Home assignment\_4\_ Reference

1. The state equation

The state space model with simplifying parameters

Where

\*\*\* Measurement noise:

* Encoder bit

The example is introduced in the textbook. The book was written in 1970… What cause to the measurement noise? The main is the angular sensor’s resolution(of course there are many noise sources as electric interference and so no..)

1. I transform the rad into degree as

sqrt(10^-7)\*180/3.14 = 0.0181(degree)

1. In order to get the error how many bits you need

360/2^16+ 360/2^15 = 0.0165 > 0.0181(the error)

1. I conclude in order to get

I need at least 16 bit angle encoder so that the least two bits are error..

1. Wonderful…He, the author of the textbook, introduces 16 bit encoder in 1970!!Amazing.
2. An encoder is very good with respect to noise. How others sensors? These sensors are more corrupted by the noise compared to an encoder

* Remote sensors such as GPS, the output is corrupted by noise so that You may not distinguish the signal and noise. Or the velocity gun to check a car velocity.
* Or inertial sensors such as accelerometers, angular accelerometers (gyro), they are not good to encoders.
* Any sensors noise should be confirmed by its manufacturers or you have to test.
* I said in my material, in order to model a real physical system into a mathematical model,

You should use the parameter values in the same unit. So in this example in MKS

The angle unit: **radian. Not degree**. It should be remembered. If the unit is different, your design, even if the algorithm is correct, will be useless!!

1. The Stability

The system is **asymptotically stable** compared to the text book. So without input, for any initial conditions, the states are convergent to zeros.

1. The system trajectory without the output noise

clearall; clc;clf

% system paramters

a = 4.6;

% system dynamics

A = [ 0 1;-1 -a];

B = [0 1]';

C = [1 0];

gam = 0.1;

Vdg=gam^2\*10; % the intensity of the disturbance

Vm =10^-7; % the noise intensity

ka = 0.787;

% the plant state space model

sys= ss(A,B,C,0);

% an trajectory example with an initial point

x0 =[-1;-1];

y0 = C\*x0; %inital values of the plant states

N = 1000; % the sampling number of data

tf = 5 ; % the final time

t = linspace(0,tf,N);

vd = Vdg\*randn(N,1); % gaussian noise ... if awgn is available, use agwn (it stands for

