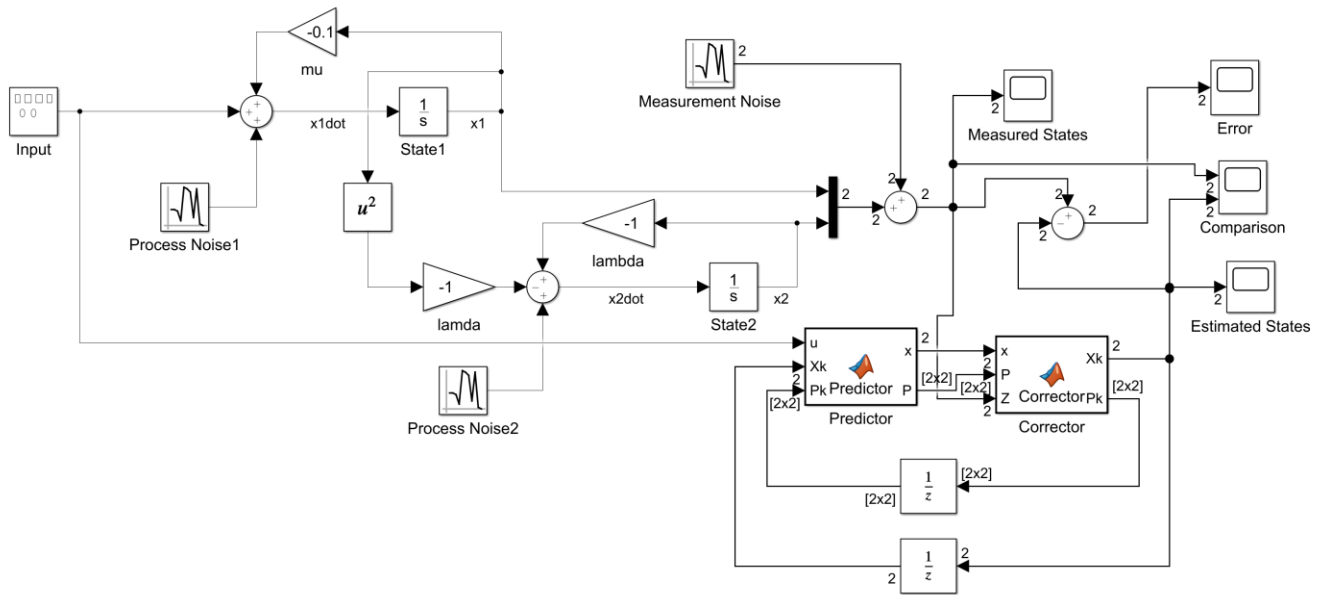


Simulink Diagram



Predictor Code

```
function [x,P] = Predictor(u,Xk,Pk)

mu=-0.1; lam=-1;

A = [mu 0;-2*lam*Xk(1) lam];

B= [1;0];

Ad=expm(A*0.01);

Bd=inv(A)*(Ad-eye(2))*B;

Q=eye(2);

F=[-A Q;zeros(2,2) transpose(A)];

G=expm(F*0.01);

Qd=transpose(G(3:4,3:4))*G(1:2,3:4);

x=Ad*Xk + Bd*u;

P=(Ad*Pk*transpose(Ad)+Qd);

end
```

Corrector Code

```
function [Xk,Pk] = Corrector(x,P,Z)

H=eye(2); R=0.1*eye(2); Rd=R/0.01;

K=P*transpose(H)*inv(H*P*transpose(H)+Rd);

Pk=(eye(2)-K*H)*P*transpose(eye(2)-K*H)+K*Rd*transpose(K);

Xk=x+K*(Z-H*x);

end
```

Noise Parameters

Block Parameters: Process Noise1

Random Number

Output a normally (Gaussian) distributed random number, repeatable for a given seed.

Parameters

Mean:

0

Variance:

1

Seed:

9

Sample time:

0.01

☒ Interpret vector parameters as 1-D

Block Parameters: Process Noise2

Random Number

Output a normally (Gaussian) distributed random number, repeatable for a given seed.

Parameters

Mean:

0

Variance:

1

Seed:

3

Sample time:

0.01

☒ Interpret vector parameters as 1-D

Block Parameters: Measurement Noise

Random Number

Output a normally (Gaussian) distributed random number, repeatable for a given seed.

Parameters

Mean:

0

Variance:

[0.1 0.1]

Seed:

[6 7]

Sample time:

0.01

☒ Interpret vector parameters as 1-D

Input Parameters

Block Parameters: Input

Signal Generator

Output various wave forms:

$$Y(t) = \text{Amp} * \text{Waveform}(\text{Freq}, t)$$

Parameters

Wave form: square

Time (t): Use simulation time

Amplitude:

3

Frequency:

1

Units: Hertz

☒ Interpret vector parameters as 1-D

Initial State Conditions

Block Parameters: State1

Integrator

Continuous-time integration of t

Parameters

External reset: none

Initial condition source: internal

Initial condition:

5

Block Parameters: State2

Integrator

Continuous-time integration of t

Parameters

External reset: none

Initial condition source: internal

Initial condition:

0

Block Parameters: Unit Delay

UnitDelay

Sample and hold with one s

Main State Attributes

Initial condition: eye(2)*5

Input processing: Elements

Sample time (-1 for inherited)

-1

Block Parameters: Unit Delay

UnitDelay

Sample and hold with one s

Main State Attributes

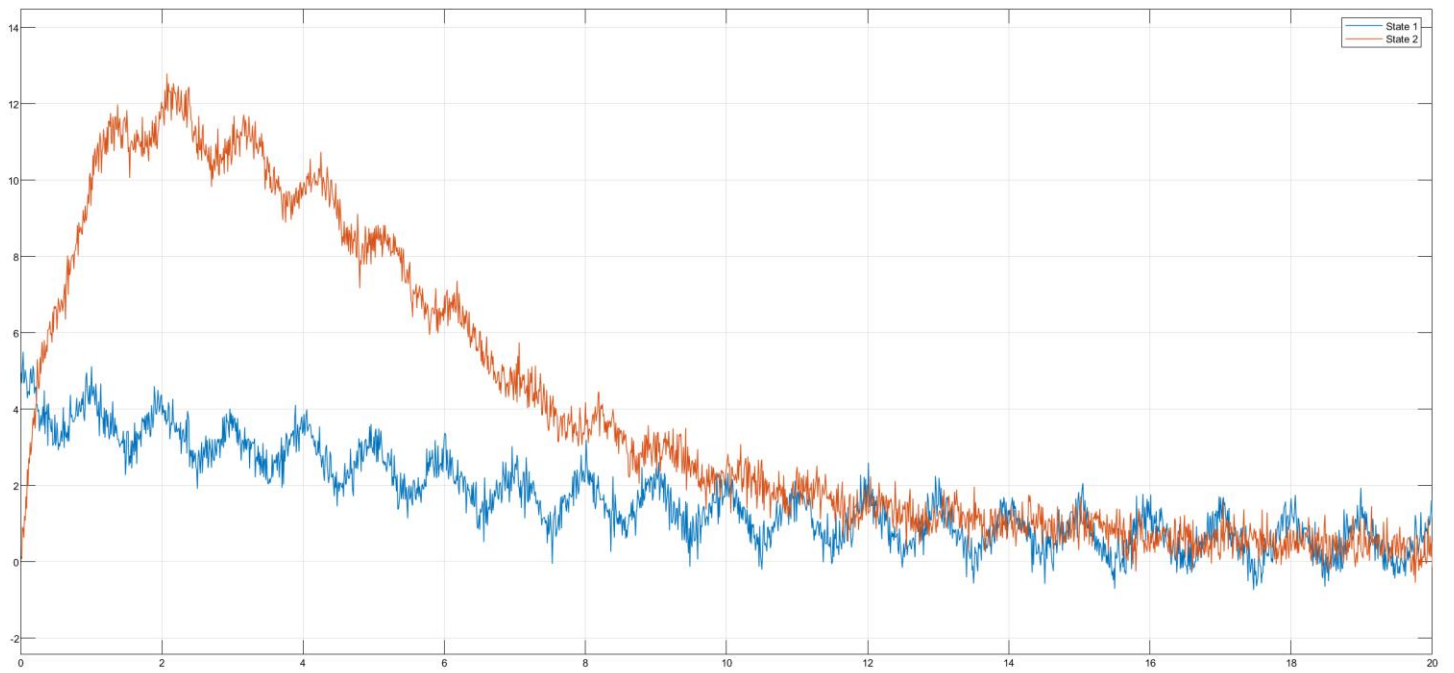
Initial condition: [5 0]

