SLIDES 9a – Principle of Separation

Recall:

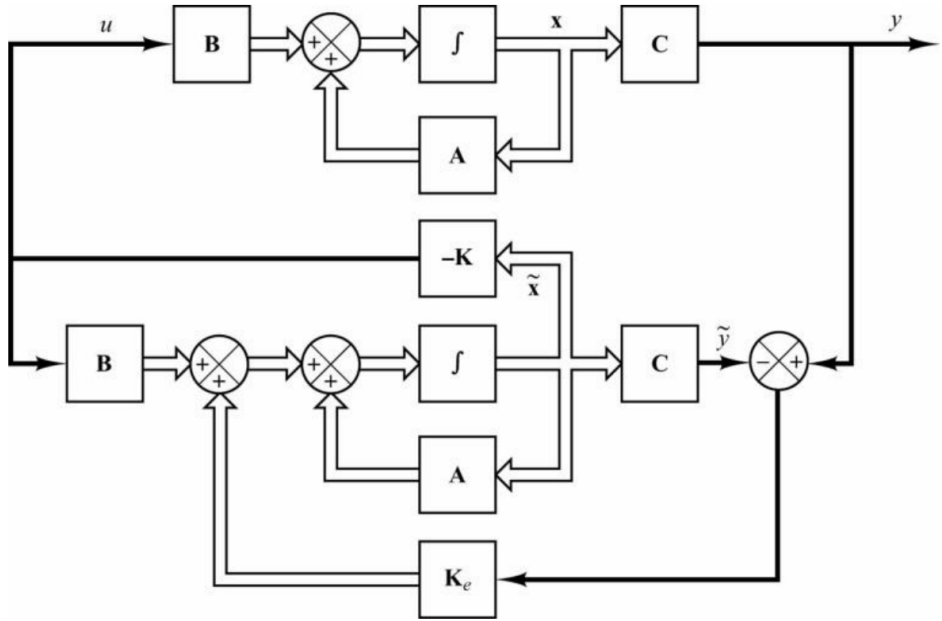
- full-state feedback is used for state-space control

$$u = -K\mathbf{x}$$

- poles are placed by choice of K via $A_c = A - BK$
- [A,B] needs to be a controllable pair
- An observer is a real-time system model with feedback
- $\dot{\tilde{\mathbf{x}}} = A\,\tilde{\mathbf{x}} + Bu + K_e(y - C\tilde{\mathbf{x}})$
- Poles are placed via $A_e = A - K_e C$
- $\boldsymbol{e} = \mathbf{x} - \tilde{\mathbf{x}}$ tend to zero because $\dot{\boldsymbol{e}} = A_e \boldsymbol{e}$

The next step is to connect the estimated states to the control input

$$u = -K\tilde{\mathbf{x}}$$


u
B
∫
x
C
y
A
–K
x̃
B
∫
C
ỹ
A
K_e

Could this be a bad idea?! What if the system becomes unstable?

According to the Principle of Separation the result is 'as stable as can be' … the poles of the total system (coupled controller and observer) are just the poles of the controller alongside the poles of the observer.

In this case we follow a simple design process

1. Place poles to design a controller (check simulated performance)
2. Place poles (faster!) to design an observer (check simulation)
3. Use the observer to provide feedback, $u = -K\tilde{\mathbf{x}}$ – couple the two systems
4. Test in simulation
5. Run in real-time

The Principle of Separation says this is OK!

In more detail ….

Separation principle

From Wikipedia, the free encyclopedia

In control theory, a **separation principle**, more formally known as a **principle of separation of estimation and control**, states that under some assumptions the problem of designing an optimal feedback controller for a stochastic system can be solved by designing an optimal observer for the state of the system, which feeds into an optimal deterministic controller for the system. Thus the problem can be broken into two separate parts, which facilitates the design.

The first instance of such a principle is in the setting of deterministic linear systems, namely that if a stable observer and a stable state feedback are designed for a linear time-invariant system (LTI system hereafter), then the combined observer and feedback is stable. The separation principle does not hold in general for nonlinear systems.

An "optimal stochastic observer" feeding into an "optimal deterministic controller" goes beyond what we have time for, but we can describe the general idea

- An optimal stochastic observer is called a Kalman filter. It's the same as our full-order state observer but the desired poles are found by balancing process noise against sensor noise to get the best balance between large K_e (believe the sensor) and small K_e (believe the model).
- An optimal deterministic controller is usually called a "Linear Quadratic Regulator" (LQR) which chooses desired poles by balancing controller performance (great performance needs large K) against control effort (low effort requires small K).

To analyze the coupled system (and check that it's stable) we first need a result about determinants …

Matrix Algebra Result

Suppose $[P, Q, R]$ are square matrices of equal dimension (e.g. 3 x 3).

The partitioned matrix

$$S = \begin{bmatrix} P & Q \\ 0 & R \end{bmatrix}$$

has determinant $\det S = \det P \times \det R$

It's not hard to prove, but it needs the formal definition of a determinant. It's probably easier to check via an example in Matlab.