% additive white gaussian noise

[y,t,x]=lsim(sys, zeros(N,1), t,x0); % without disturnbance

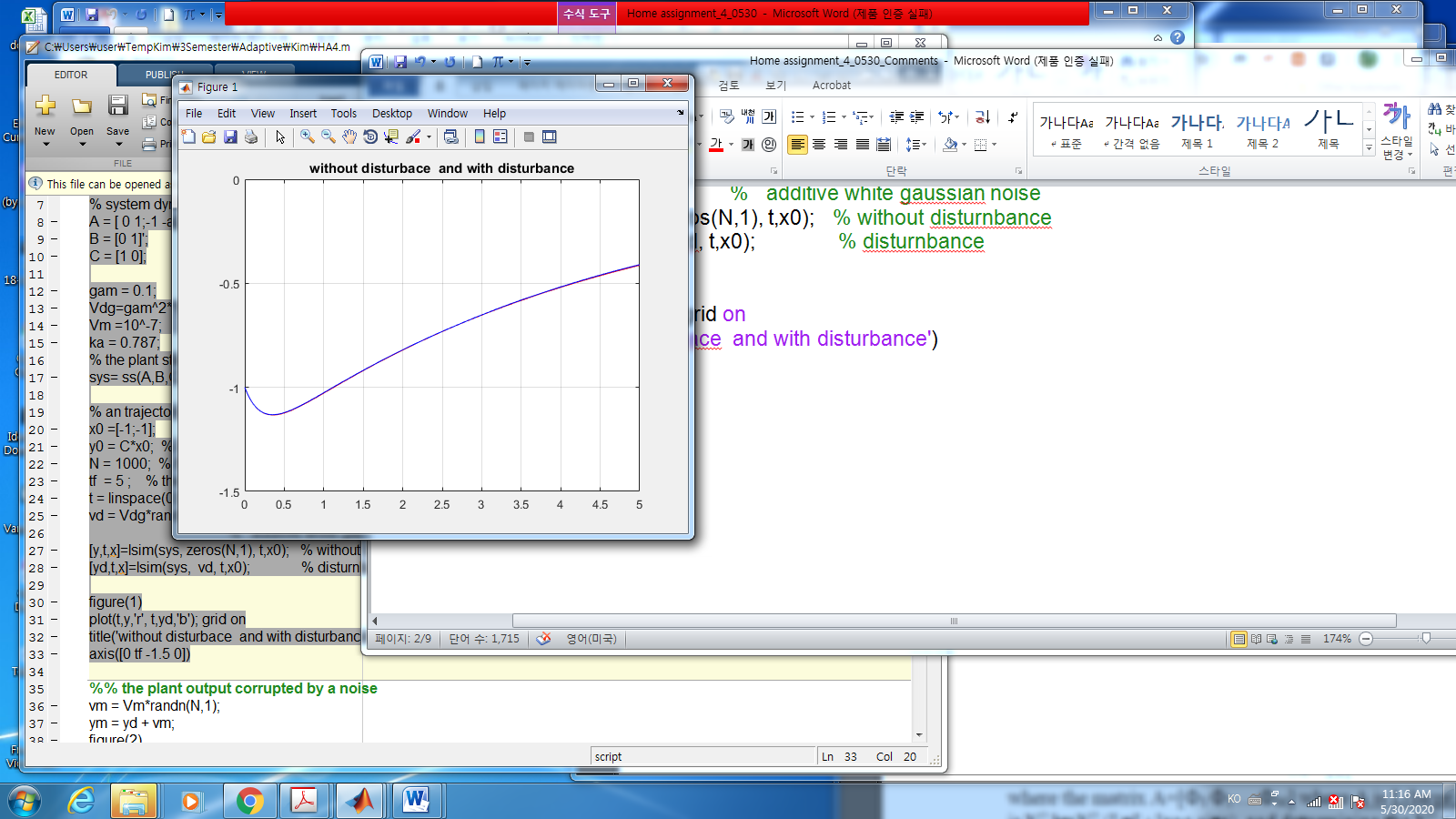
[yd,t,x]=lsim(sys, vd, t,x0); % disturnbance

figure(1)

plot(t,y,'r', t,yd,'b'); grid on

title('without disturbace and with disturbance')

axis([0 tf -1.5 0])



%%%%%%%------------- Output without output noise

You may see the output converge to zero. If you change the system matrix as the poles are more negative, the output converges to zero faster independent of the initial point x0.

--------------------------%%%%%%%%%%%%%%%%%%%%%%%

* Output with the output noise

%% the plant output corrupted by a noise

vm = Vm\*randn(N,1);

ym = yd + vm;