Input processing: Elements

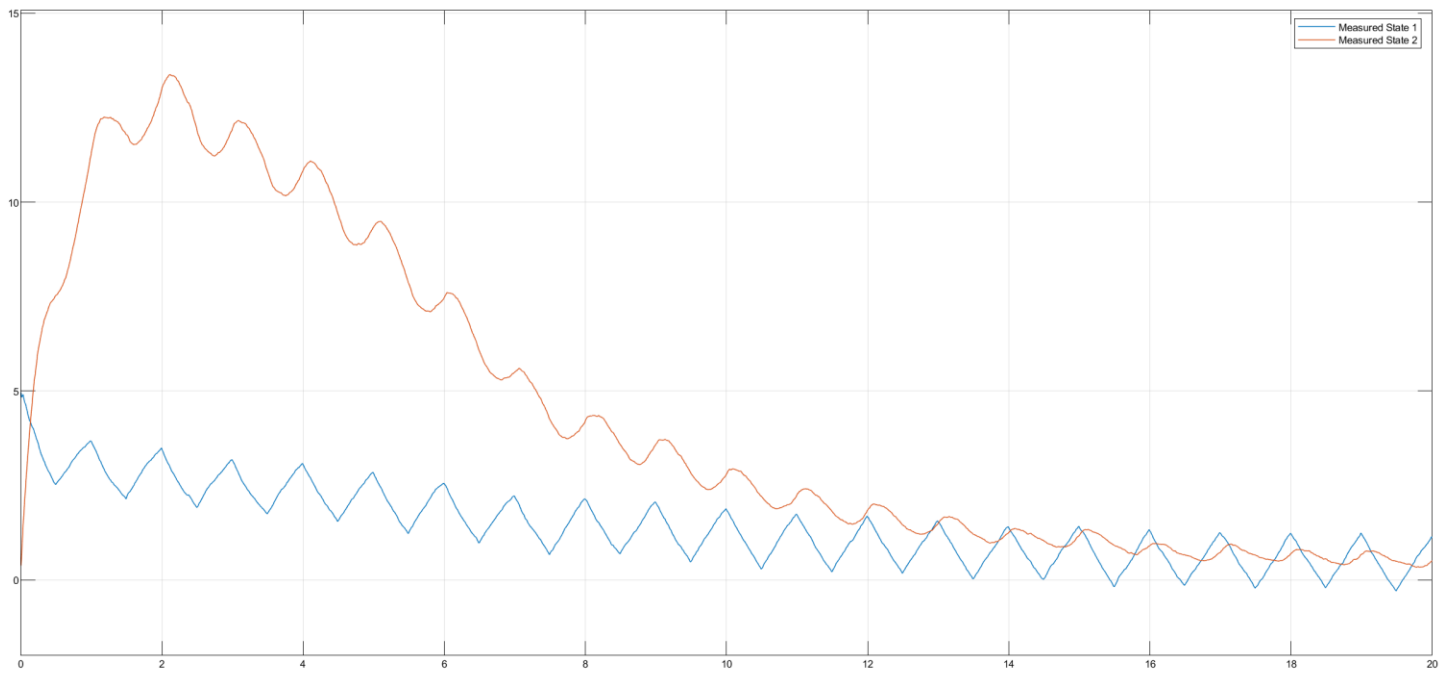
Sample time (-1 for inherited)