It follows that the eigenvalues of S are just a compilation of the eigenvalues of P and R … suppose λ is an eigenvalue of P, then it's also an eigenvalue of S and vice-versa (consider $S - \lambda I$).

Coupled System

$$\dot{\mathbf{x}} = A\mathbf{x} + Bu$$

$$\dot{\mathbf{x}} = A\mathbf{x} - BK\tilde{\mathbf{x}}$$

$$\dot{\mathbf{x}} = A\mathbf{x} - BK\mathbf{x} + BK\mathbf{x} - BK\tilde{\mathbf{x}}$$

$$\dot{\mathbf{x}} = A_c\mathbf{x} + BK(\mathbf{x} - \tilde{\mathbf{x}})$$

$$\dot{\mathbf{x}} = A_c\mathbf{x} + BK\boldsymbol{e} \quad (1)$$

Also

$$\dot{\boldsymbol{e}} = A_e\boldsymbol{e} \quad (2)$$

Equations (1) and (2) form a combined state-space system with $\boldsymbol{X} = \begin{pmatrix}\mathbf{x}\\ \boldsymbol{e}\end{pmatrix}$ …

$$\begin{pmatrix}\dot{\mathbf{x}}\\ \dot{\boldsymbol{e}}\end{pmatrix} = \begin{pmatrix}A_c & BK\\ 0 & A_e\end{pmatrix}\begin{pmatrix}\mathbf{x}\\ \boldsymbol{e}\end{pmatrix}$$

From the 'matrix algebra result' the eigenvalues of the coupled systems are just the compiled list of the eigenvalues of A_c and A_e … so nothing will go wrong!

Provided:

- The plant is linear
- We have a reasonably accurate knowledge of [A, B, C, D]

The separation principle gives us confidence that using observer estimates is a good approach which can work even if the plant has some nonlinearities and even if we don't *exactly* know all the plant parameters.

Now let's look at a Simulink example …

Example

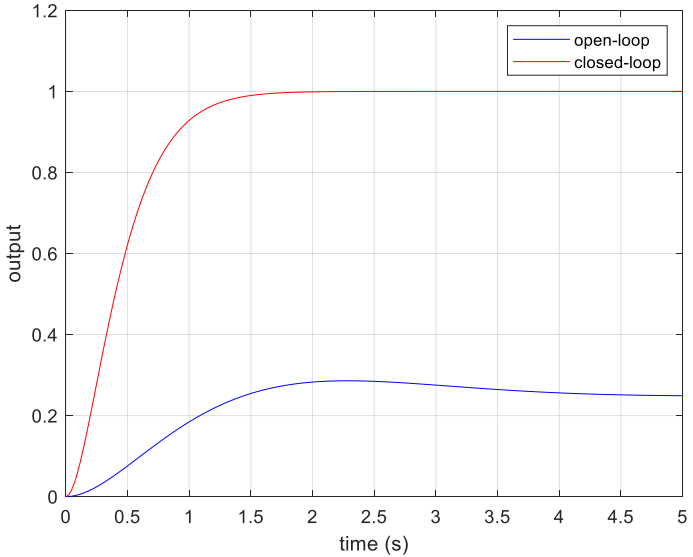
Use Matlab and Simulink to design a state-space controller, including state observer, for the plant with the following transfer function

$$G(s) = \frac{s+1}{(s+2)(s^2+2s+2)}$$

This follows on from an earlier example. We can extend the Simulink model …. but first design the state-space controller.