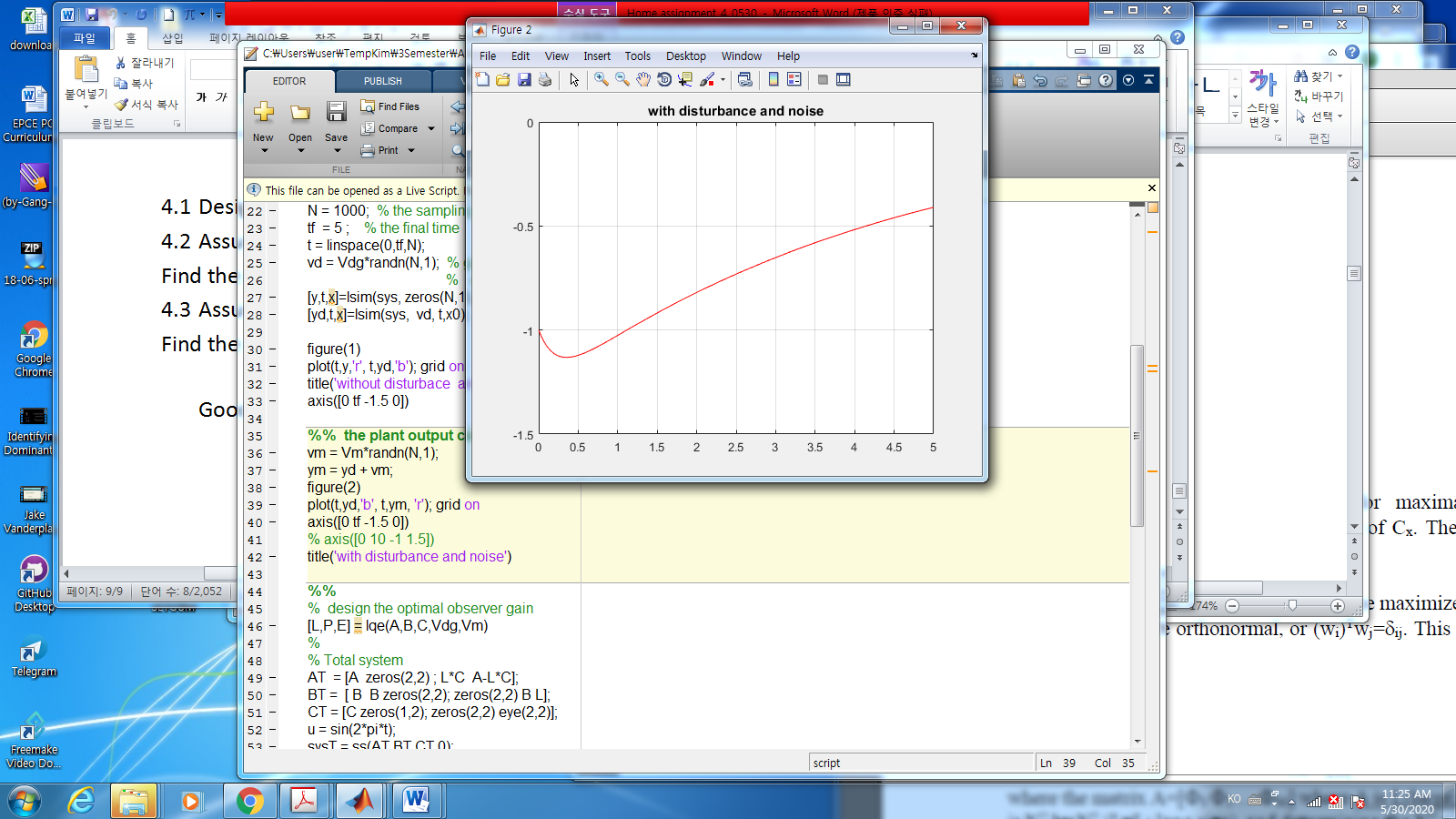
figure(2)

