Homework 2C: Asynchronous Sampling Report

Timur Uzakov

The homework exercise is based on simulation model of a continuous-time process, the furnace. The transfer function is following:

$$Y(s) = \frac{1}{(1 + s * \tau)^2}(U(s) + D(s))$$

Disturbance is considered as a Wiener process, with intensity Q = 0.001. The property of the Wiener signal is that it is composed of a previous noise value plus drift, which is equal to intensity multiplied by square root of sampling time and a random number.

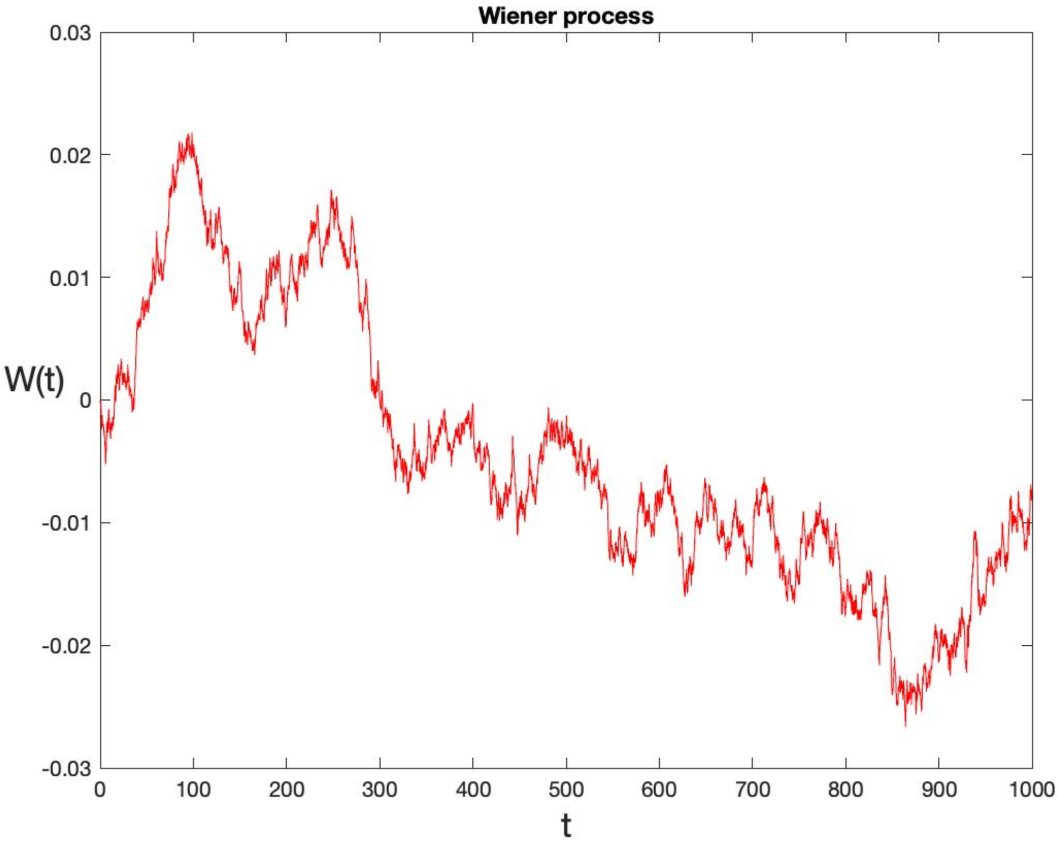

Fig. 1 - Example of Wiener Process

Problem 1: Finding continuous-time stochastic state-space mode

Problem Statement:

```
% Y(s) = 1/(1+s*tau)^2 *(U(s)+D(s))
% Find continusous-time stochastic state space model
% Create disturbance as a Wiener process
% Intensity of disturbance Q = 0.001
```

Solution

```
.
x = Ax + Bu + Gw       {State equation}
y = Cx + Du +  v       {Measurements}

[A,B,C,D] = tf2ss(1,[tau^2 2*tau 1]);
G = [1;0];
sys = ss(A,[B G],C,[D 0]);
step_input = ones(1,5000+1);
disturbance = [0,WienerNoise];
```