-1

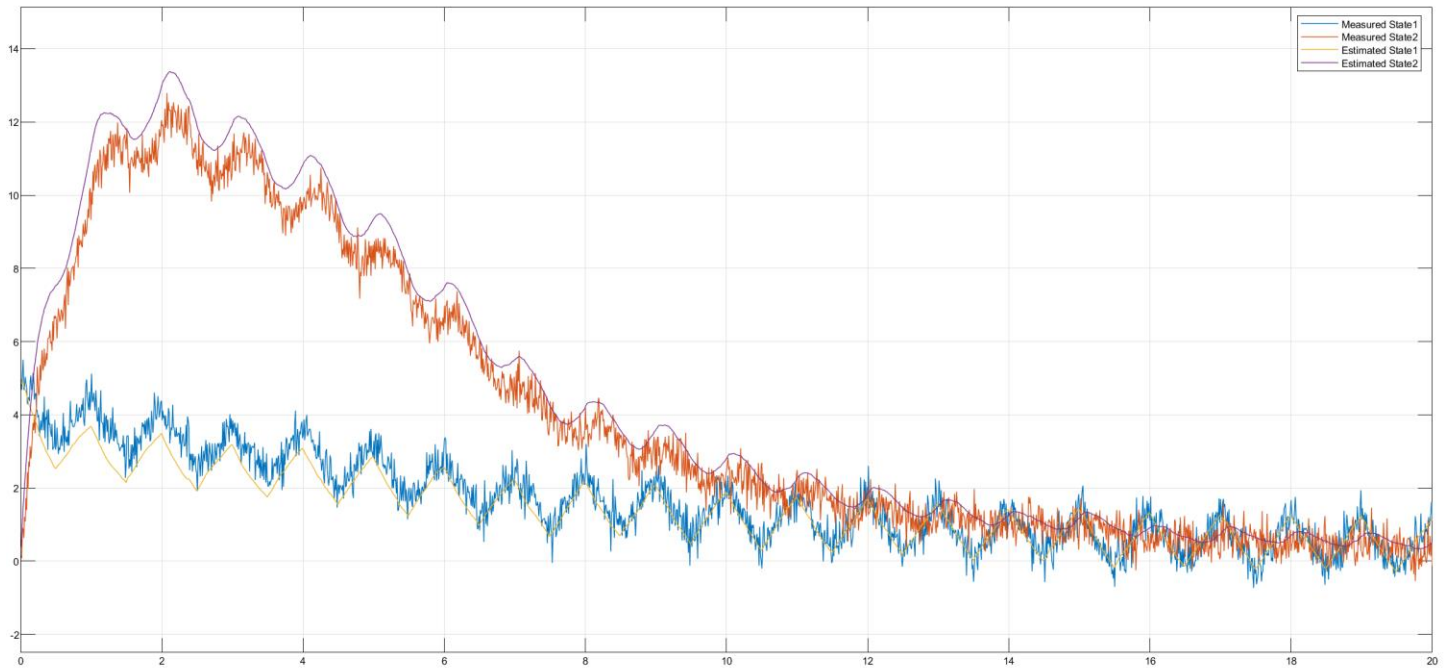
Measured States



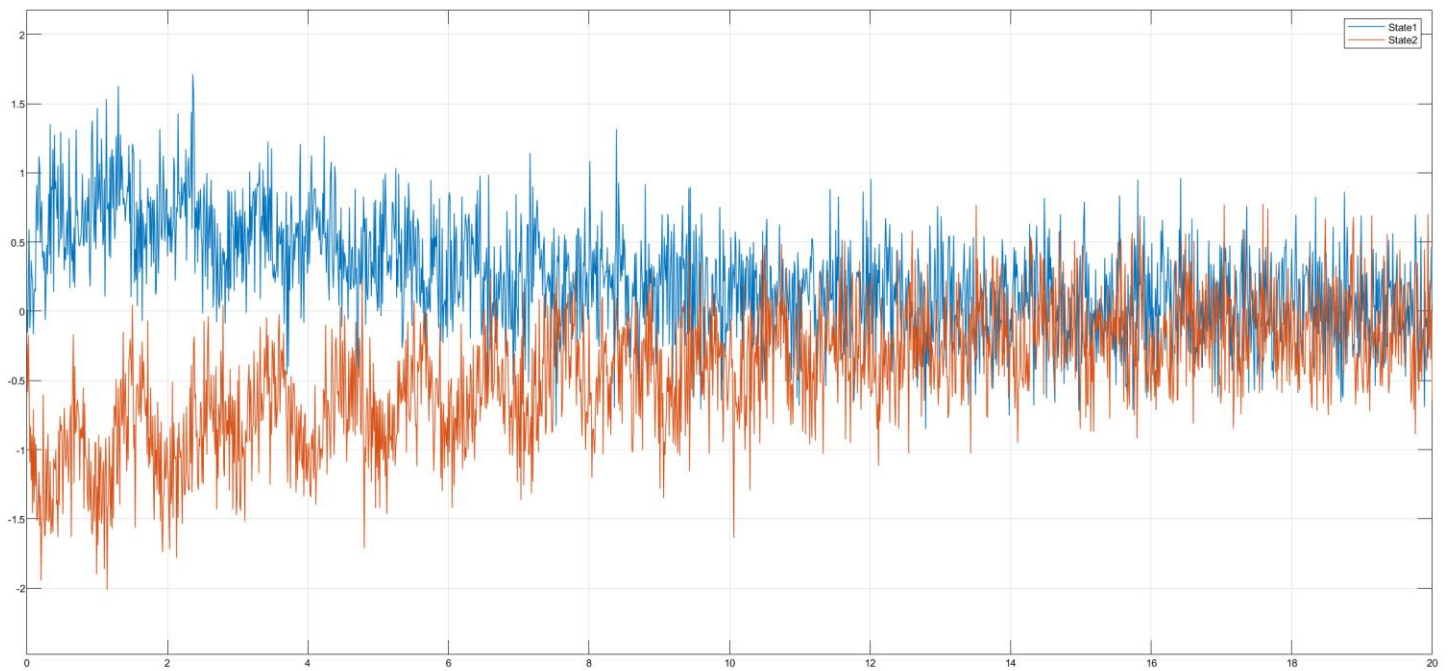
Filtered States



Comparison between measured and estimated states

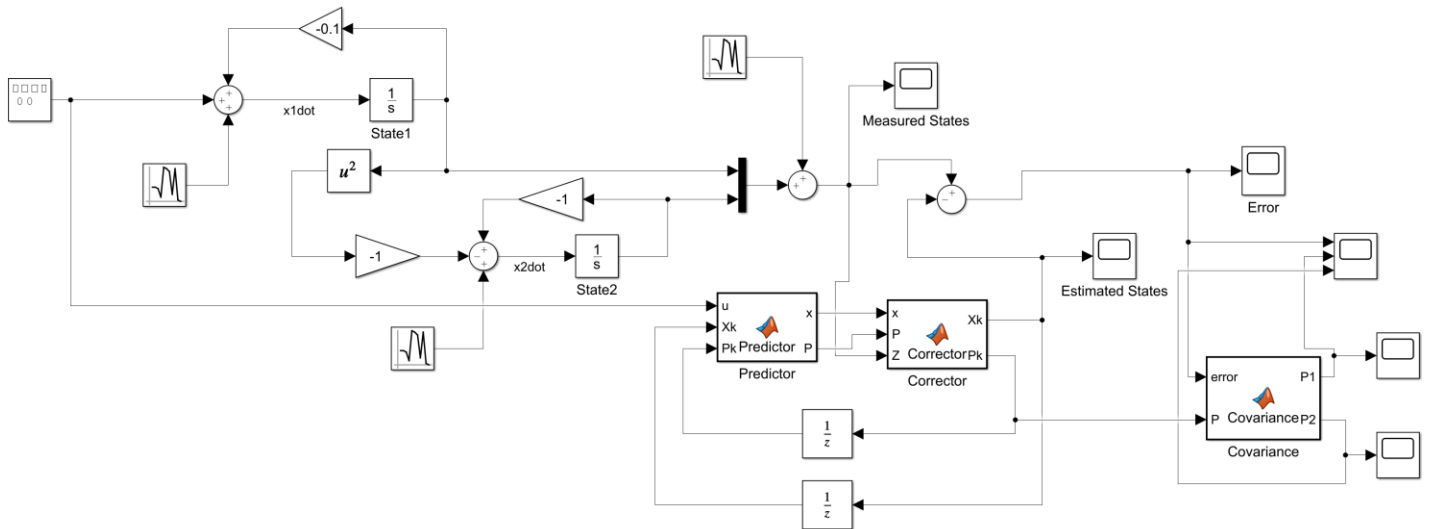


Error Plot



As observed from the comparison and error plots, the Kalman Filter is slightly inaccurate in the initial stages but it converges quickly within 10 seconds and brings the error values within the 0.5 range.

Simulink Diagram



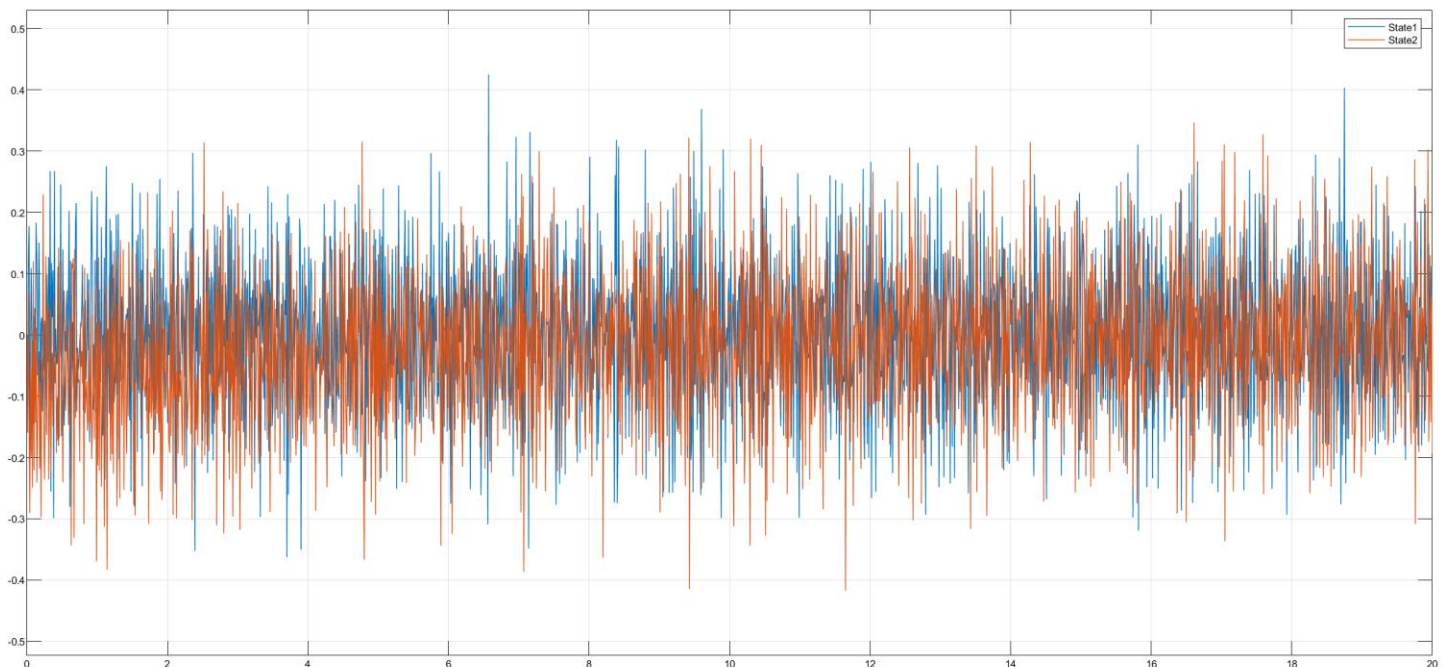
Covariance Function Code

```
function [P1,P2] = Covariance(error,P)
P1=[+sqrt(P(1,1)) -sqrt(P(1,1)) error(1)];
P2=[+sqrt(P(2,2)) -sqrt(P(2,2)) error(2)];
end
```

