CDS110a - Wednesday 27 October 2004

Note: there was indeed a unit problem with the reachability matrix in last lecture; see FAQ.

1. Quick review of reachability and pole-placement
2. State feedback vs. output feedback; note that state feedback is often not the natural design paradigm
   a. Note that stabilization often does not require full knowledge of state, but state steering and output tracking generally will
   b. Example - stabilize the origin for a simple harmonic oscillator - clearly sufficient just to know velocity for damping
   c. Example - steering in the predator-prey system, *e.g.*, would seem to require full state knowledge
3. Can state be determined from output signal?
   a. Multiple output with C a square matrix - invertible?
   b. Multiple or single output with C rectangular - sometimes still okay if K has low "support" (just a remark)
   c. Look at an example with C not invertible, but where intuitively you should be okay
4. State estimation by derivatives, as in readings
   a. Observability matrix and observability
   b. Note problems with this approach in finite precision
5. State observer with innovations
   a. Derive
   b. Show that this reduces to a pole-placement problem
   c. Same observability criterion as above...
   d. Mention duality with controllability scenario
6. Output feedback; theorem on pole-placement with an observer

A quick review of reachability

Last time we saw that a linear system

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

is *reachable* if the reachability matrix

$$[B \quad AB \quad A^2B \quad \cdots \quad A^{n-1}B]$$

has full rank. This means that for any initial state $x(0) = x_0$, desired final state x_f and 'target time' T it is possible to find a control input $u(t),\ t \in [0,T]$ that steers the system to reach $x(T) = x_f$. If a system is reachable, it is furthermore possible to solve the *pole placement* problem, in which we want to design a state feedback law

$$u = Kx$$

such that we can pick any eigenvalues we want for the controlled dynamics

$$\begin{aligned}\dot{x} &= Ax + Bu \\ &= (A + BK)\,x.\end{aligned}$$

Here A and B are given, and we must find a K to achieve the desired eigenvalues for $(A + BK)$.

Before moving on, let's look at a simple example (from Åström and Murray, Ex. 5.3) of a system that is *not* reachable:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = -\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u.$$

Here we can easily compute

$$W_r = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix},$$

which clearly has determinant zero. This can be understood by noting the complete 'symmetry' of the way that u modifies the evolution of x_1 and x_2. For example, if $x_1(0) = x_2(0)$ there is no way to use u to achieve $x_1(T) \neq x_2(T)$ at any later time.

State feedback versus output feedback

Note that in our discussion of stabilization and pole-placement so far, we have assumed that it makes sense to design a control law of the form

$$u = Kx.$$

This is called a 'state feedback' law since in order to determine the control input $u(t)$ at time t, we generally need to have full knowledge of the state $x(t)$. In practice this is often not possible, and thus we usually specify the available output signals when defining a control design problem:

$$\begin{aligned}\dot{x} &= Ax + Bu, \\ y &= Cx.\end{aligned}$$

Here the output signal $y(t)$, which can in principle be a vector of any dimension, represents the information about the evolving system state that is made available to the controller via sensors. An 'output feedback' law must take the form

$$u(t) = f\,[y(\tau \leq t)],$$

where, in general, we can allow $u(t)$ to depend on the entire history of $y(\tau)$ with $\tau \leq t$ (more on this below and later in the course). Output feedback is a natural setting for practical applications. For example, if we are talking about cruise control for an automobile, x may represent a complex set of variables having to do with the internal state of the engine, wheels and chassis while y is only a readout from the speedometer. Hopefully it will seem natural that it is usually prohibitively difficult to install a sensor to monitor every coordinate of the system's state space, and also that it will often be unnecessary to do so (cruise control electronics can function quite well with just the car's speed).

One simple example of a system in which full state knowledge is clearly not necessary is stabilization of a simple harmonic oscillator. If the natural dynamics of the plant is

$$m\ddot{x} = -kx,$$

and our actuation mechanism is to apply forces directly on the mass, then the control system looks like

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{-k}{m} & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u,$$

(where x_1 is now the position and x_2 the velocity). We can clearly stabilize the equilibrium point at the origin by the feedback law

$$u = -bx_2 = \begin{bmatrix} 0 & -b \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

which makes the overall equation of motion

$$\ddot{x}_1 = -\frac{k}{m}x_1 - b\dot{x}_1,$$

which we recognize as a damped harmonic oscillator. Thus it is clear that the controller only needs to know the velocity of the oscillator in order to implement a successful feedback strategy. So even if we go to a SISO output feedback formulation of this problem,