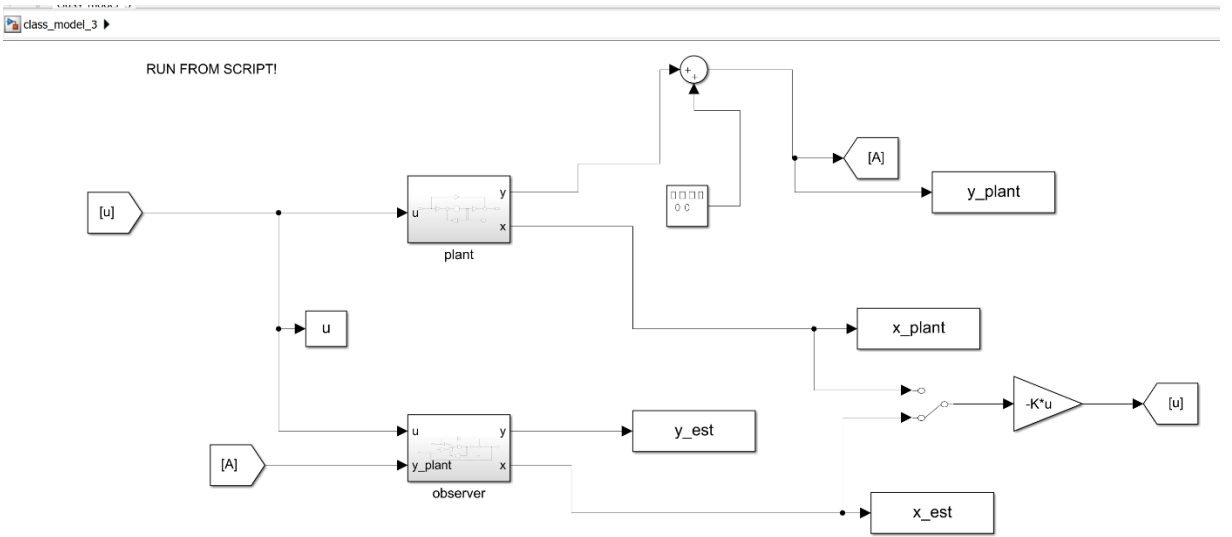
```
%CCF
A=[0 1 0;0 0 1;-4 -6 -4];
B=[0 0 1]';
C=[1 1 0];
D=0;

%controller design
e0=eig(A); %stable but design a controller to give a faster response
p=[-1,-4+1j,-4-1j]; %cancel the zero!
K=place(A,B,p);
```

```
%check step response in Matlab (optional)
if 1
    sys0=ss(A,B,C,D);
    Ac=A-B*K;
    ssg=-C*inv(Ac)*B+D;
    sys1=ss(Ac,B/ssg,C,D);
    [y0,t0]=step(sys0,5);
    [y1,t1]=step(sys1,5);
    figure,plot(t0,y0,'b')
    hold on
    plot(t1,y1,'r')
    xlabel('time (s)'),ylabel('output')
    legend('open-loop','closed-loop')
    grid
    return
end
```

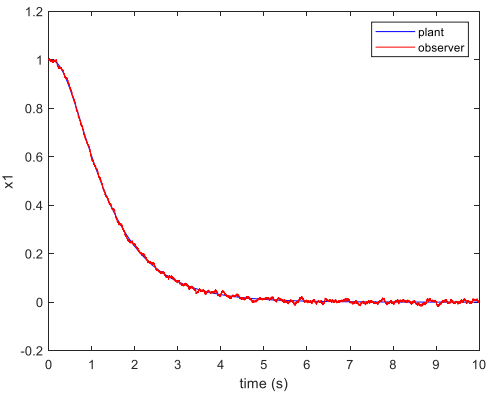


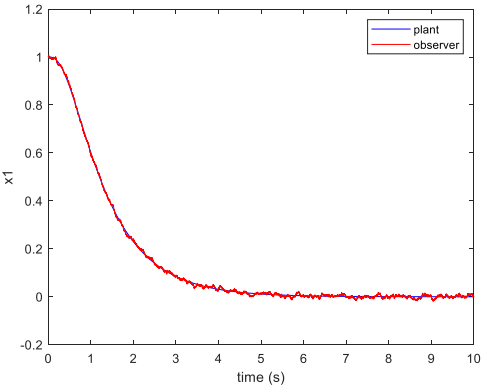
This assumes measurement of the states and the simulation uses the ideal conditions of Matlab. The next step is to (re-) design the state observer and use this for feedback … fingers crossed it does not go unstable!

Here is the combine Simulink model. Note the use of a manual switch, so we can switch between true states and estimated states in the feedback



This is set up as a regulator problem, so we just see the effect of starting at some nonzero initial condition, e.g., $x(0) = [1 \;\; 0 \;\; 0]$. The left plot uses the true states for feedback, the right plot uses estimates





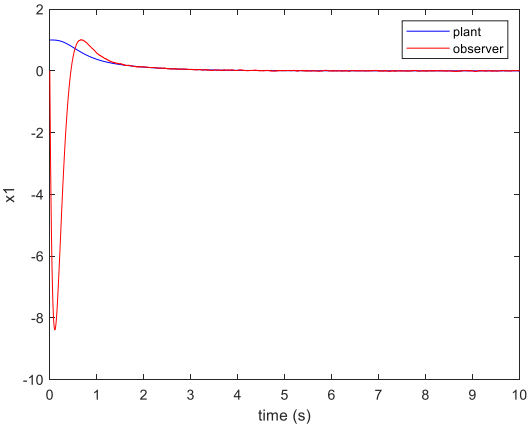
We have included some sensor noise in the model, hence the small wiggle in the estimated states. There is hardly any difference between the responses.

We have slightly 'cheated' here, starting the observer from the same states. This is actually the normal situation … we let the observer settle before applying active control. But we can see the effect of starting the observer from [0 0 0].

Don't panic!

The red curve is just in software, the blue response is the real one and it's perfectly well settled!

The Matlab script is class_script_3.m and the model is class_model_3



Exercise Build your own version of this, starting from the posted versions 1 and 2. Check you agree with the above results and check the responses of x2 and x3.

Summary

- We have now covered the most important results in the module, designing controller and observers and connecting the two together
- Simulink is closely connected to Matlab – it's make it easier to run the model from a script (which also sets up the various matrices)
- Simulink gives us flexibility to simulate different versions of the model, including any nonlinear versions.