The stochastic state-space model has been found, B matrix has been augmented with vector G, which represents that there is noise in

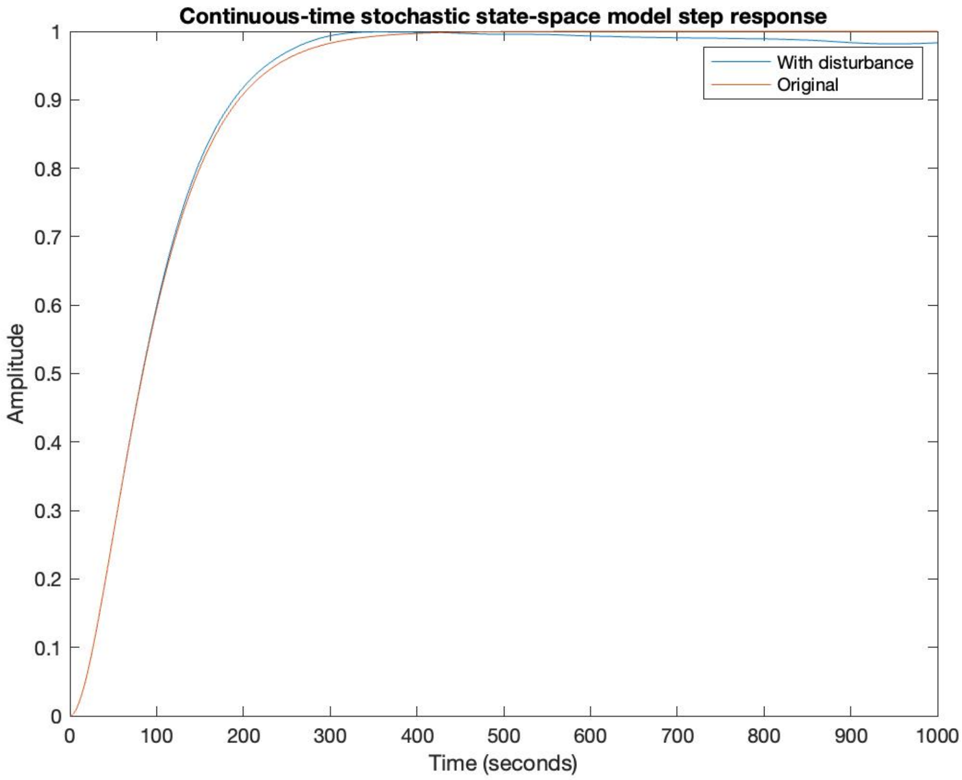

Fig.2 System response: original and with disturbance

input(disturbance), measurement noise was added later, when testing Kalman filtering.

The step response was obtained to observe the ground truth behaviour and behaviour with disturbance. In order to verify that my stochastic state-space model is correct, I have compared step responses of the model above and the one produced by the following function in Matlab:

```
sys = tf(1,[tau^2 2*tau 1]);
```