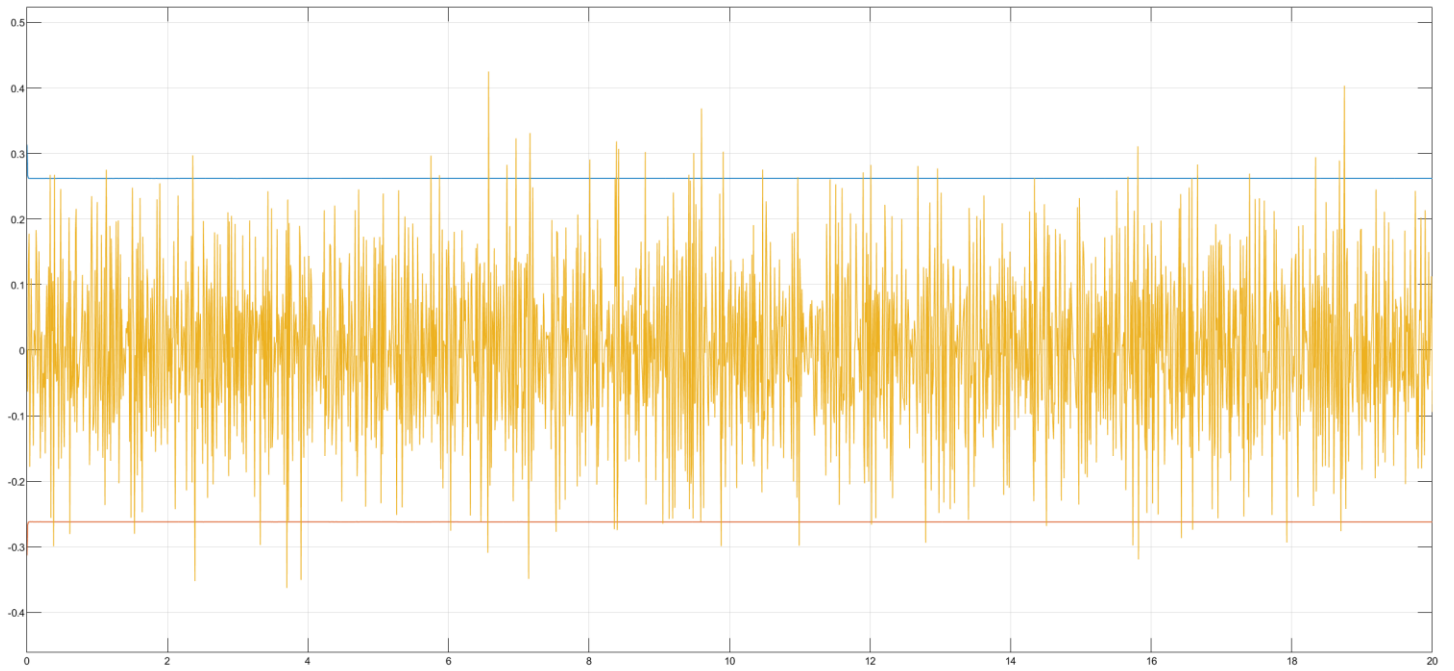
The following Q and R matrices have been used to get error within 0.5 and the error within the covariance about 80% of the time

```
Q=15*eye(2);
R=0.001*eye(2);
```

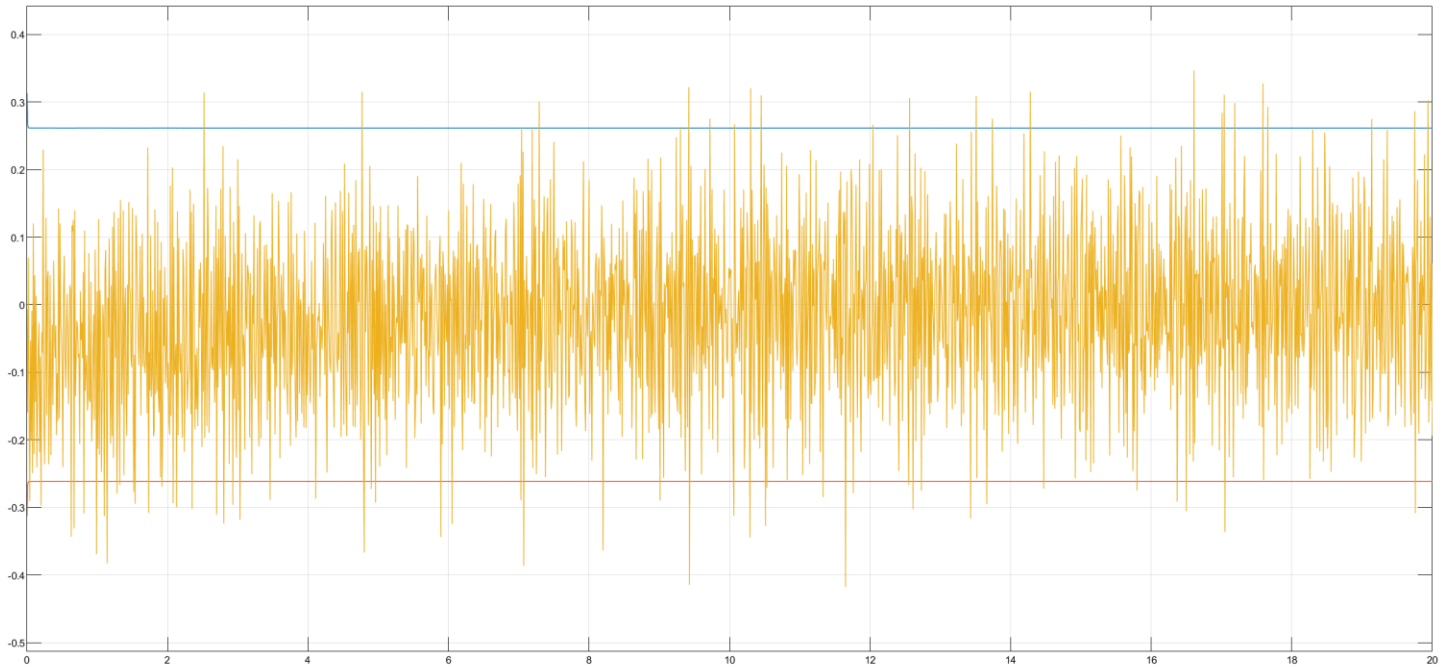
Error Plot (within 0.5)