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

$$y = Cx,$$

we are obviously fine for any C of the form $(\alpha \neq 0)$

$$C = \begin{bmatrix} 0 & \alpha \end{bmatrix},$$

since $x_2 = y/\alpha$ and we can implement an output-feedback law of the form

$$u = -bx_2 = -\frac{b}{\alpha}y.$$

In contrast to this, imagine a steering problem for the predator-prey system that we talked about in the last lecture. Suppose for instance we want to design a controller that will take the fox and rabbit population from an arbitrary initial state at $t = 0$ to some specific final state such as $(36, 51)$ at time $t = T$. Even if we restrict our attention to the immediate vicinity of the natural equilibrium point, and assume that a linearized model is sufficient for the design, it seems quite unlikely that we could succeed without requiring knowledge of both the fox and rabbit populations at time $t = 0$.

Clearly, if C is a square matrix and y has the same dimension as x, everything will be easy if C is invertible. As a generalization of what we did for the simple harmonic oscillator above, we could just design a state feedback controller K, set

$$\hat{x} = C^{-1}y,$$

and apply feedback

$$u = K\hat{x} = KC^{-1}y.$$

However this is a special case and not the sort of convenience we want to count on!

State estimation

At this point it might seem like we would need completely new theorems about reachability and pole-placement for output-feedback laws, when $u(t)$ is only allowed to depend on $y(\tau < t)$. However, it turns out that we can build naturally on our previous results by appealing to a *separation method*. The basic idea is that we will try to construct a procedure for processing the data $y(\tau < t)$ to obtain an estimate $\hat{x}(t)$ of the true system state $x(t)$, and then apply a feedback law $u = K\hat{x}$ based on this estimate. This can be possible even when C is not invertible (not even square). The controller thus assumes the structure of a dynamical system itself, with $y(t)$ as its input, $u(t)$ as its output and $\hat{x}(t)$ as its internal state. There are various ways of designing 'state estimators' to extract $\hat{x}(t)$ from $y(\tau < t)$, of which we will discuss two, and there is also a convenient procedure for determining whether or not y contains enough information to make full state reconstruction possible in principle. The latter test looks a lot like the test for reachability, for not accidental reasons.

Let's start by thinking about the simple harmonic oscillator again. We noted that in order to stabilize the equilibrium point at the origin, it would be most convenient to have an output signal that told us directly about its velocity x_2. However, you may have already realized that in a scenario with $(\alpha \neq 0)$

$$C = \begin{bmatrix} \alpha & 0 \end{bmatrix},$$

$$y = \alpha x_1,$$

it should be simple to obtain a good estimate of x_2 via

$$\hat{x}_2 = \frac{d}{dt}\alpha^{-1}y.$$

This is certainly a valid procedure for estimating x_2, although in practice one should be wary of taking derivatives of measured data since that tends to accentuate high-frequency noise.

In a similar spirit, we note that for any dynamical system

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

$$y = Cx,$$

if we hold u at zero we can make use of the general relations

$$\begin{aligned}
\dot{y} &= C\dot{x} = CAx,\\
\ddot{y} &= C\ddot{x} = C\frac{d}{dt}\dot{x} = C\frac{d}{dt}Ax = CA\dot{x} = CA^2x,\\
&\vdots\\
\frac{d^n}{dt^n}y &= CA^n x.
\end{aligned}$$

If we look at how this applies to our modified simple harmonic oscillator example with

$$C = \begin{bmatrix} \alpha & 0 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{bmatrix},$$

$$y = C \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \alpha x_1,$$

we have

$$CA = \begin{bmatrix} \alpha & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & 0 \end{bmatrix} = \begin{bmatrix} 0 & \alpha \end{bmatrix},$$

$$\dot{y} = CA \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \alpha x_2,$$

and we start to get a sense for how the natural dynamics A can move information about state space variables into the 'support' of C. Hopefully it should thus seem reasonable that in order for a system to be *observable*, we require that the observability matrix

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

have full rank. Informally, if a system is observable then we are guaranteed that we can design a procedure (such as the derivatives scheme above) to extract a faithful estimate $\hat{x}$ from y. However, it will generally be necessary to monitor y for some time (and with good accuracy) before the estimation error

$$\tilde{x}(t) \equiv x(t) - \hat{x}(t)$$

can be made small. In the derivatives scheme, for instance, we can't estimate high derivatives of $y(t)$ until we see enough of it to get an accurate determination of its slope, curvature, etc.