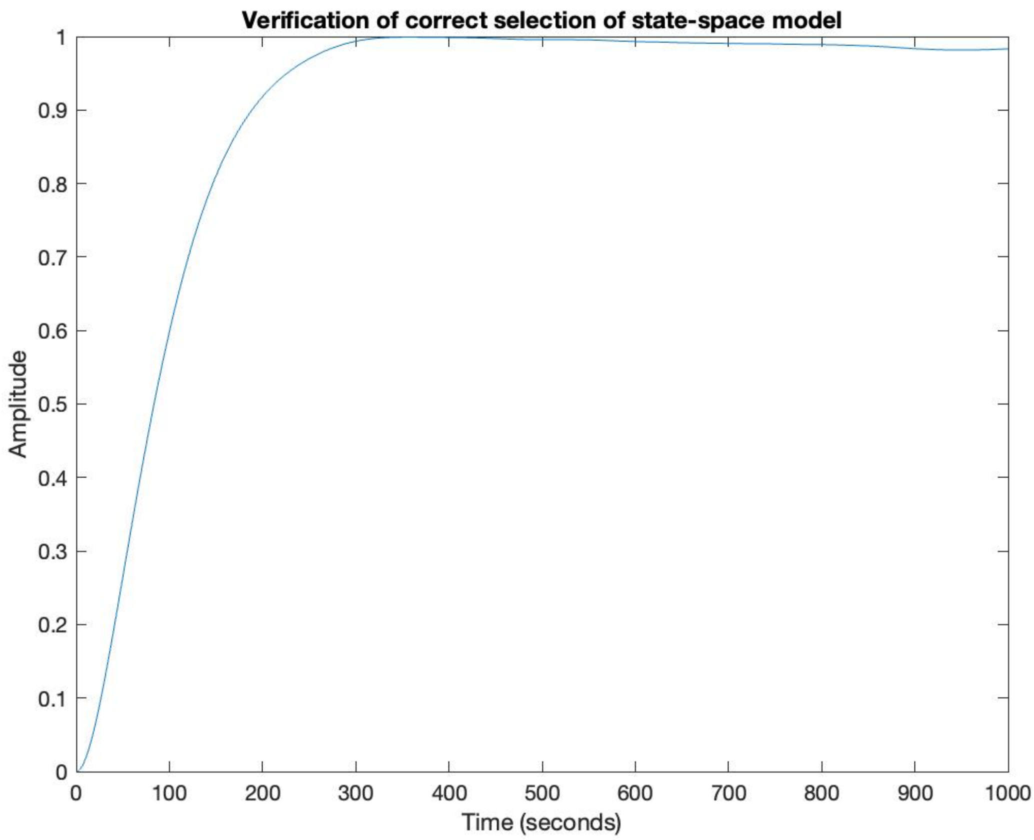

Fig. 3 Verification of model

By comparing Fig.2 and Fig.3 , it can be seen that my stochastic state-space model has identical response with disturbance to the model produced by tf command.