plot(t,yd,'b', t,ym, 'r'); grid on

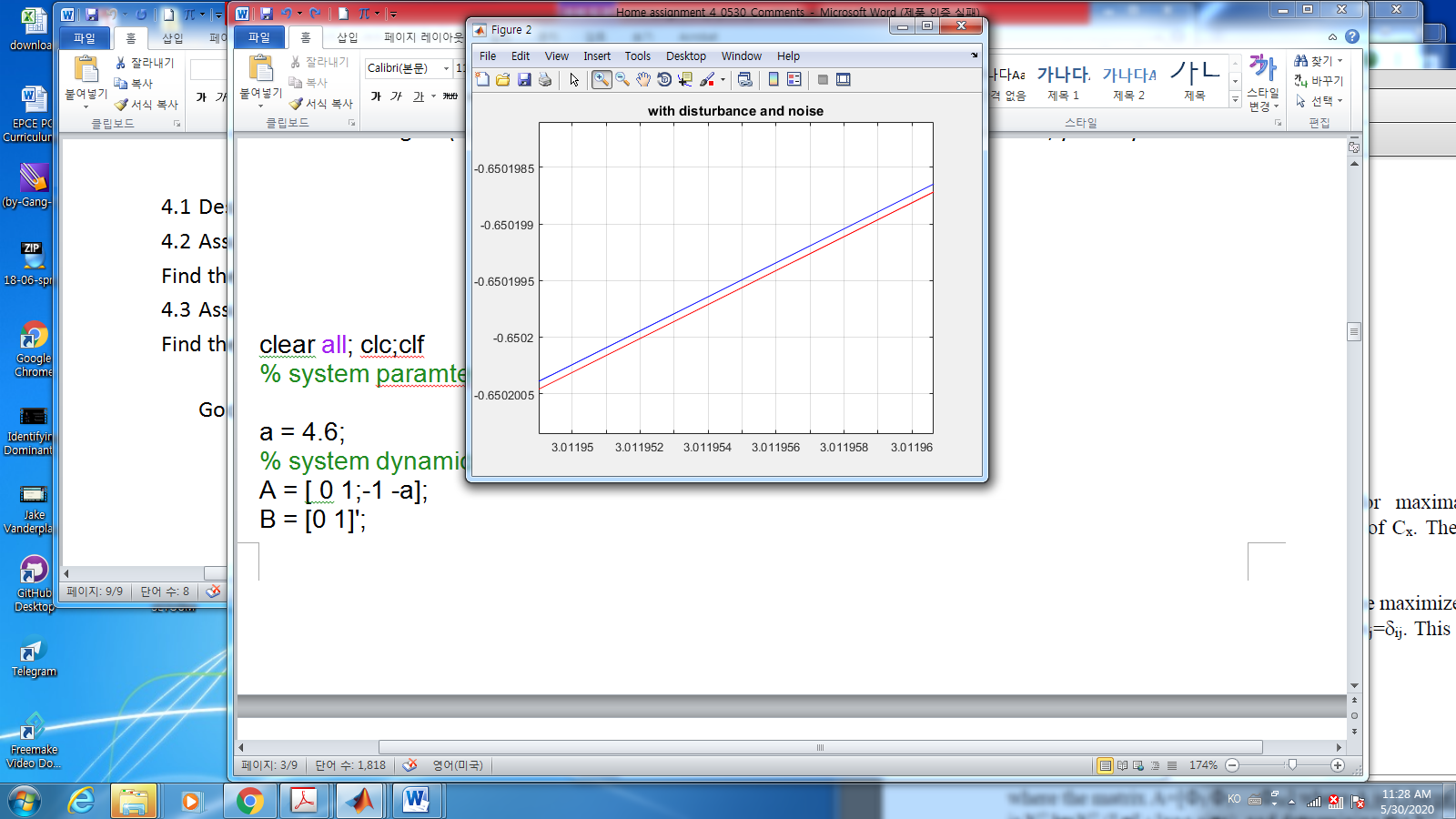
axis([0 tf -1.5 0])

% axis([0 10 -1 1.5])

title('with disturbance and noise')



%%%%%%%%------ the effect of output noise

 In the figure (2) there may be the just one trajectory, but you may extend the axis, you may see

Yap. There are two trajectories. Remember **the y-axis unit is radian**.

1. The observer design
   1. the observer gain

% design the optimal observer gain

[L,P,E] = lqe(A,B,C,Vdg,Vm)

Simple. Very simple command due to matlab.

* 1. The total system simulation
     1. when input is zero with initial observer

% Total system

AT = [A zeros(2,2) ; L\*C A-L\*C];

BT = [ B B zeros(2,2); zeros(2,2) B L];

CT = [C zeros(1,2); zeros(2,2) eye(2,2)];

u =zeros(N,1);

% u = sin(2\*pi\*t);

sysT = ss(AT,BT,CT,0);

%uT = [zeros(N,1),vd ,zeros(N,1), vm];

uT = [u,vd ,u, vm];

[yTd,t,x] = lsim(sysT, uT, t,[x0; 0;0]); % input with u = 0 and disturbance

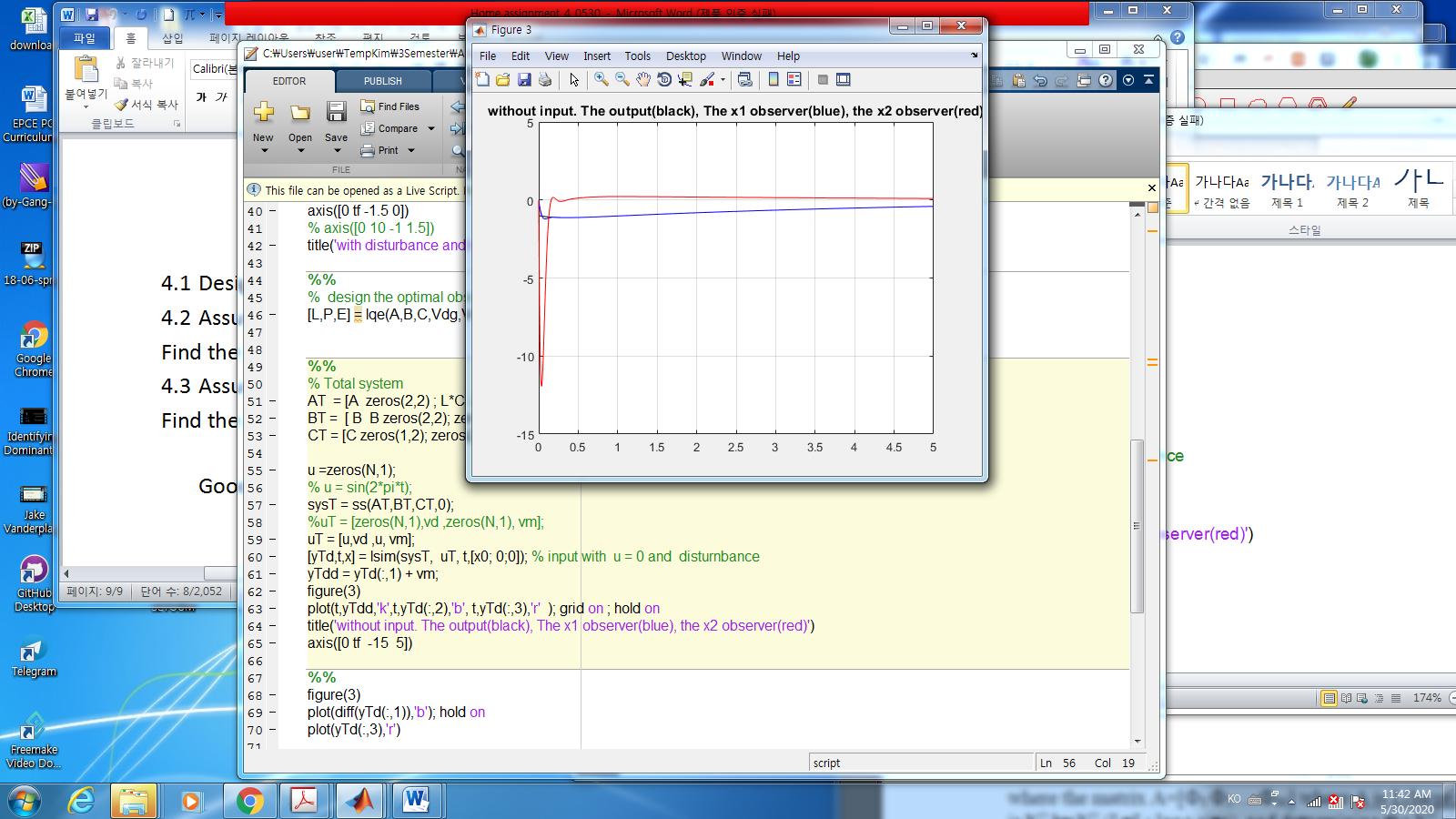
yTdd = yTd(:,1) + vm;

figure(3)

plot(t,yTdd,'k',t,yTd(:,2),'b', t,yTd(:,3),'r' ); grid on ; hold on

title('without input. The output(black), The x1 observer(blue), the x2 observer(red)')

axis([0 tf -15 5])



With the initial point of the observer, you may see

even if the initial point are different.

