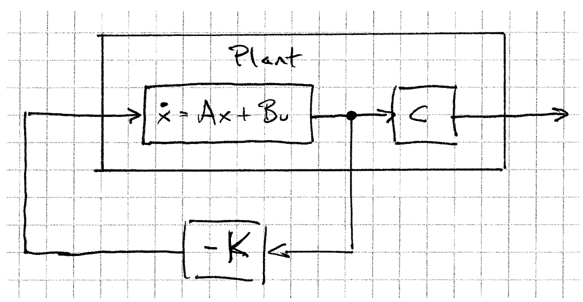
16-642 Fall 2017: Reference Notes for Linear Observers

George Kantor

6-8 November 2017

1 Observability and Observers

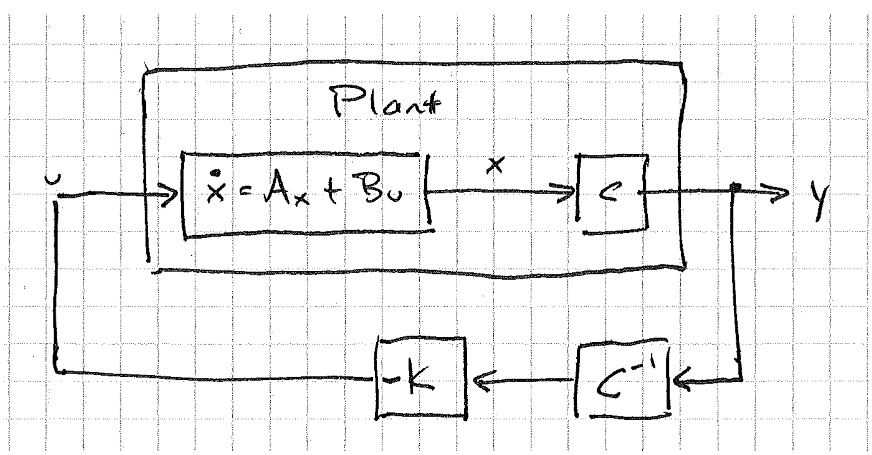
So we have some really powerful tools for state feedback (eigenvalue placement and LQR). The problem is that we may not know the state. In general, we only know the output $y = Cx$:



One possible solution: if C is invertible, then we can get the state as

$$x = C^{-1}y.$$

The resulting control system then looks like this:



Unfortunately this hardly ever happens (C is rarely even square!).

So if we want to do state feedback, we have to build something that reconstructs the state using

- what we know about the system dynamics (A, B, C)
- the output y measured over some period of time
- the input u measured over the same period of time

The thing we want to build is called a *state observer* that estimates the state of the system by looking things we can see (i.e., the inputs and outputs.)

1.1 Observability

It may not be possible to build an observer for a given system (just like it may not be possible to build a stabilizing controller for a given system). So we introduce the concept of "observability":

A system

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

is said to be observable if it is possible to reconstruct the initial state $x(0)$ by observing the input $u(t)$ and the output $y(t)$ over some finite time interval $t \in [0, t_f]$.

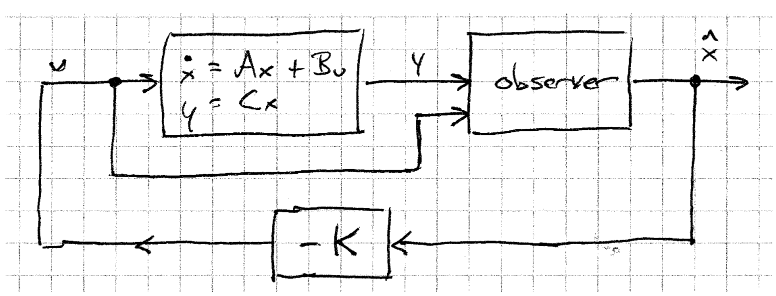
Obserbability Test:
A system is observable if and only if

$$\mathrm{rank}(W_o) = n,$$

where

$$W_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}.$$

If the system is observable, then we should be able to build an observer that takes in the system inputs and ouputs and uses them to create and estimate of the state. Then this estimate can be used for state feedback:



Now let's try to build an observer.

First Try: Just build a copy of the equations of motion of the actual system and integrate them to get the estimate, i.e.,

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

If $\hat{x}(0) = x(0)$, then $\hat{x}(t) = x(t)$ for all time, so the observer works. Unfortunatley, there are some big problems with this:

1. we don't know $x(0)$, so we can't possibly set $\hat{x}(0) = x(0)$.
2. errors due to incorrect $x(0)$ and noise on $u(t)$ get integrated and grow without bound as time progresses.