Covariance Plot for State 1

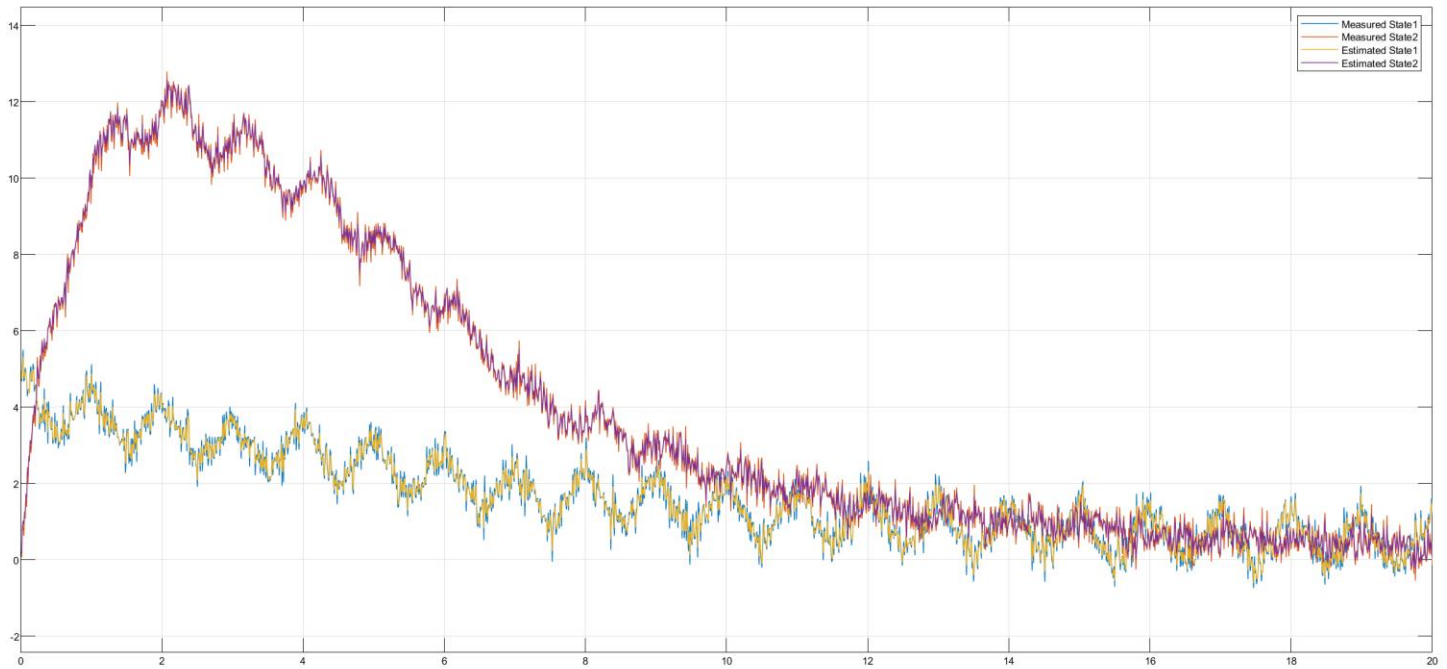


Covariance Plot for State 2



Note: The magnitudes cross the covariance lines multiple times. However, the error is within the covariance limits about 80% of the time. So, the average time of those peaks is around 20% of the total time.

Comparison Plots with new Q and R matrices



Now everything in the Simulink diagram is kept same, only the initial states of the filter are changed to $[0,0]$ for observing the effects on estimated values

Defining the initial states in Filter

Block Parameters: Unit Delay ×

UnitDelay
Sample and hold with one sample period delay.

Main State Attributes

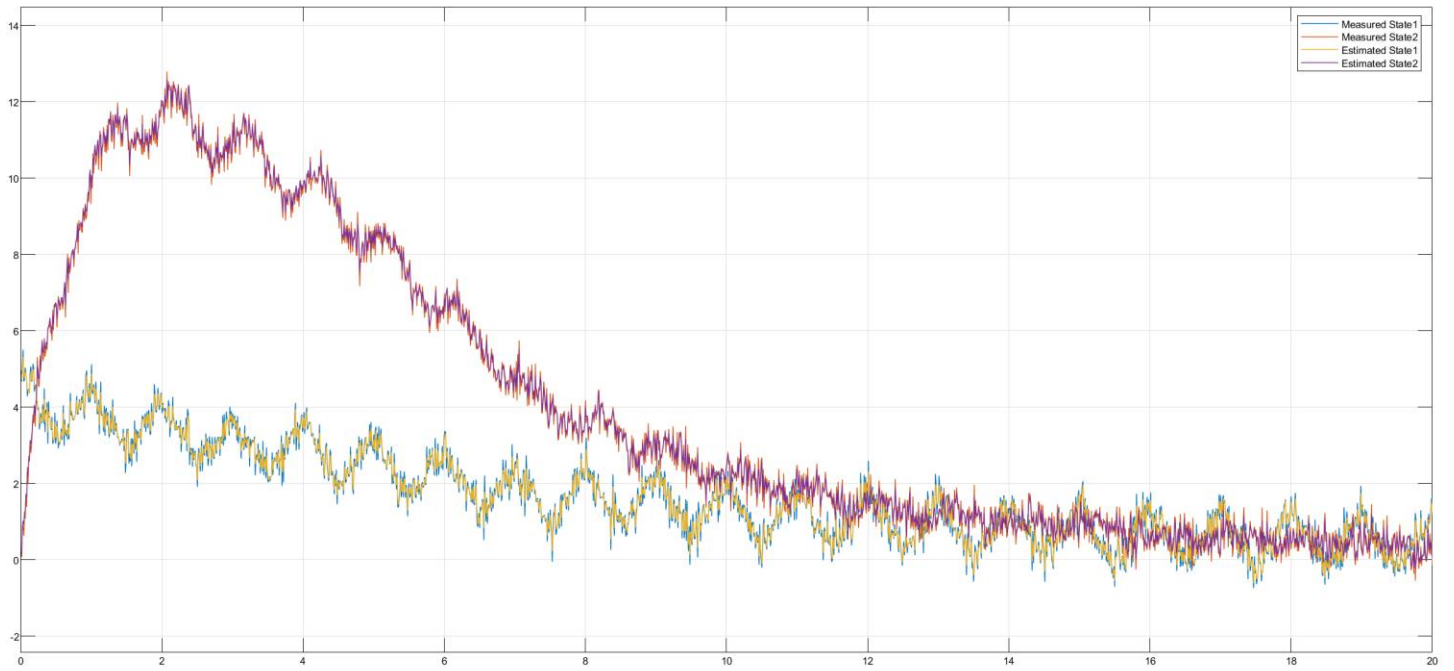
Initial condition:

Input processing: Elements as channels (sample based)

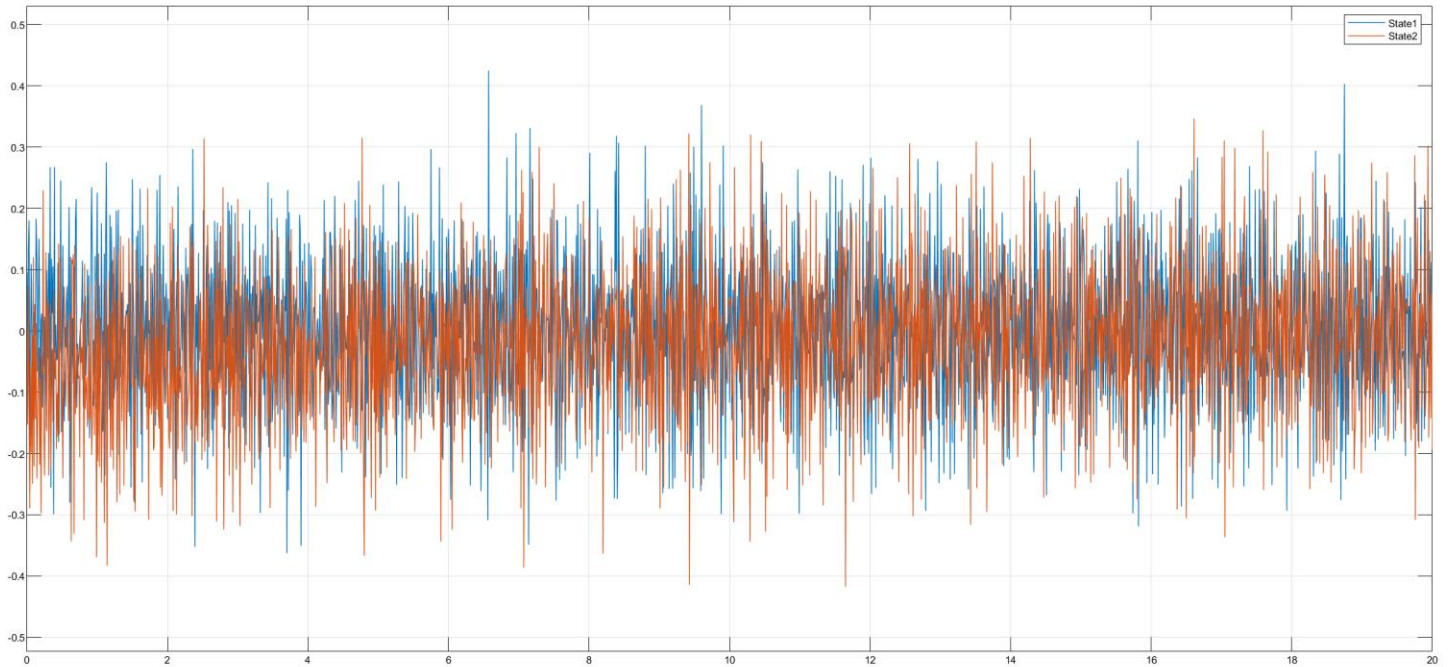
Sample time (-1 for inherited):

? OK Cancel Help Apply

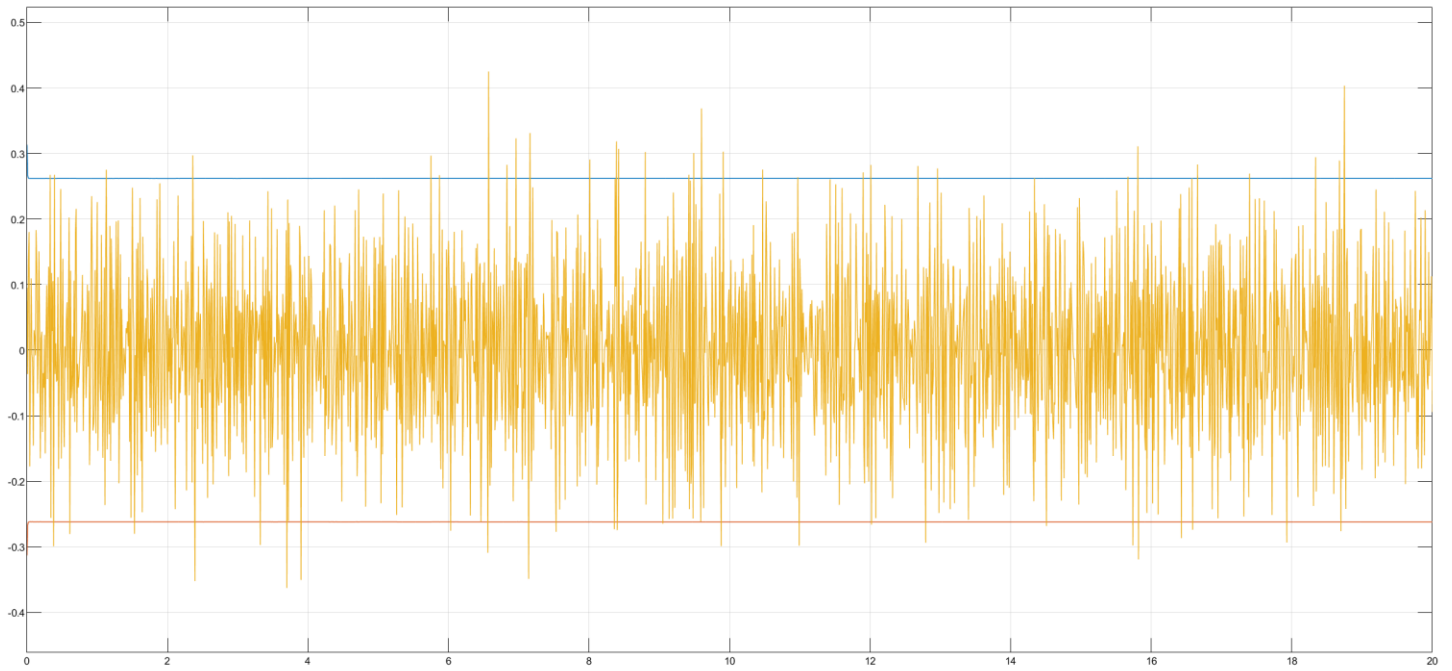
Comparison Plots for Measured and Estimated States



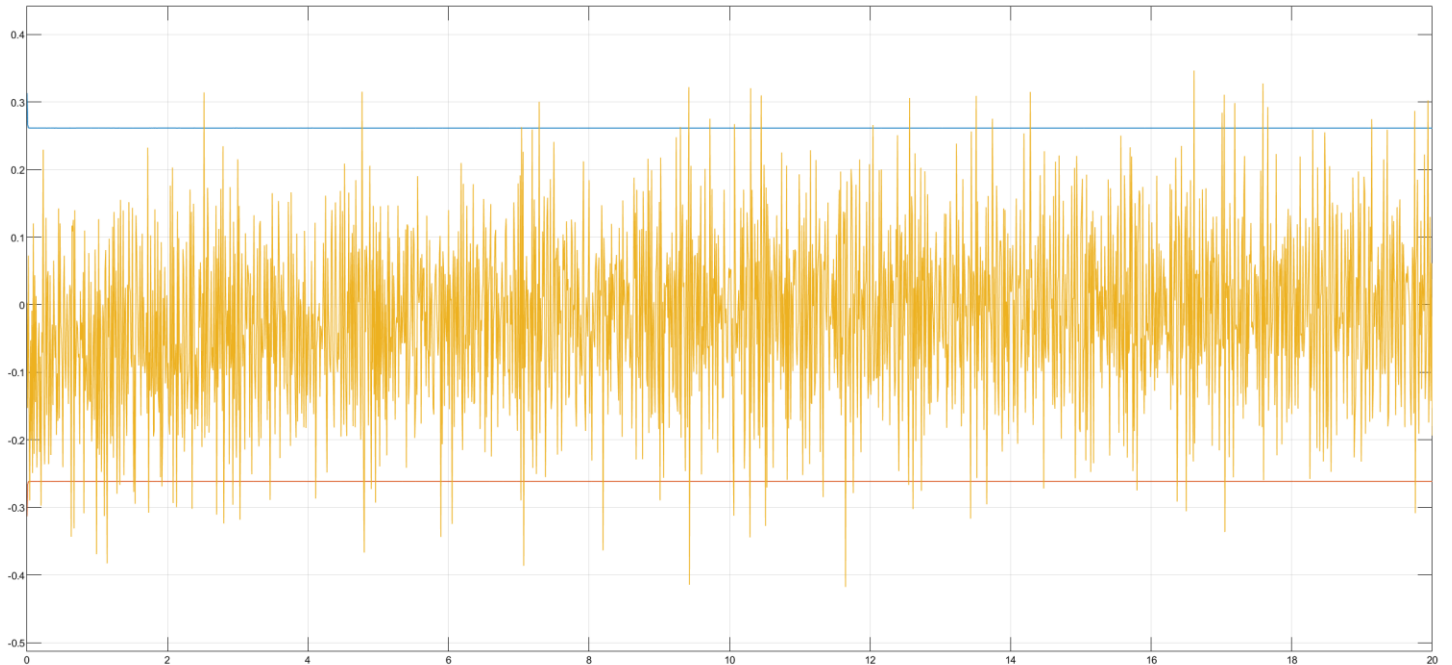
Error Plot



Covariance Plot for State 1



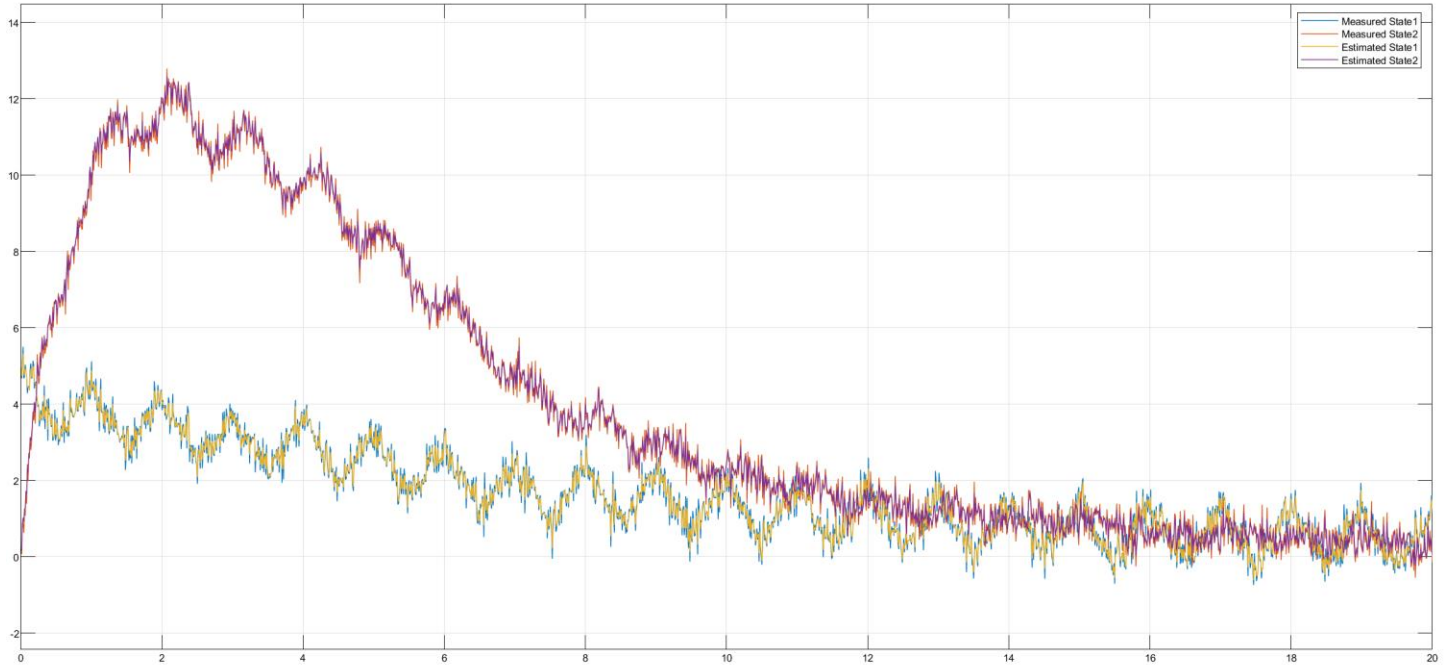
Covariance Plot for State 2



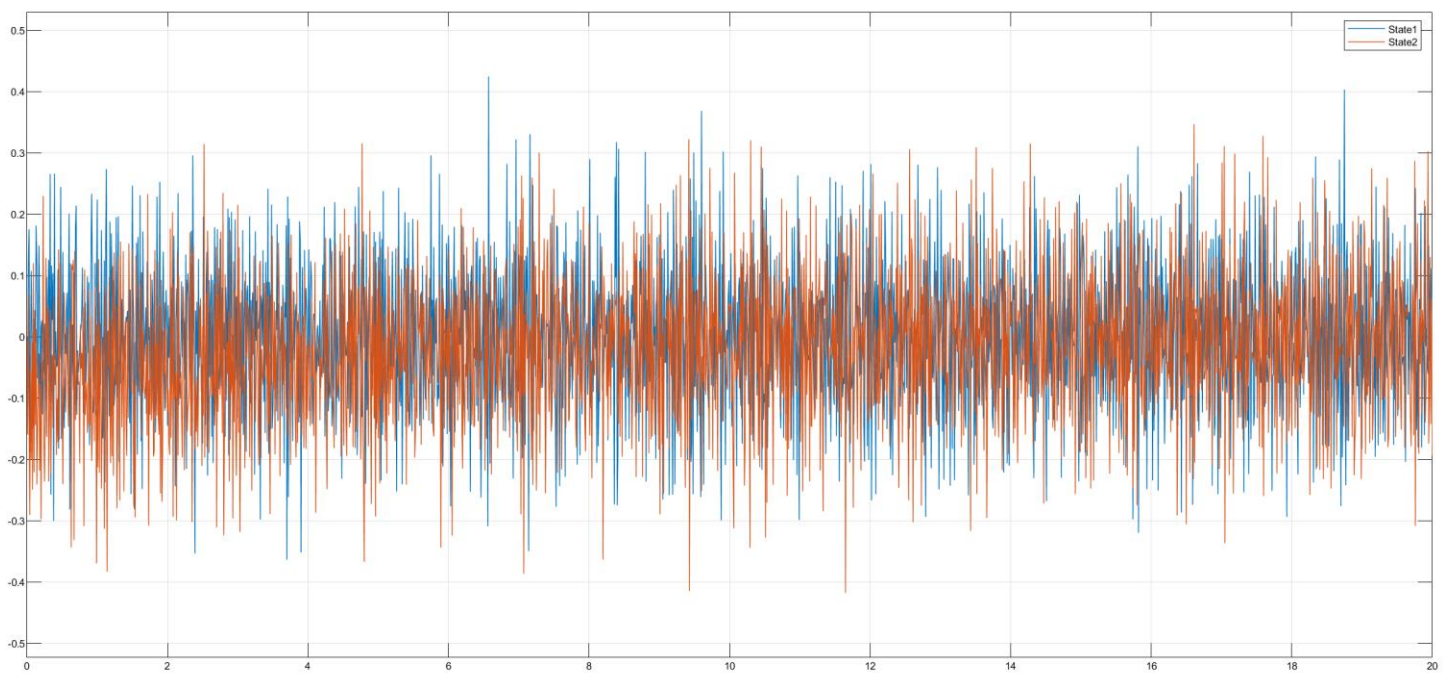
For this particular case/system, the change in initial value of first state from 5 to 0 has no noticeable effect on the output of the filter. For many systems, a significant change in the initial values of the filter may throw it off into another direction.