And you may check

So far the observer is good with no input.

* + 1. Input is not zero

% AT = [A zeros(2,2) ; L\*C A-L\*C];

% BT = [ B B zeros(2,2); zeros(2,2) B L];

% CT = [C zeros(1,2); zeros(2,2) eye(2,2)];

% sysT = ss(AT,BT,CT,0);

u = sin(2\*pi\*t);

uT = [ka\*u,vd ,ka\*u, vm];

[yTd,t,x] = lsim(sysT, uT, t,[x0; 0;0]); % input with u = 0 and disturnbance

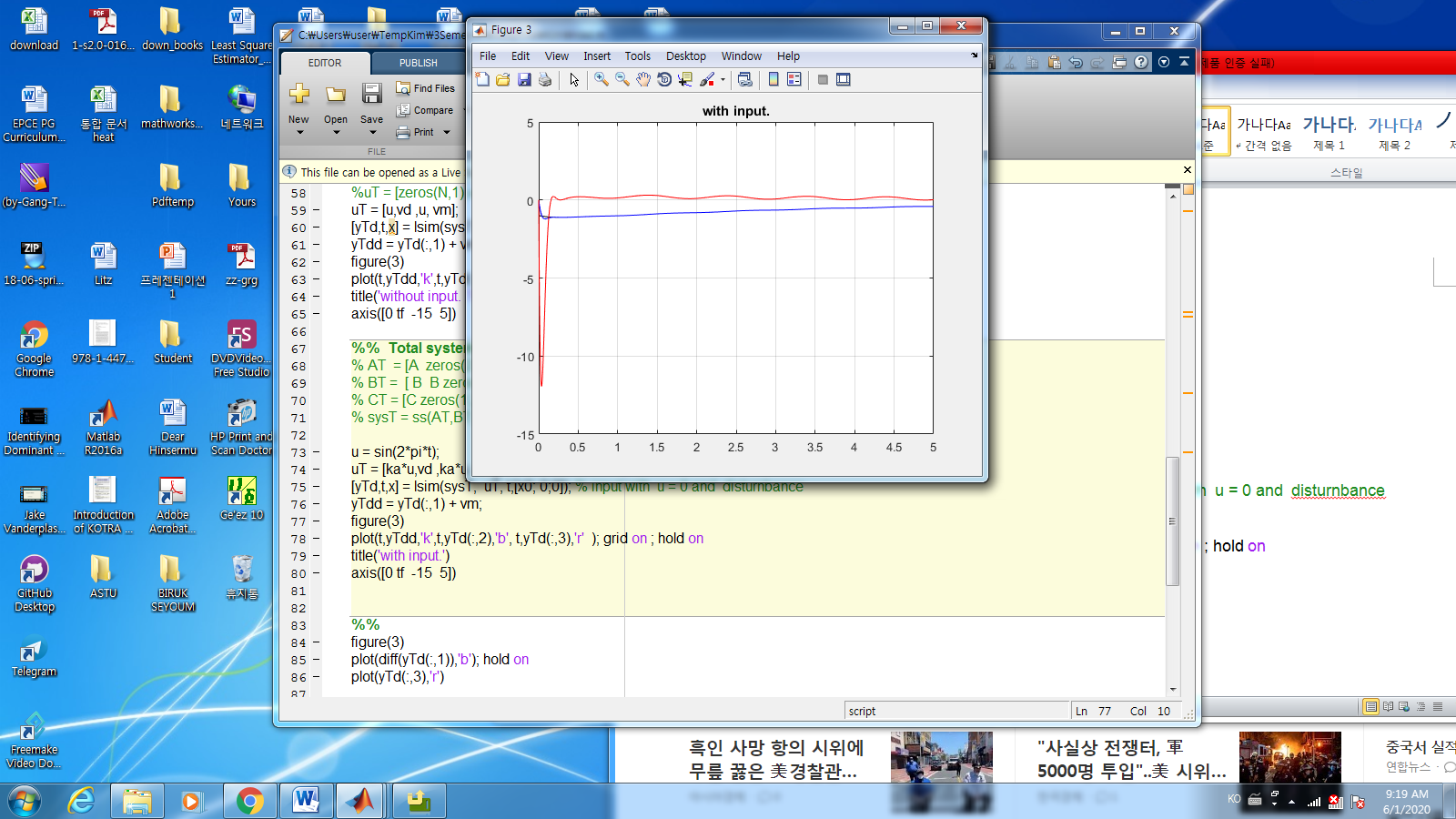
yTdd = yTd(:,1) + vm;