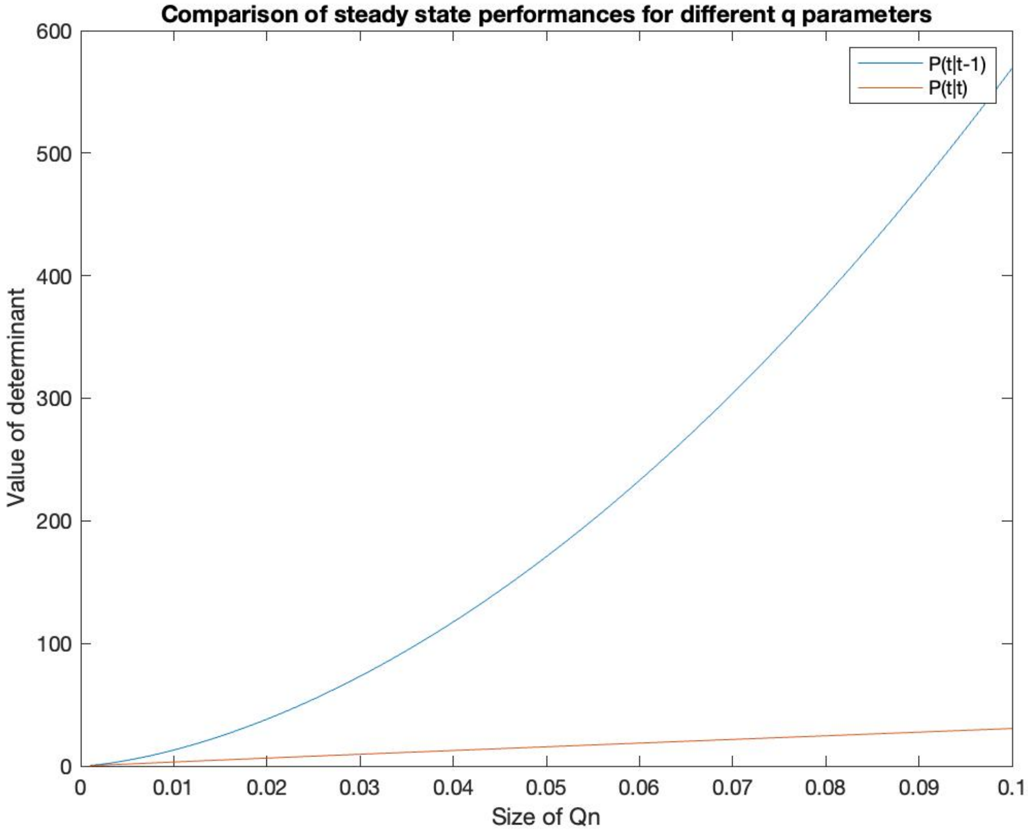

Fig. 4 Automatic finding of the most optimal weight, based on size of determinant of prediction co-variance

Problem 2: Synchronous discrete-time Kalman Filter design and evaluation of co-variances

Problem Statement:

```
% Find discrete-time Kalman-filter model
% Ts = 20s
% Evaluate predicted and filtered values
```

Solution:

The sampling time for discrete Kalman Filter is given and set to 20s. Measurement noise R was selected as 0.01^2 times identity matrix and Q was selected with varying weight from 0.001 to 0.1. It turned out that the smaller is weight Q, the smaller are resulting co-variances. (Fig. 4)

Kalman filter is designed using *kalmd* command, which returns system model *kest,* Kalman gain *L* and prediction *P* and filtering *Z* co-variances.

```
[kest,L,P,M,Z] = kalmd(sys,Qn,Rn,Ts);
```

The most optimal values are following:

P p(t|t-1)	1	2
1	*0.025*	*0.003*
2	*0.003*	*62.1751*

Z p(t|t)	1	2
1	*0.0250*	*0.0030*
2	*0.0030*	*62.0517*

Problem 3: Set of corresponding models and asynchronous sampling KF design

Problem Statement:

```
% Find Kalman Filter models
% Use asynchronous sampling with controller computation time
% Tc = 10/1/0.1/s
% Compare properties of noise models
% Evaluate P (predicted value)
```

Solution:

For solving this task, the equations for deterministic part and stochastic part of asynchronous sampling were obtained. (Slide 108 in the course's lecture slides)

$\epsilon = \dfrac{T_s - T_c}{T_s}$
$A_{async} = e^{A_c T_s}$
$B_{async} = \int_0^{T_s} e^{A_c \nu} d\nu B_c$
$C_{async} = C_c e^{A_c \epsilon T_s}$
$D_{async} = C_c \int_0^{\epsilon T_s} e^{A_c \nu} d\nu B_c$

$Q = \int_0^{T_s} e^{A_c\nu} Q_c e^{A_c^T\nu} d\nu$
$S = \int_0^{\epsilon T_s} e^{A_c\nu} Q_c e^{A_c^T\nu} C_c^T d\nu$
$R = \int_0^{\epsilon T_s} C_c e^{A_c\nu} Q_c e^{A_c^T\nu} C_c^T d\nu + R_c$

Equations from slides, asynchronous schema

For each of the controller sampling times the sets of models were constructed and predicted co-variance values have been presented.

To calculate Q,S,R , integration rules have been used, which resulted in matrices of sizes (2x2), (2x1) and (1x1), respectively.

$B_{async} = A^{-1}(e^{A_c Ts} - e^{A_c * 0})B_c$
$D_{async} = C_c A_c^{-1}(e^{A_c \epsilon Ts} - e^{A_c * 0})B$
$Q = Q_c(A + A^T)^{-1}(e^{(A+A^T)Ts} - e^{(A+A^T)*0})$
$S = Q_c(A + A^T)^{-1}(e^{(A+A^T)\epsilon Ts} - e^{(A+A^T)*0})C^T$
$R = R_c + C_c Q_c(A + A^T)^{-1}(e^{(A+A^T)\epsilon Ts} - e^{(A+A^T)*0})C^T$