Now, only the value of μ is changed to -0.01 and the initial states are reverted back to $[5,0]$ and the simulation is rerun.

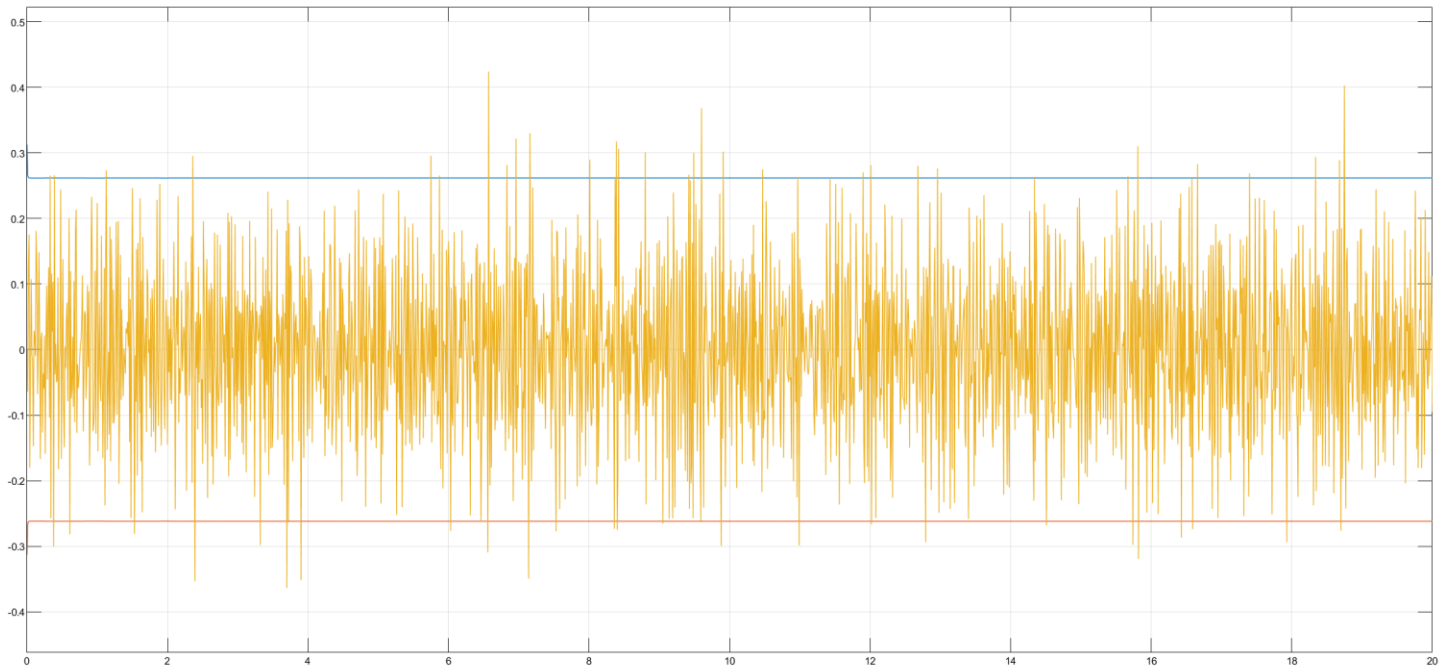
Comparison Plots for measured and estimated values



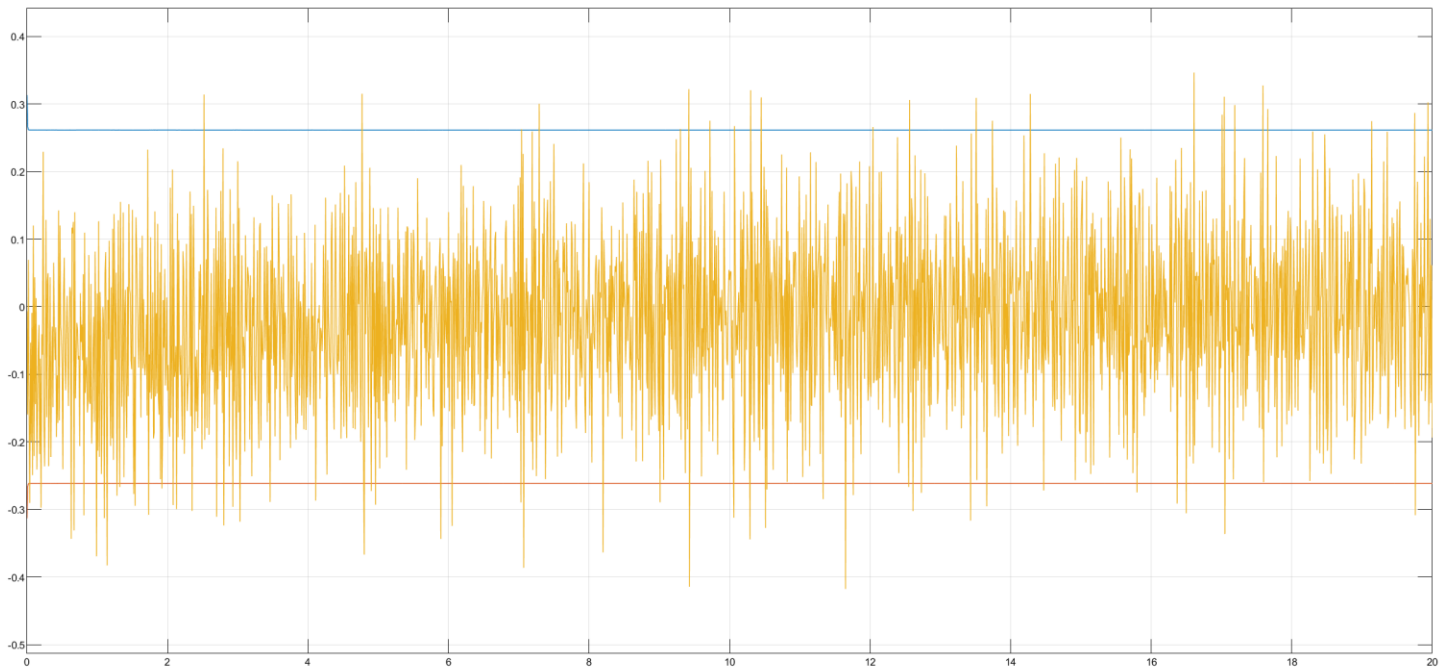
Error Plot



Covariance Plot for State 1



Covariance Plot for State 2



As observed from the above plots, there is no noticeable change in the output of the filter when the value of μ is changed from -0.1 to -0.01 in the predictor block and the initial states are reversed to the original values of $[5, 0]$.

This means that for this particular case, the filter does a good job even when the physics specified in the predictor block is slightly incorrect.

References

1. System Identification and Estimation – Dr. Michael A. Niestroy (Faculty, Electrical Engineering, The University of Texas at Arlington)
2. Intelligent Control Systems – Dr. Frank L. Lewis (Professor, Electrical Engineering, The University of Texas at Arlington)
3. Dynamic Systems Modeling and Simulation – Dr. David A. Hullender (Professor, Mechanical and Aerospace Engineering, The University of Texas at Arlington)