Proposed Fix: Add a correcting term that adjusts the observer equation based on the difference between the measured output and what the output would be if $\hat{x}$ was correct:

$$\dot{\hat{x}} = A\hat{x} + Bu + K_o(y - C\hat{x}).$$

Two questions:

1. will this work?
2. how do we choose K_o?

We can answer both of these questions by looking at the **error dynamics**. (strong hint: the following derivation is very important, and everyone will be expected to know and thoroughly understand it.) Define $e(t)$ to be the error in the state estimate $\hat{x}(t)$, i.e.,

$$e(t) = x(t) - \hat{x}(t)$$

Now look at how e changes:

$$\begin{aligned}
\dot{e} &= \dot{x} - \dot{\hat{x}} \\
&= Ax + Bu - (A\hat{x} + Bu + K_oC(x - \hat{x})) \\
&= A(x - \hat{x}) - K_oC(x - \hat{x}) \\
&= (A - K_oC)e.
\end{aligned}$$

So the error dynamics are the same as the dynamics for an unforced system. Specifically, if the eigenvalues of the matrix $A - K_oC$ all have negative real part, then the error is guaranteed to go to zero as $t \to \infty$. So the problem of building an observer boils down to the problem of placing the eigenvalues of $A - K_oC$. So we're faced with a question that is a lot like the question we had for state feedback control: given a desired set of eigenvalues, can we find a suitable K_o? If the system is observable then the answer is "yes". Of course, there is a theorem that looks a lot like the eigenvalue placement theorem for controllers:

eigenvalue Placement Theorem for Observers: Let $\Lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ be any allowable set of eigenvalues. The pair (A, C) is observable if and only if there exists a K_o such that

$$\text{eig}(A - K_oC) = \Lambda$$

Next time, I'll show you how we can leverage what we already know about the controller problem to solve the observer problem. In fact, they are really just different versions of the same problem. This concept is sometimes called *duality*.

2 Duality

Lets start by going over one similarity you've probably already noticed, namely the similarity between the controllability test.

A pair (A, B) is controllable if the $n \times nm$ matrix

$$Q_c = \begin{bmatrix} B & AB & A^2B & \cdots & A^{n-1}B \end{bmatrix}$$

has rank n.

A pair (A, C) is observable if the $pn \times n$ matrix

$$Q_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

has rank n.

We can use a simple matrix fact to combine these two to come up with a new fact.

a simple matrix fact:

$$\mathrm{rank}(M) = \mathrm{rank}\left(M^T\right)$$

for any matrix M.

Here's how we can use this fact:

$$\mathrm{rank}(Q_o) = \mathrm{rank}\left(Q_o^T\right)$$

$$= \mathrm{rank}\left(\begin{bmatrix} C^T & A^T C^T & \left(A^T\right)^2 C^T & \cdots & \left(A^T\right)^{n-1} C^T \end{bmatrix}\right)$$

If we look at this last espression for a while, we recognize that it is exactly the same as the expression in the controllability test with A replaced by A^T and B replaced by C^T. This leads to the new fact:

fact: (duality between controllabilty and observability)
The pair (A, C) is observable if and only if the pair (A^T, C^T) is controllable.

Now the eigenvalue placement theorem tells us that if (A, B) is controllable, then we can find a K_c to abritrarily place the eigenvalues of $A - BK_c$.

Using the duality fact, we can then say that if the pair (A, C) is observable, then the pair (A^T, C^T) is controllable, and we can find a K_{tmp} so that the eigenvalues of the matrix $A^T - C^T K_{\mathrm{tmp}}$ match any deisred allowable set of eigenvalues Λ_o. Also, we could find K_{tmp} using MATLAB:

$$K_{\mathrm{tmp}} = \texttt{place}(A^T, C^T, \Lambda_o).$$

We are so very close to solving eigenvalue placement for observers. We need one more simple matrix fact:

another simple matrix fact:

$$\mathrm{eig}(M) = \mathrm{eig}\left(M^T\right)$$

for any square matrix M.

