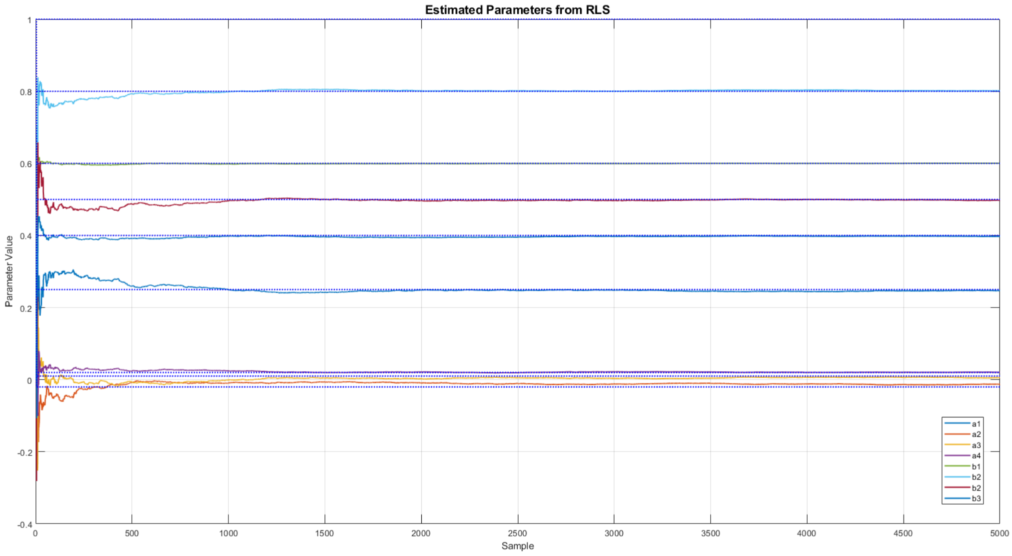


Figure 52: Estimated parameters vs. real data using RLS method

Name	a_1	a_2	a_3	a_4
	b_1	b_2	b_3	b_4
Real Parameters	0.25	-0.02	0.01	0.02
	0.6	0.8	0.5	0.4
(RLS)	0.2469	-0.012551	0.0058007	0.02055
	0.60042	0.80171	0.49792	0.39741

Table 47: Real System and RLS Estimation

4.3 Conclusion on Comparing RLS and KF

When the system is stable as it is in this section, and there is no feedback effect both Recursive Least Square and Kalman Filter approach are operating very good and the y bot suggest a very good estimation.

But we can report the RMSD error of each method which is:

$$RMSD_{RLS} = 0.0044095$$

$$RMSD_{KF} = 0.030798$$

Therefore RLS method has a better performance compared to KF method.

Also the KF method gives us an oscillatory estimation which can cause some problems.

