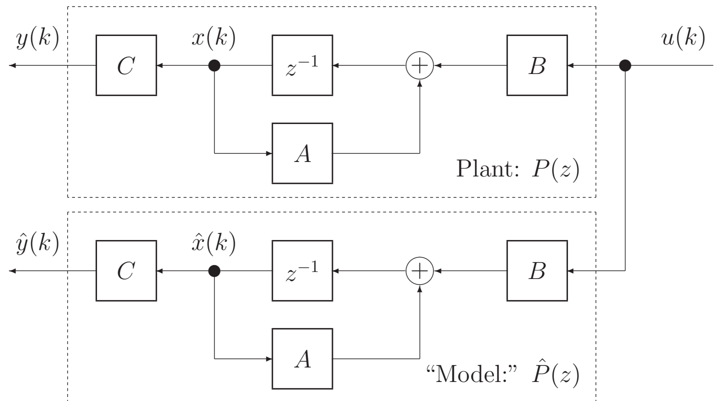
State Estimation

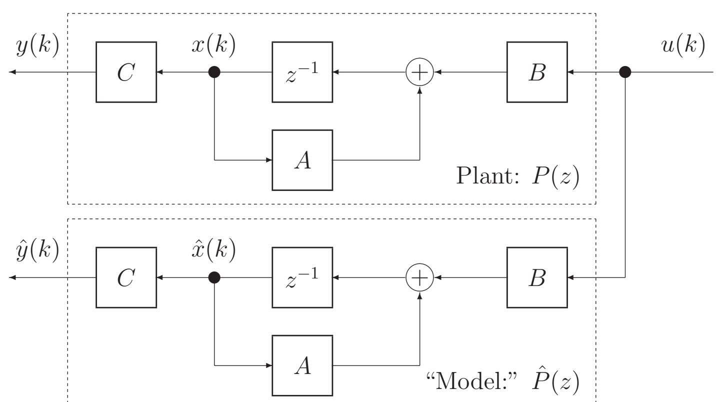
State feedback design assumes that we can measure the complete state.

What do we do if we cannot? **Estimate it**.

Approach: create a "model" of the system and use its state instead of the measured state.



Propagating the model:



To calculate an estimated state, $\hat{x}(k)$, we must choose an initial estimated state, $\hat{x}(0)$, and run it through out model.

Note that our controller will generate $u(k)$ so we know what this is for times, $k = 0, \ldots, k-1$.

Error properties:

Define the state estimation error:

$$\tilde{x}(k) := x(k) - \hat{x}(k).$$

Applying the state equation for both the model and the plant gives,

$$\begin{aligned}\tilde{x}(k+1) &= x(k+1) - \hat{x}(k+1) \\ &= A\,x(k) - A\,\hat{x}(k) \\ &= A\,(x(k) - \hat{x}(k)) \\ &= A\,\tilde{x}(k).\end{aligned}$$

So the dynamics of the error, $\tilde{x}(k)$, are the same as the open-loop dynamics of the plant.

If the plant is open-loop unstable, the state estimation error, $\tilde{x}(k)$, will blow up.

Error properties:

Look at the solution to the state equations:

$$\begin{aligned}x(k) &= A^k\,x(0) + \sum_{j=0}^{k} A^{(k-j)}B\,u(j) \\ \hat{x}(k) &= A^k\,\hat{x}(0) + \sum_{j=0}^{k} A^{(k-j)}B\,u(j)\end{aligned}$$

Subtracting these gives,

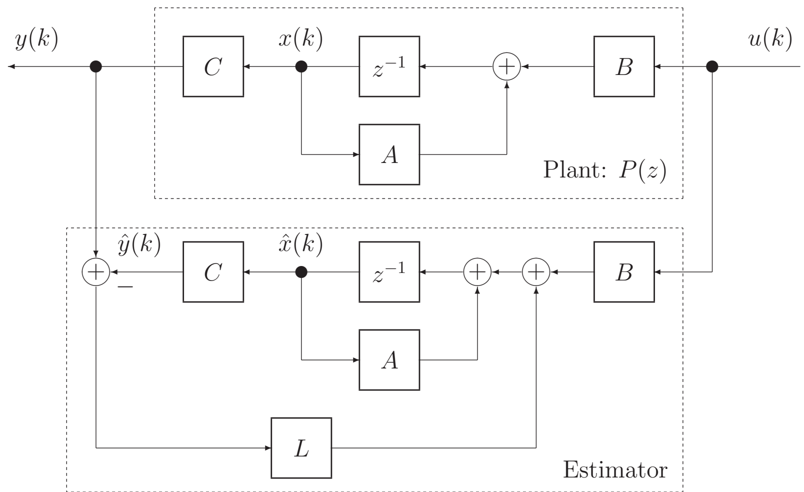
$\tilde{x}(k) = A^k\,\tilde{x}(0)$ $\longleftarrow$ the estimation error depends only on the initial error.

Again, it's easy to see that if A is stable the transient caused by the initial estimation error will decay to zero.

Can we do better?

Make use of the measurement, $y(k)$.

Estimator: use the information in the error between $y(k)$ and $\hat{y}(k)$.



Use some output error feedback, $L(y(k) - \hat{y}(k))$, to change the next state estimate, $\hat{x}(k+1)$.

Error properties

The estimated state update equation is now,

$$\begin{aligned} \hat{x}(k+1) &= A\,\hat{x}(k) \;+\; B\,u(k) \;+\; L(y(k) - \hat{y}(k)) \\ &= A\,\hat{x}(k) \;+\; B\,u(k) \;+\; LC(x(k) - \hat{x}(k)) \end{aligned}$$

Now subtract this from the true state update equation to get the error equation,

$$\begin{aligned} \tilde{x}(k+1) &= x(k+1) \;-\; \hat{x}(k+1) \\ &= A\,x(k) + B\,u(k) \quad - \quad [\; A\,\hat{x}(k) \;+\; B\,u(k) \;+\; LC(x(k) - \hat{x}(k)) \;] \\ &= A\,(x(k) - \hat{x}(k)) - LC(x(k) - \hat{x}(k)) \\ &= A\,\tilde{x}(k) - LC\,\tilde{x}(k)) \\ &= (A - LC)\,\tilde{x}(k). \end{aligned}$$

The estimator gain matrix, L, changes the error dynamics from A to $A - LC$.

Note the similarity with state-feedback which modifies the plant state dynamics via $A - BK$.

Designing an estimator

This comes down to choosing an L matrix.

Notice that the dynamics of the error are given by $A - LC$, so we can view this as a pole placement problem.

Ackermann's formula still works.

Note that

$$\text{eig}(A - LC) \;=\; \text{eig}(A - LC)^T \;=\; \text{eig}(A^T - C^T L^T),$$

and this is exactly the same as the state feedback pole placement problem: $A - BK$.

Ackermann's formula for L

Select pole positions for the error: $\eta_1, \eta_2, \cdots, \eta_n$.

Specify these as the roots of a polynomial,

$$\gamma_o(z) \;=\; (z - \eta_1)(z - \eta_2)\cdots(z - \eta_n).$$

We will again use this polynomial with the A matrix as the variable,

$$\gamma_o(A) \;=\; (A - \eta_1 I)(A - \eta_2 I)\cdots(A - \eta_n I).$$

Designing L

Define the *"observability matrix"*,

$$\mathcal{O} \;:=\; \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \qquad \text{and now} \qquad L \;=\; \gamma_o(A)\, \mathcal{O}^{-1} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

This can have an analogous problem to the controllability matrix, $\mathcal{O}$ may not be invertible.

Observability

The system is *"observable"* if and only if $\mathcal{O}$ is full rank.

Observability means that the states can be estimated from measurements of the output.

Example: Consider $f = m\dfrac{d^2x}{dt^2}$ (a double integrator).

Good choices of state are x (position), and v (velocity).

Is the system observable by measuring x?

Is the system observable by measuring v?

State feedback with estimated states

We can now put the two pieces together:

State feedback: $u(k) \;=\; -K\,x(k)$ $\longleftarrow$ this is based on the true state.

and,

Estimator: $\hat{x}(k+1) \;=\; (A - LC)\,\hat{x}(k) \;+\; B\,u(k) \;+\; L\,y(k).$

To do this we use the estimated state to calculate the feedback:

$$u(k) \;=\; -K\,\hat{x}(k).$$

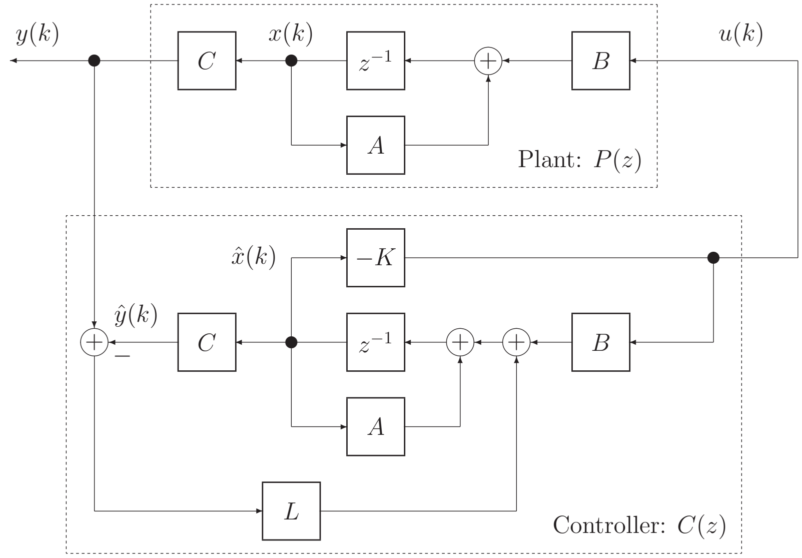
This idea is also referred to as *"certainty equivalence"*; using our best estimate in place of reality.

But will it work?

What is the effect of using $\hat{x}(k)$ in place of $x(k)$?

How will this affect the closed-loop poles of the plant?

The big picture: using both K and L to create $C(z)$



Closed-loop dynamics

The complete closed-loop system has $2n$ poles (n in the plant and n in the controller).

To derive the complete $2n \times 2n$ state-space representation consider,

$$\begin{aligned} x(k+1) &= A\,x(k) \;+\; B\,u(k) \\ &= A\,x(k) \;-\; BK\,\hat{x}(k) \\ &= A\,x(k) \;-\; BK\,(x(k) - \tilde{x}(k)) \\ &= (A - BK)\,x(k) \;+\; BK\,\tilde{x}(k) \end{aligned}$$

This gives an equation for n of the states. To get the other n consider,

$$\tilde{x}(k+1) \;=\; (A - LC)\,\tilde{x}(k).$$

Putting these together gives:

$$\begin{bmatrix} x(k+1) \\ \tilde{x}(k+1) \end{bmatrix} \;=\; \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x(k) \\ \tilde{x}(k) \end{bmatrix}.$$

Closed-loop dynamics

Because this is block upper triangular,

$$\mathrm{eig}\left(\begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix}\right) \;=\; \{\,\mathrm{eig}(A - BK)\,,\,\mathrm{eig}(A - LC)\,\}\,.$$

The closed loop poles of the plant are not changed!

The fact that we can design the two parts (state feedback and estimator) independently, and still get the same closed-loop dynamics when we join them together is known as the *"separation principle."*

Transient effects

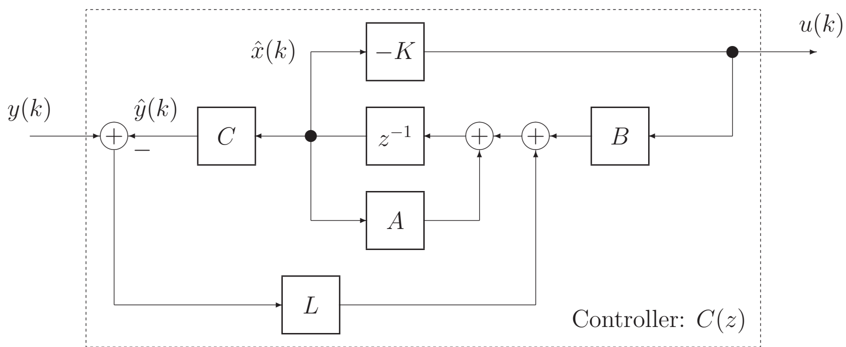
Estimator errors ($\tilde{x}(k) \neq 0$) don't affect the closed-loop poles of the plant, but they do cause an error transient.

This comes from the $BK\,\tilde{x}(k)$ term in the $x(k+1)$ update equation.

$$\begin{bmatrix} x(k+1) \\ \tilde{x}(k+1) \end{bmatrix} \;=\; \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x(k) \\ \tilde{x}(k) \end{bmatrix}.$$

Because $\tilde{x}(k) \longrightarrow 0$ as $k \longrightarrow \infty$ (if we designed L properly), this transient affect decays away.

What does the controller look like?



$$\begin{aligned} \hat{x}(k+1) &= A\,\hat{x}(k) \;+\; B\,u(k) \;+\; L(y(k) - C\,\hat{x}(k)) \\ &= (A - LC)\,\hat{x}(k) \;+\; B\,u(k) \;+\; L\,y(k) \\ &= (A - LC - BK)\,\hat{x}(k) \;+\; L\,y(k) \qquad \longleftarrow \text{dynamics are specified by } L \text{ and } K \\ \text{and} \quad u(k) &= -K\,\hat{x}(k) \end{aligned}$$

How should we design L?

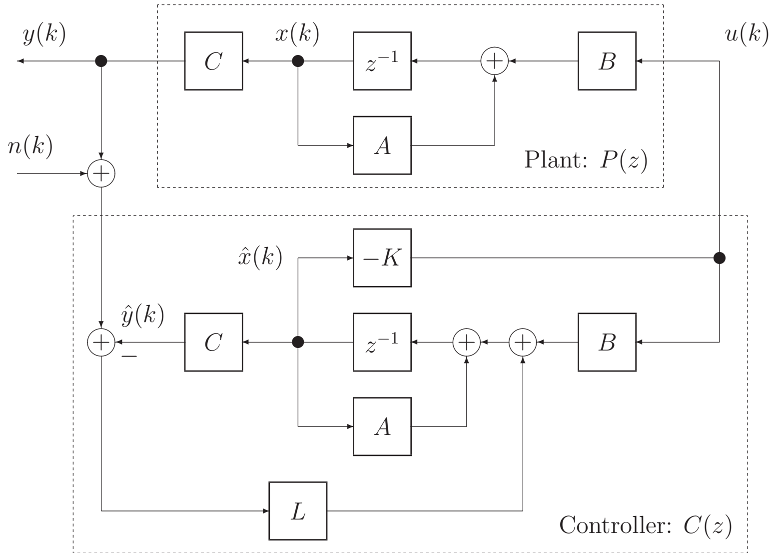
If we make the poles of $A - LC$ fast the estimator error transient will decay quickly.

What is the tradeoff here though?

What prevents us from putting the poles as close to zero as we want?

Is there a problem if L is large?

Noise!



Tradeoffs in the design of L

Noise enters the estimator equations,

$$\hat{x}(k+1) \;=\; (A - LC - BK)\,\hat{x}(k) \;+\; L\,y(k) \;+\; L\,n(k),$$

and corrupts the estimate.

The error is now given by,

$$\tilde{x}(k+1) \;=\; (A - LC)\,\tilde{x}(k) \;-\; L\,n(k),$$

so, although it is still stable, it doesn't decay to zero if $n(k) \neq 0$.

Note that the larger the size of L, the more the noise affects the error.

The complete closed-loop system is:

$$\begin{bmatrix} x(k+1) \\ \tilde{x}(k+1) \end{bmatrix} \;=\; \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x(k) \\ \tilde{x}(k) \end{bmatrix} \;+\; \begin{bmatrix} 0 \\ -L \end{bmatrix} n(k).$$

Rule of Thumb: The estimator error poles should be placed 2 to 4 times faster than the closed-loop poles.

This is a trade-off between estimator error transients and noise magnification.

A final caveat:

The separation principle depended on the estimator model and the plant model being linear and **identical**.

In reality this never happens. Our estimator model is only an approximation to physical reality.

Our closed-loop poles will move as a result of these modeling errors.

Estimator sensitivity to errors in A:

- Because of the error feedback (via $y(k)$), $A - LC$ is usually less sensitive than A to errors between the model dynamics and the true system dynamics.
- However, too large a gain, L, may actually increase the sensitivity to errors.
- This is analagous to the Bode integral limitations on using output feedback to reduce the closed-loop sensitivity.

Finding the correct trade-off between noise amplification, modeling error sensitivity, and the estimation error dynamics, is often a matter of experiment.