Recall the problem: we wish to find a K_o to place the eigenvalues of $A - K_o C$. Using this last fact, we see that

$$\mathrm{eig}(A - K_o C) = \mathrm{eig}\left(A^T - C^T K_o^T\right).$$

We already know how to find K_o^T to place the eigenvalues of the matrix on the right hand side (i.e., just set $K_o^T = K_{\mathrm{tmp}}$.) So we can finally say what we need to say about assigning observer eigenvalues:

Theorem: eigenvalue placement for observers The pair (A, C) is observable if and only if for any allowable set of eigenvalues Λ_o there exists a matrix K_o so that

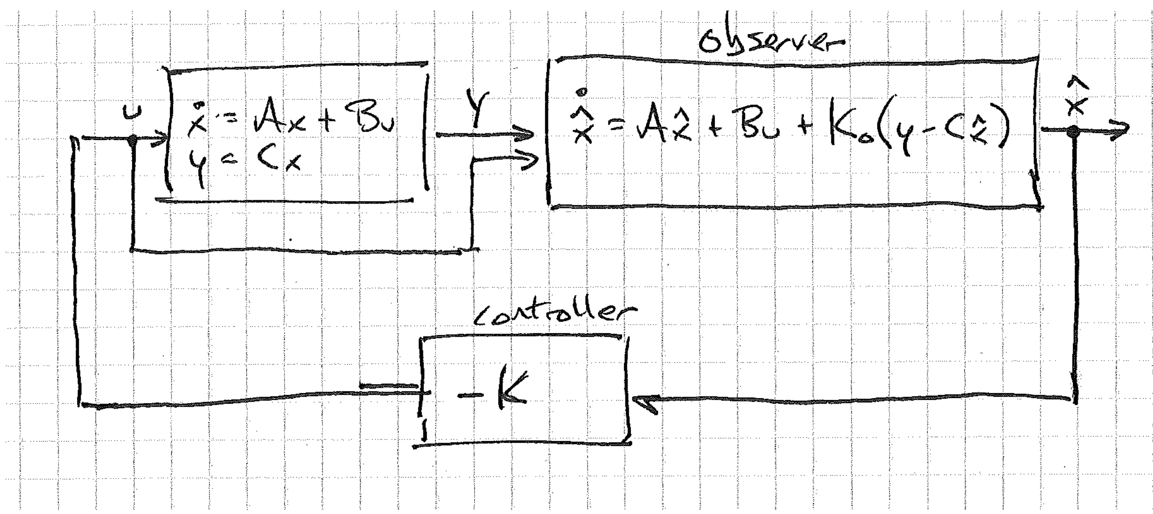
$$\mathrm{eig}(A - K_o C) = \Lambda_o.$$

Further, we can find K_o using the MATLAB place command as follows:

$$K_o = \left(\texttt{place}\left(A^T, C^T, \Lambda_o\right)\right)^T.$$

3 Putting It State Feedback and Observer Together

Here's a block diagram of the overall observer/state feedback control system:



Now we can clearly state the process of controller design based on state feeback. Given

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

take the following steps:

1. find a state feedback matrix K_c such that the closed loop system

$$\dot{x} = (A - BK_c)x$$

   has desired dynamics. (do this using either eigenvalue placement or LQR.)

2. find an observer matrix K_o such that the error dynamics

$$\dot{e} = (A - K_oB)e$$

   has the desired dynamics. This is usually done using eigenvalue placement, and a good rule of thumb is to place all of the observer eigenvalues to the left of all of the closed loop controller eigenvalues so that the estimate $\hat{x}$ has a chance of keeping up with x.

There are no guarantees that this will work, but if you make the error dynamics of the observer much faster than the unforced dynamics of the closed loop system then things usually work out OK.

3.1 Working Out the Math

Let's take a closer look at the math behind this observer/controller scheme:

$$\dot{x} = Ax + Bu$$

$$\dot{\hat{x}} = A\hat{x} + Bu + K_o(Cx - C\hat{x})$$

$$u = -K_c\hat{x}$$

Subbing in for u gives:

$$\dot{x} = Ax - BK_c\hat{x}$$

$$\dot{\hat{x}} = A\hat{x} - BK_c\hat{x} + K_o(Cx - C\hat{x})$$

$$= (A - BK_c - K_oC)\hat{x} + K_oCx$$

We can now put these together into one big equation:

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \underbrace{\begin{bmatrix} A & -BK_c \\ K_oC & A - BK_c - K_oC \end{bmatrix}}_{\triangleq A_z} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$$

So if we define

$$z = \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$$

The big equation becomes

$$\dot{z} = A_z z$$

and we can analyze the stability by looking at the eigenvalues of A_z. It turns out that

$$\text{eig}A_z = \{\text{eig}(A - K_oC), \text{eig}(A - BK_c)\}$$