4.4 KF Method Slow Changes in Parameters(Covariance Resetting)

4.4.1 KF ,Slow Changes, One time P Resetting

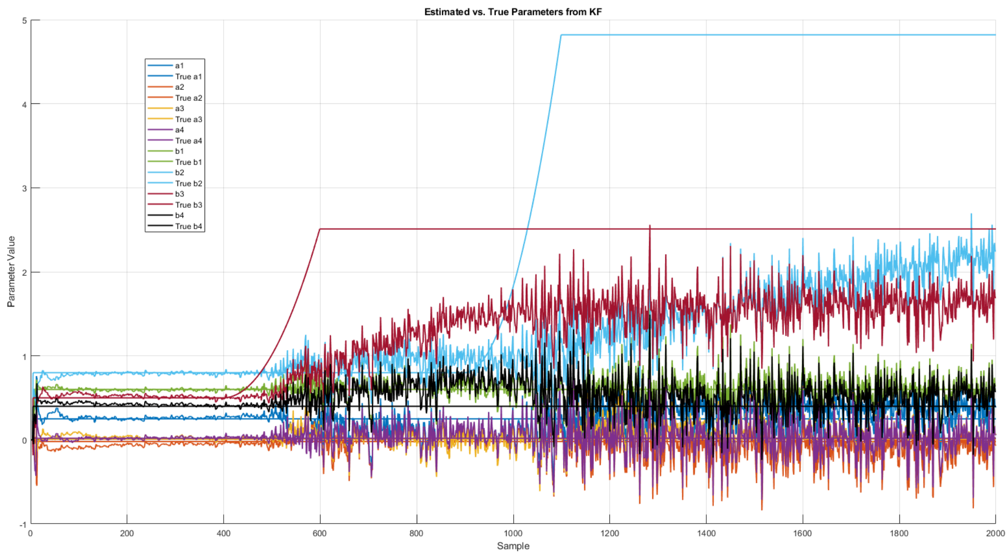


Figure 53: Estimated parameters vs. real data using KF method (One-P-Resetting)

Name	a_1	a_2	a_3	a_4
	b_1	b_2	b_3	b_4
Real Parameters	0.25	-0.02	0.01	0.02
	0.6	0.8	0.5	0.4
(KF)	0.38739	-0.068528	0.060823	0.055232
	0.64321	2.2472	1.6935	0.57695

Table 48: Real System and KF Estimation (One-P-Resetting)

4.4.2 KF ,Slow Changes, Two time P Resetting

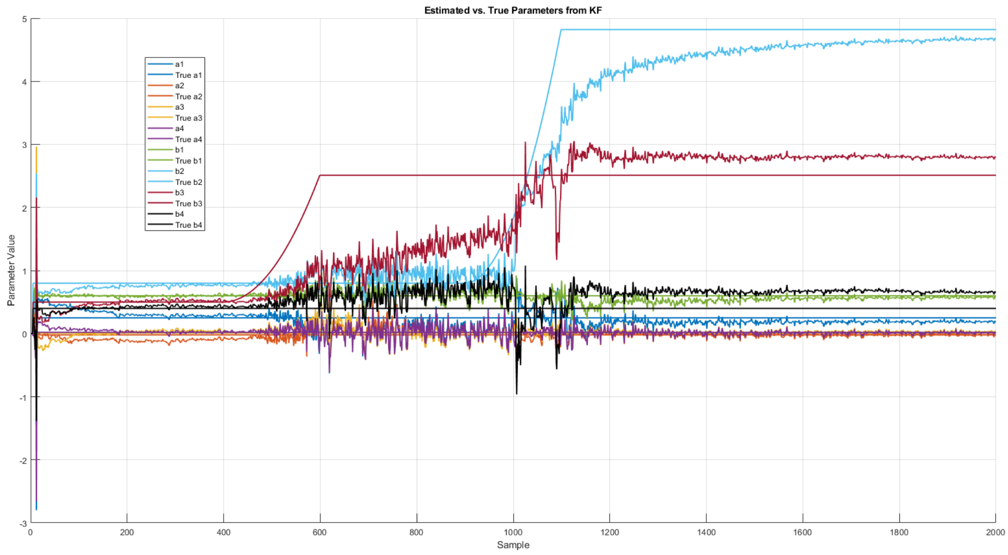


Figure 54: Estimated parameters vs. real data using KF method (Two-P-Resetting)

Name	a_1	a_2	a_3	a_4
	b_1	b_2	b_3	b_4
Real Parameters	0.25	-0.02	0.01	0.02
	0.6	0.8	0.5	0.4
(KF)	0.18798	-0.022487	0.03022	0.012872
	0.57442	4.67	2.7925	0.65488

Table 49: Real System and KF Estimation (Two-P-Resetting)