Equations after integration, asynchronous schema

```
Ts = 20;
Tcs = [10 1 0.1];
B = [B G];
for i = 1:3
    Tc = Tcs(i);
    etta = (Ts-Tc)/Ts;
    Aasync = expm(A*Ts);
    Basync = inv(A)*(expm(A*Ts)-expm(A*0))*B;
    Casync = C*expm(A*etta*Ts);
    Dasync = C*inv(A)*(expm(A*etta*Ts)-expm(A*0))*B;

    Q = Qc*inv(A+A')*(expm((A+A')*Ts) -expm((A+A')*0));
    S = Qc*inv(A+A')*(expm((A+A')*etta*Ts) -expm((A+A')*0))*C';
    R = Rc + C*Qc*inv(A+A')*(expm((A+A')*etta*Ts) -expm((A+A')*0))*C';
    disp("Tc")
    disp(Tcs(i))
    [kest,L,P,M,Z] = kalman(ss(Aasync,Basync,Casync,Dasync,Ts),Q,R,S);
    P
end
```

The resulting prediction co-variance values are following:

Tc = 10		
P	*0.0669 x 10^9*	*0.5210 x 10^9*
	0.5210 x 10^9	*4.0548 x 10^9*

Tc = 1		
P	*0.6902 x 10^4*	*0.9280 x 10^4*
	0.9280 x 10^4	*4.0247 x 10^4*

Tc = 0.1		
P	*0.0714 x 10^4*	*0.0960 x 10^4*
	0.0960 x 10^4	*1.0645 x 10^4*

Conclusion: the co-variance values are smaller when T_c is less than half of T_s

Problem 4: Illustrate the impact of neglecting the mutual covariance S between the process and measurement noise in the case of asynchronous sampling.

Problem Statement:

```
% Show impact of neglecting S in case of asynchronous sampling
```

Solution:

For solving this task, again an iteration loop has been constructed:

```
for i = 1:3
    Tc = Tcs(i);
    etta = (Ts-Tc)/Ts;
    Aasync = expm(A*Ts);
    Basync = inv(A)*(expm(A*Ts)-expm(A*0))*B;
    Casync = C*expm(A*etta*Ts);
    Dasync = C*inv(A)*(expm(A*etta*Ts)-expm(A*0))*B;
```

```
    Q = Qc*inv(A+A')*(expm((A+A')*Ts) -expm((A+A')*0));
    S = Qc*inv(A+A')*(expm((A+A')*etta*Ts) -expm((A+A')*0))*C';
    R = Rc + C*Qc*inv(A+A')*(expm((A+A')*etta*Ts) -expm((A+A')*0))*C';
    disp("Tc")
    disp(Tcs(i))
    [kest,L,P,M,Z] = kalman(ss(Aasync,Basync,Casync,Dasync,Ts),Q,R,S);
    disp("With S")
    P
    disp("Value of determinant with S")
    disp(det(P))
    disp("With S=0")
    [kest,L,P,M,Z] = kalman(ss(Aasync,Basync,Casync,Dasync,Ts),Q,R);
    P
    disp("Value of determinant without S")
    det(P)
end
```

At first part, it assigns S from formula and, at the second part, it sets S to 0 (which is default)

The comparison table is provided below:

Tc = 10					
With S			***Without S***		
P	*0.0669 x 10^9*	*0.5210 x 10^9*	***P***	*0.0669 x 10^9*	*0.5210 x 10^9*
	0.5210 x 10^9	*4.0548 x 10^9*		*0.5210 x 10^9*	*4.0548 x 10^9*
Determinant	*5.5126 * 10^10*			*5.5126 * 10^10*	

Tc = 1					
With S			***Without S***		
P	*0.6902 x 10^4*	*0.9280 x 10^4*	***P***	*1.1686 x 10^4*	*1.5715 x 10^4*
	0.9280 x 10^4	*4.0247 x 10^4*		*1.5715 x 10^4*	*6.5784 x 10^4*
Determinant	*1.9168 x 10^8*			*5.2181 x 10^8*	

Tc = 0.1					
With S			***Without S***		
P	*0.0714 x 10^4*	*0.0960 x 10^4*	***P***	*0.0767 x 10^5*	*0.1032 x 10^5*
	0.0960 x 10^4	*1.0645 x 10^4*		*0.1032 x 10^5*	*1.1069 x 10^5*
Determinant	*6.6826 x 10^6*			*7.4302 x 10^8*	

Conclusion: Inclusion of cross-correlation S improves prediction co-variance significantly. For example, the sampling is 10 times more accurate for T_c = 0.1.