figure(3)

plot(t,yTdd,'k',t,yTd(:,2),'b', t,yTd(:,3),'r' ); grid on ; hold on

title('with input.')

axis([0 tf -15 5])



Even if the input is not zero, the observer looks good.

* + 1. The poles of the total system: Separation Principle.

Now I will consider the eigenvalues of the total system:

eig(AT)

eig(A)

eig(A-L\*C)

>> ans =

-0.2288 + 0.0000i

-4.3712 + 0.0000i

-22.4675 +22.2533i

-22.4675 -22.2533i

ans =

-0.2288

-4.3712

ans =

-22.4675 +22.2533i

-22.4675 -22.2533i

OK. I see the eigenvalues of total system are decomposed into two groups.

1. The plant eigenvalues
2. The observer eigenvalues.

That means for any optimal observer gains, it we use the observer dynamics as

the eigenvalues of the plant are independent. That is important as follows

* You may design for the controller independent of the observer dynamics as long as implemented as (1)
* After you design the optimal observer, then you may design your plant as a full state feedback to get a feedback gain

1. **Pole placements : any places in LHP(s-domain)**
2. **One of alternatives is to use Linear quadratic Regulator(LQR)**

* One of the advantages of the observer(Kalman observer) over the other controller

1. For “PID”. If you use “P” controller, you may not assign the poles as you want
2. For “PID”. If you use “PI” controller, you may assign the poles as you want, BUT the additional poles is introduced in the plant as

which is complicate to compare the performances for different . Even if the poles are same, the values of may be different. It is not unique.

1. I should remind you that the design an Kalman observer need a computer.
2. Separation Principle

This is the important concept introduced by Kalman. Let’s see more in detail.

In textbook or others in my materials, with controller

Hence the system matrix is

1. What is eigenvalues of the total system?

Hmmm… It is difficult… How to simplify to get this question? OK.

Define a new variable as

Then from (5.3) and (5.6)

What is the eigenvalues of ?

The block operation gives us that the eigenvalues of is decomposed to eigenvalues of and .

1. OK. Another method. Linear transformation.

Let’s define new variables as

Now we may have new variables dynamics as

🡪

OK. Now

Hence

which gives to

In linear algebra, the eigenvalue of is the same as .

So the total systems eigenvalues are the same as the eigenvalues of . Good.

1. The behavior of the observer.

Let’s see the total systems. In (5.6)

So the observer is faster convergent then the controller of !!

It is important to remember **the rate of convergent of the observer is faster than the rate of the plant.**

%%% --------- the last comments on the LQR/LQG

Up to now, the solution of the optimal controllers with the path constraints (the plant) is LQR/LQG.

So if an additional constraints imposed on the optimal controller, there is few(?) analytic solutions. Now we may have a powerful computer, even if there are no solutions, we may find numerical optimal solutions, which is machine learning, data handling, MPC and so on. It’s is your turn to develop another good controller. I hope so.

--------------------------------------------------------------------------%%%%%%%%%%%%