4.4.3 KF ,Slow Changes, Four time P Resetting

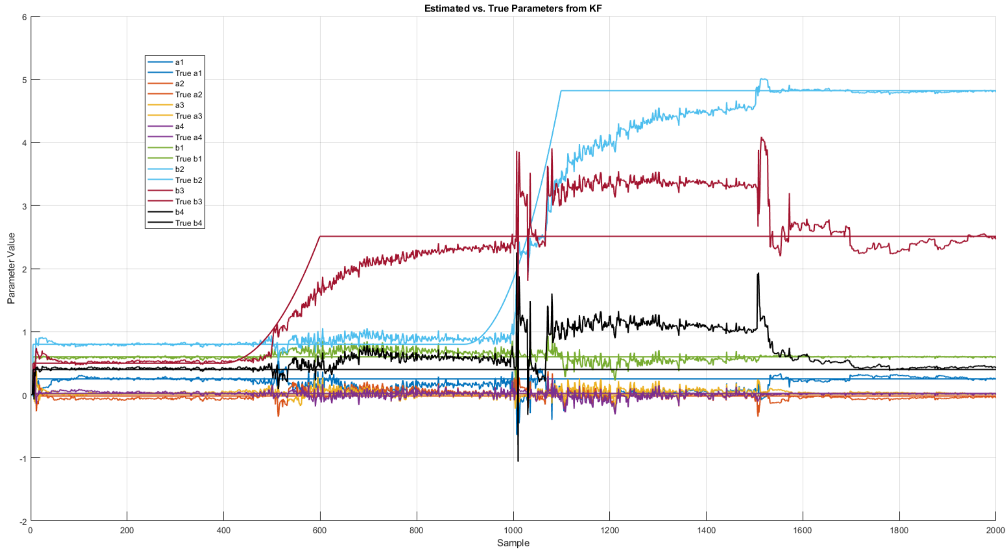


Figure 55: Estimated parameters vs. real data using KF method (Four-P-Resetting)

Name	a_1	a_2	a_3	a_4
	b_1	b_2	b_3	b_4
Real Parameters	0.25	-0.02	0.01	0.02
	0.6	0.8	0.5	0.4
(KF)	0.18798	-0.022487	0.03022	0.012872
	0.57442	4.67	2.7925	0.65488

Table 50: Real System and KF Estimation (Four-P-Resetting)

4.4.4 KF ,Slow Changes, Five time P Resetting

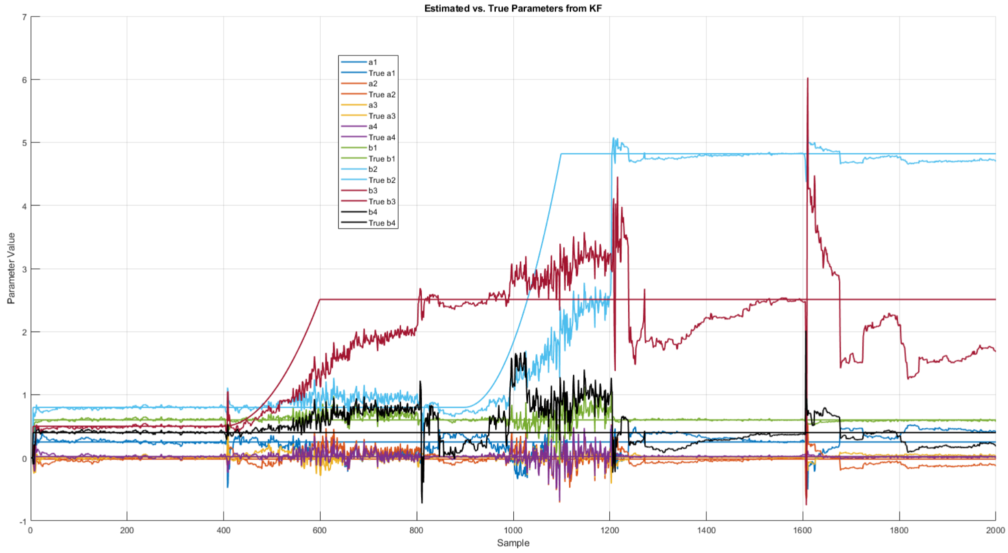


Figure 56: Estimated parameters vs. real data using KF method (Five-P-Resetting)

Name	a_1 b_1	a_2 b_2	a_3 b_3	a_4 b_4
Real Parameters	0.25	-0.02	0.01	0.02
	0.6	0.8	0.5	0.4
(KF)	0.41013	-0.12279	0.029948	0.0033153
	0.57962	4.7025	1.6852	0.18873

Table 51: Real System and KF Estimation (Five-P-Resetting)

4.5 Conclusion On KF Method Where There is Slow Changes in Parameters

The conclusion is pretty much the same as RLS method. When using P-Resetting-KF method we can see that for one time covariance resetting we cannot get an acceptable estimation. When we reset the covariance matrix two times we get a much better estimation. But resetting the covariance matrix more than two times is not a very good thing to do because after the second covariance resetting the parameters are estimated very good so resetting it again is not a good idea. But it does not mean that always resetting twice is the best option. It really depends on how and when the parameters are changing.

5 Non Linear System Identification

Consider the following nonlinear pendulum system and identify it using the RLS method. Then investigate the effect of changing the model order on the optimal cost function. Also, determine the appropriate model order. For this part, you can use the AIC criterion.

$$AIC = N ln(V_N(\hat{\theta}) + 2p)$$

Where N is the number of data and p is the number of parameters.

$$\begin{cases} x_1[k+1] = a_{12}x_2[k] \\ x_2[k+1] = a_{21}sin(x_1[k])a_{22}x_2[k] \\ a_{12} = 2.2 \\ a_{21} = \frac{g}{l} = -4 \\ a_{22} = -\frac{k}{m} = -0.34 \end{cases}$$

We want to calculate a_{12}, a_{21}, and a_{22} using the RLS method. First, we assume we do not know the original form of the coefficients. We start the system by setting $x_1[1]$ to 0 and $x_2[1]$ to 0, so x_1 and x_2 up to the 200th sample are obtained. In fact, a single initial excitation on x_1 is sufficient for x_1 and x_2 to be determined. After x_1 and x_2 are determined, we run the RLS algorithm separately on equations 1 and 2. In the first equation, x_1 is the output, and in the second equation, x_2 is the output.

The first RLS is performed as follows:

```
%% RLS_1
% Phi
Phi = zeros(N,1);
Phi(1,:)=[0];
for i=2:N
Phi(i)=x2(i-1);
end
phi = Phi';

% Initialize P
P(:,:,1) = q1*eye(1);

```

```
T_RLS = q2*ones(1,N);
T_RLS(:,1) = 0;
K = zeros(1,N);
K(:,1) = 0;
y = x1;
for t=2:N
p_inv = inv(P(:,:,(t-1))) + Phi(t,:)' * Phi(t,:);
P(:,:,t) = inv(p_inv);
K(:,t) = P(:,:,t) * Phi(t,:)';
T_RLS(:,t) = T_RLS(:,t-1) + K(:,t) * (y(t) - Phi(t,:) * T_RLS(:,t-1));
end
```

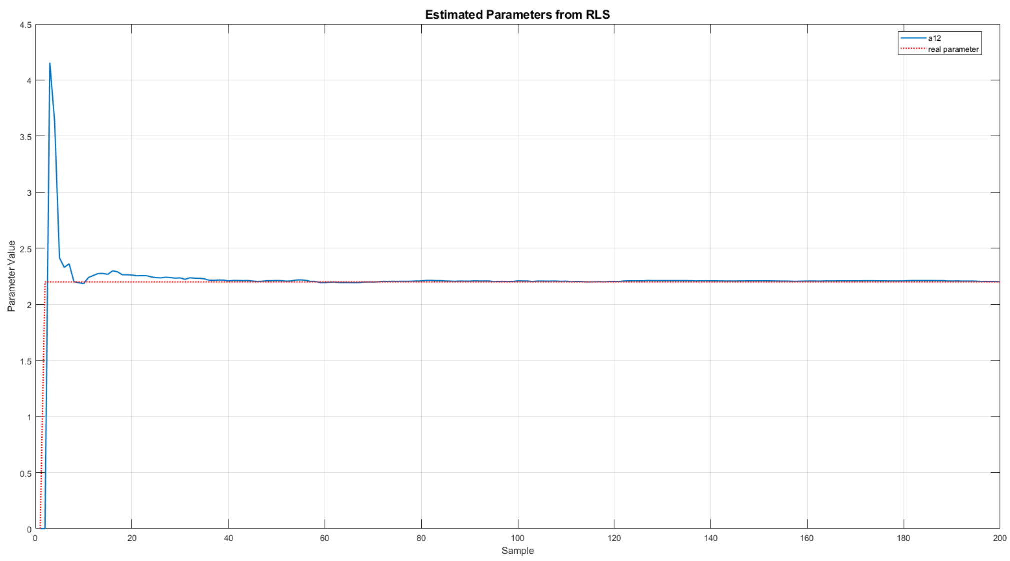


Figure 57: Estimated non linear parameters vs. real data using RLS method (Five-P-Resetting)

The second RLS is also performed as follows:

```
%% RLS_2
% Phi
Phi = zeros(N,2);
Phi(1,:)=[0 0];
Phi(2,:) = [sin(x1(1)) x2(1)];
for i=3:N
Phi(i,:)=[sin(x1(i-1)) x2(i-1)];
end
```

```
phi = Phi';

% Initialize P

P(:,:,1) = q1*eye(2);
P(:,:,2) = P(:,:,1);

T_RLS = q2*ones(2,N);
T_RLS(:,1) = 0;
T_RLS(:,2) = 0;

K = zeros(2,N);
K(:,1) = 0;
K(:,2) = 0;
y = x2;
for t=2:N
p_inv = inv(P(:,:,(t-1))) + Phi(t,:)' * Phi(t,:);
P(:,:,t) = inv(p_inv);
K(:,t) = P(:,:,t) * Phi(t,:)';
T_RLS(:,t) = T_RLS(:,t-1) + K(:,t) * (y(t) - Phi(t,:) * T_RLS(:,t-1));
end
```

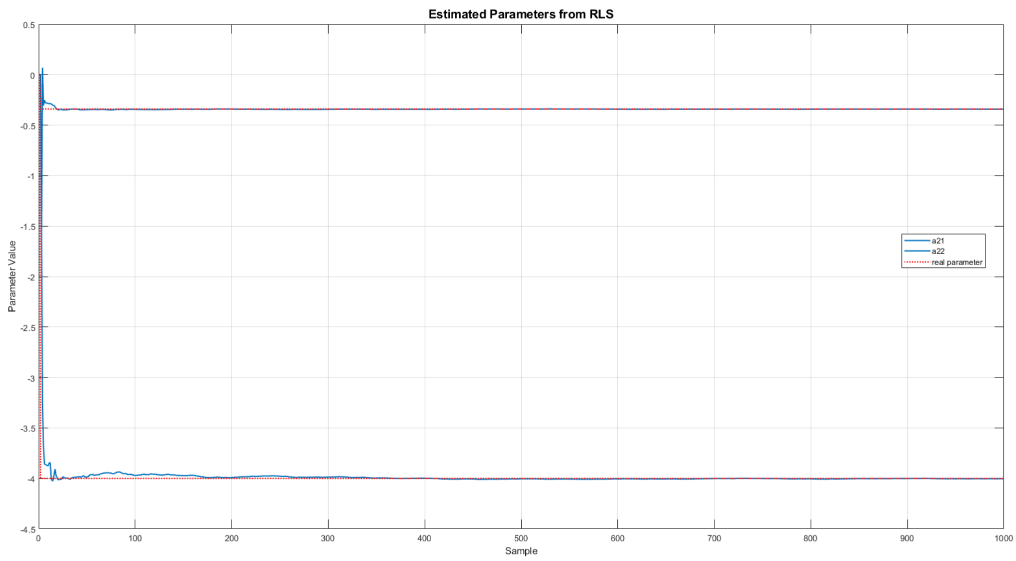


Figure 58: Estimated non linear parameters vs. real data using RLS method (Five-P-Resetting)

As it is visible in the plots the estimation is very good when we separate the estimation equations.