Problem 5: Evaluate the filtered values of state covariance matrix P(t|t) using filter design for system model with de-correlated noise.

Problem Statement:

```
% Evaluate filtered P(t|t)
% Use filter design for system with not correlated noise (S=0)
```

Solution:

To solve this task, the appropriate section in lecture slides has been applied. Noise properties of KF model have been used as before, however, A,B and Q matrices were updated accordingly to formulas in the slides (115).

$A' = A - SR^{-1}C$
$B' = B - SR^{-1}D$
$Q' = Q - SR^{-1}S^T$
$R = R$

Equations from slides, de-correlated noise

```
Tc = Tcs(3);
etta = (Ts-Tc)/Ts;
```

```
Q = Qc*inv(A+A')*(expm((A+A')*Ts) -expm((A+A')*0));
S = Qc*inv(A+A')*(expm((A+A')*etta*Ts) -expm((A+A')*0))*C';
R = Rc + C*Qc*inv(A+A')*(expm((A+A')*etta*Ts) -expm((A+A')*0))*C';

A_decor = A - S*inv(R)*C;
B_decor = B - S*inv(R)*D;
Q_decor = Q - S*inv(R)*S';

R = R;

disp("State covariance matrix P(t|t) for system model with decorrelated
noise")

[kest,L,P,M,Z] = kalman(ss(A_decor,B_decor,C,D,Ts),Q_decor,R);
Tc
Z
det(Z)
```

The following are resulting filtered values:

Tc = 0.1		
Z	*9.1118 x 10^4*	*1.6578 x 10^4*
	1.6578 x 10^4	*5.5383 x 10^4*
Determinant	*4.7715e+09*	

Conclusion: the best values of filtering co-variance $P(t\,|\,t)$ is achieved with small T_c = 0.1s.

Confirmation of working of KF

It was interesting to depict and verify if the Kalman filter does work. Two cases were tested: synchronous version and asynchronous version. (Fig.5 and Fig.6)

Conclusion: lessons learned

From the straight line of the graph on Fig.6 , it is possible to conclude that asynchronous sampling can withstand the effect of disturbance on the input signal as well as filter out the noise on the measurement signal.

The best performance is achieved with T_c set to a value that is less than half of the filtering period, that is 1 or 0.1 s. It is very important to include cross-correlation (not to set it to 0), because the accuracy becomes better by 10 times.

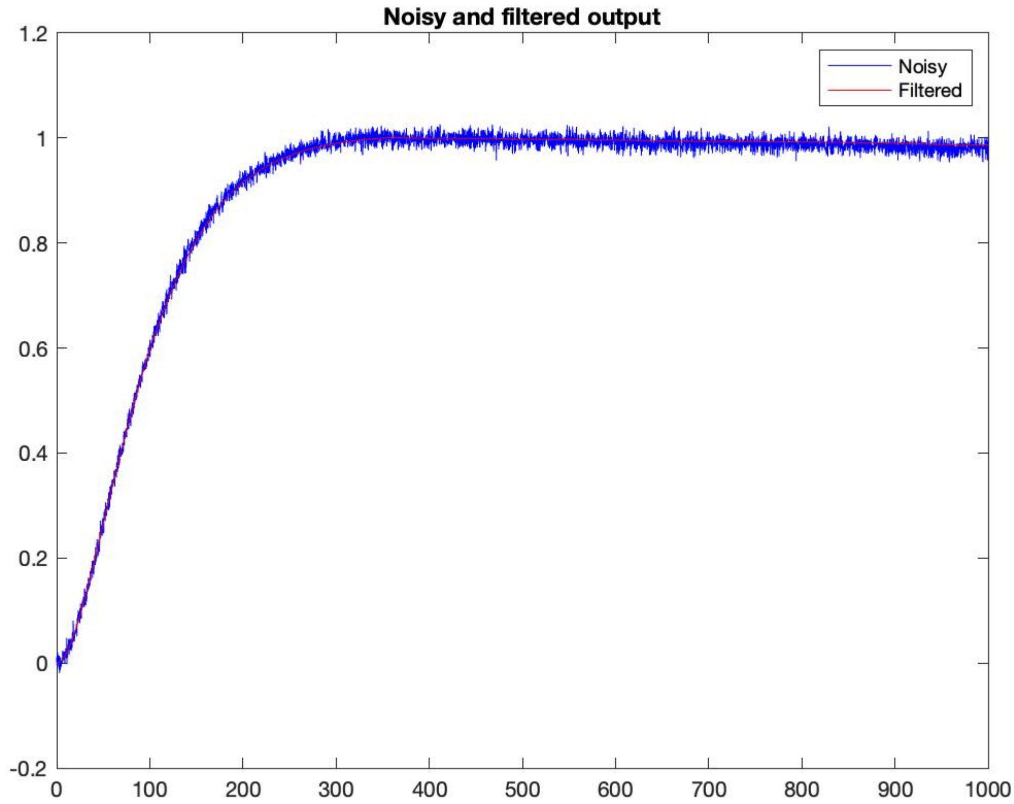

Fig.5 Filtering process: synchronous

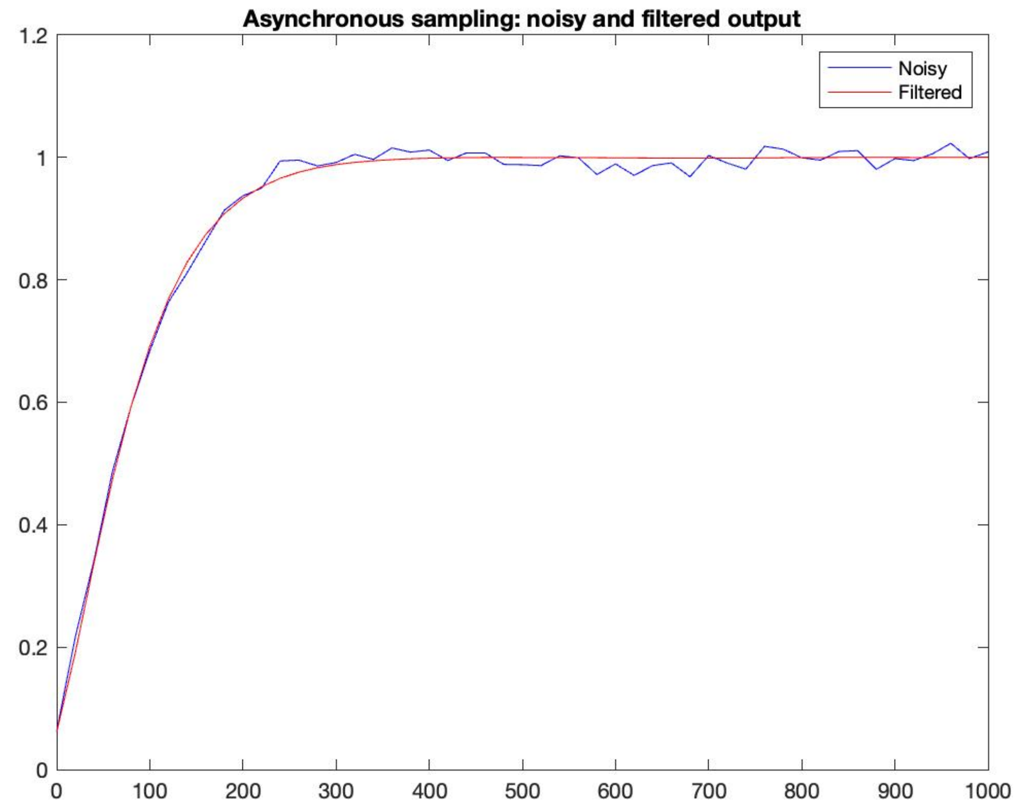

Fig. 6 Asynchronous filtering