State observer with innovations

A more common (and more robust) method for estimating x from y is to construct a state observer that applies corrections to an initial guess $\hat{x}$ until $C\hat{x}$ becomes an accurate predictor of y.

Suppose that at some arbitrary point in time t we have an estimate $\hat{x}(t)$. How should we update this estimate to generate estimates of the state $x(t')$ with $t' > t$? Most simply, we could integrate

$$\frac{d}{dt}\hat{x} = A\hat{x} + Bu,$$

assuming we know A and B for the plant. It is generally assumed that we know u since this signal is under our control! Then we notice that the estimation error $\tilde{x}$ evolves as

$$\begin{aligned}\frac{d}{dt}\tilde{x} &= \frac{d}{dt}(x - \hat{x}) \\ &= (Ax + Bu) - (A\hat{x} + Bu) \\ &= A(x - \hat{x}) \\ &= A\tilde{x}.\end{aligned}$$

Hence, this strategy has the nice feature that if A is stable,

$$\lim_{t \to \infty} \tilde{x} = 0,$$

meaning that our estimate will eventually converge to the true system state. Note that this works even if B and u are non-zero.

What if we are not so lucky as to have sufficiently stable natural dynamics A? As mentioned above, a good strategy is to try to apply corrections to $\hat{x}$ at every time step, in proportion to the so-called innovation,

$$w \equiv y - C\hat{x}.$$

Here $y - C\hat{x}$ is the error we make in predicting $y(t)$ on the basis of $\hat{x}(t)$. Clearly when $\tilde{x}$ is small, so is $w.$ A 'Luenberger state observer' can thus be constructed as

$$\frac{d}{dt}\hat{x} = A\hat{x} + Bu + L(y - C\hat{x}),$$

where L is a 'gain' matrix that is left to our design. This observer equation results in

$$\begin{aligned}\frac{d}{dt}\tilde{x} &= \dot{x} - \frac{d}{dt}\hat{x} \\ &= (Ax - Bu) - (A\hat{x} + Bu + L(y - C\hat{x})) \\ &= A(x - \hat{x}) - L(y - C\hat{x}) \\ &= A(x - \hat{x}) - LC(x - \hat{x}) \\ &= (A - LC)\tilde{x}.\end{aligned}$$

Hence we see that our design task should be to choose L, given A and $C,$ such that $(A - LC)$ has nice stable eigenvalues.

This should remind you immediately of the pole-placement problem in state feedback, in which we wanted to choose $K,$ given A and $B,$ such that $(A + BK)$ had desired eigenvalues. Indeed, one can map between the two problems by noting that the transpose of a matrix M^T has the same eigenvalues as $M.$ Thus we can view our observer design problem as being the choice of L^T such that

$$(A - LC)^T = A^T - C^T L^T$$

has nice stable eigenvalues, and this now has precisely the same structure as before. Indeed, there is a complete 'duality' between state feedback and observer design, with correspondences

$$A \leftrightarrow A^T, \quad B \leftrightarrow -C^T, \quad K \leftrightarrow L^T, \quad W_r \leftrightarrow W_o^T.$$

(Note that if we use a state feedback law $u = -Kx$ in the original pole-placement problem, rather than $u = Kx$ as we did in L5.1, we recover $B \leftrightarrow C^T$.) Hence it should be clear, for example, how Matlab's **`place`** function can be used for observer design. And as long as the observability matrix has full rank, we are guarenteed to be able to find an L such that $(A - LC)$ has arbitrary desired eigenvalues.

Note: Matlab's **`place`** function assumes that you are using $u = -Kx$. Thus if you call

```
place(A,B,eigs)
```

it will return a matrix K such that $(A - BK)$ has the requested eigenvalues.

Pole-placement with output feedback

As discussed in section 5.6 of Åström and Murray, the following theorem holds (here we simplify to the $r = 0$ case):

For a system

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

$$y = Cx,$$

the controller described by

$$u = -K\hat{x},$$

$$\frac{d}{dt}\hat{x} = A\hat{x} + Bu + L(y - C\hat{x})$$

gives a closed-loop system with the characteristic polynomial

$$\det(sI - A + BK)\det(sI - A + LC).$$

This polynomial can be assigned arbitrary roots if the system is observable and reachable.

The overall setup is summarized in the following cartoon: