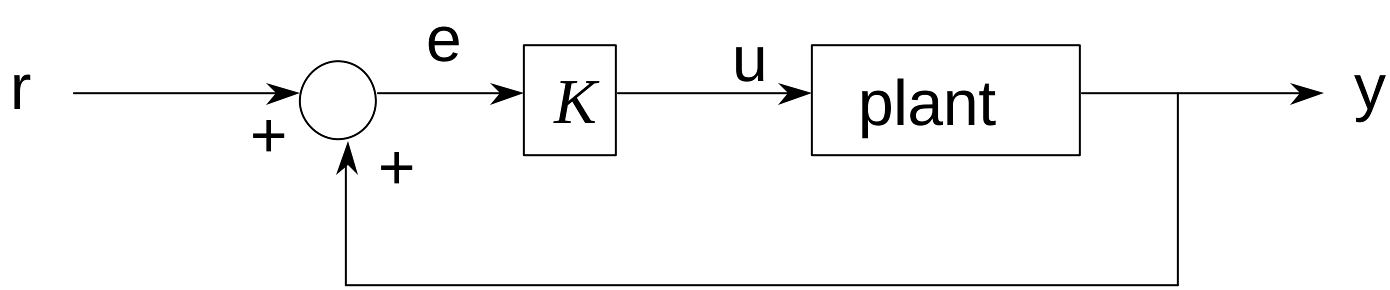
Chapter 10: State Feedback and Observers

Recall the most basic controller; the feedback controller with gain K.



Note that we are taking the scalar output variable and feeding it back, through a gain factor.

By using standard control techniques, we can choose K for various performance criteria. We could also make the controller into a transfer function itself, and use root-locus, Bode plots, or another design tool to ***place the closed-loop poles***.

Poles can also be "placed" using the technique of **State-Variable Feedback**:

$$\dot{x} = Ax + bu$$
$$y = cx + du$$

Assume a single input / single output system.

Suppose we construct "state-feedback"

$$u = Kx + Ev$$

new "reference" input

"feedforward" gain (usually I)

state vector (nx1)

state feedback gain matrix (1xn)

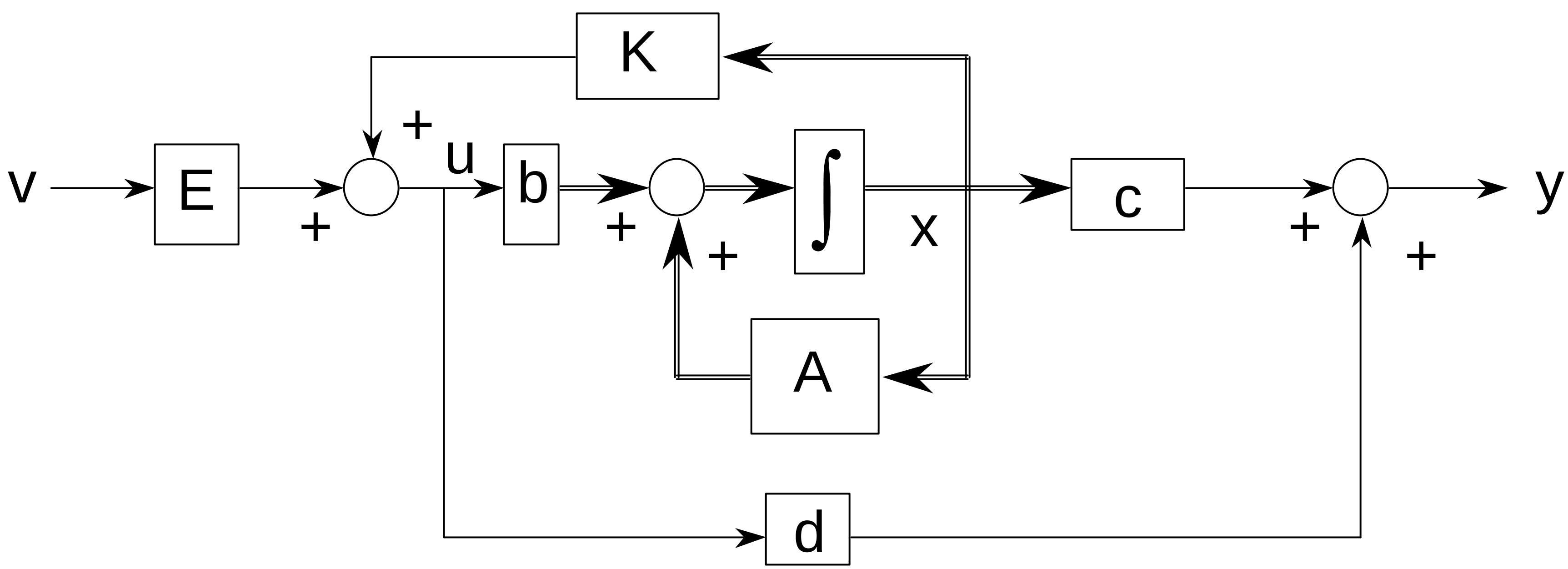
Then

$$\dot{x} = Ax + b(Kx + Ev)$$
$$= (A + bK)x + bEv$$

$$\dot{x} = Ax + b(Kx + Ev)$$

$$= \boxed{(A + bK)}x + bEv$$

Our goal is to select a gain matrix K so that this new "system" matrix has eigenvalues where we want them, rather than using those of the original A-matrix.



There is a special case when we can easily find a gain matrix K in order to choose any eigenvalues we want for the "closed-loop" system: the **controllable canonical form**.

$$A = \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & 0 & \ddots & \\ & & & \ddots & 1 \\ -a_0 & -a_1 & \cdots & -a_{n-2} & -a_{n-1} \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

$$c = [\text{ arbitrary }]$$

Characteristic Polynomial: $\phi(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda^1 + a_0$

Consider the *controllability matrix* for this system:

$$P = \left[b \;\vdots\; Ab \;\vdots\; \cdots \;\vdots\; A^{n-1}b \right]$$

$$= \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ & \vdots & \vdots & \iddots & -a_{n-1} \\ \vdots & 0 & 1 & -a_{n-1} & \\ 0 & 1 & \iddots & & \\ 1 & -a_{n-1} & & & (\neq 0) \end{bmatrix}$$

This matrix will obviously have rank *n* iff the b matrix has a non-zero value as its final element (hence the name "controllable canonical form").

One may ask: How do we know a controllable system is still controllable *after* state feedback?

Theorem: Let the *closed loop* controllability matrix be denoted (assuming $E = I$):

reference scaling matrix

$$P_{CL} = \left[b \;\vdots\; (A + bK)b \;\vdots\; \cdots \;\vdots\; (A + bK)^{n-1} b \right]$$

Then rank(P_{CL}) = rank(P). (so the CL system is controllable iff the open-loop system is controllable.)

PROOF: It can be shown that

$$\left[b \;\vdots\; (A+bK)b \;\vdots\; (A+bK)^2 b \;\vdots\; \cdots \;\vdots\; (A+bK)^{n-1} b \right] =$$

$$\left[b \;\vdots\; Ab \;\vdots\; \cdots \;\vdots\; A^{n-1} b \right] \begin{bmatrix} I & Kb & K(A+bK)b & \cdots & K(A+bK)^{n-2}b \\ 0 & I & +Kb & & \\ & & I & \ddots & \\ \vdots & & & \ddots & +Kb \\ 0 & & \cdots & 0 & I \end{bmatrix}$$

Because this matrix has one's on the diagonal, it is nonsingular. Because it is nonsingular, it does not change the rank of the matrix it multiplies, which is just the original controllability matrix. ■

NOTE: This result does ***<u>NOT</u>*** extend to observability! That is, state feedback *might* make a previously observable system unobservable!!!

For simplicity, let's use a zero input and compute what happens with feedback matrix K. Let (i.e., $V = 0$)

$$K = \begin{bmatrix} k_0 & k_1 & \cdots & k_{n-1} \end{bmatrix}$$

so for the controllable canonical form:

$$bK = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} k_0 & k_1 & \cdots & k_{n-1} \end{bmatrix} = \left[\begin{array}{cccc} & & \mathbf{0} & \\ \hdashline k_0 & k_1 & \cdots & k_{n-1} \end{array} \right]$$

Giving:

$$A + bK = \left[\begin{array}{cccc} 0 & 1 & 0 & \cdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \\ \hdashline -a_0 + k_0 & -a_1 + k_1 & \cdots & -a_{n-1} + k_{n-1} \end{array} \right]$$

So that the characteristic equation of this closed loop matrix is:

$$\phi(\lambda) = \lambda^n + (a_{n-1} - k_{n-1})\lambda^{n-1} + \cdots + (a_0 - k_0)$$

It is clear that because we are allowed to choose the k's arbitrarily, we can assign, or "place" all of the eigenvalues wherever we want them (provided that any complex ones occur in complex conjugate pairs).

Because any controllable system can be transformed into controllable canonical form, we make make the statement:

controllability $\longleftrightarrow$ the ability to place the poles anywhere through *state feedback*

This gives us a technique to stabilize unstable systems and do much more:

Example: Consider the system

$$\dot{x} = Ax + bu, \qquad \text{where } A = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

The eigenvalues of the A-matrix are -2 and +5, so the system is initially unstable. We first examine the controllability of the system to see if there's any hope to stabilize it:

$$P = [b \quad Ab] = \begin{bmatrix} 1 & 4 \\ 1 & 6 \end{bmatrix}$$

This matrix has full rank so the system is controllable.

Suppose we decide we would like the closed-loop poles (eigenvalues) of the system to be at -5 and -6. We compute the result of state feedback:

$$A + bK = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} k_0 & k_1 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 4 & 2 \end{bmatrix} + \begin{bmatrix} k_0 & k_1 \\ k_0 & k_1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 + k_0 & 3 + k_1 \\ 4 + k_0 & 2 + k_1 \end{bmatrix}$$

So the characteristic equation is:

$$\phi(\lambda) = \begin{vmatrix} \lambda - k_0 - 1 & -3 - k_1 \\ -4 - k_0 & \lambda - 2 - k_1 \end{vmatrix} =$$ A BIG MESS TO DEAL WITH!!

To make the algebra easier, we compute the controllable canonical form:

The inverse of the controllability matrix is:

$$P^{-1} = \begin{bmatrix} 3 & -2 \\ -.5 & .5 \end{bmatrix}$$

So if we compute our similarity transformation as we did in chapter 8,

$$U^{-1} = \begin{bmatrix} -.5 & .5 \\ 1.5 & -.5 \end{bmatrix}$$

Then

$$\overline{A} = U^{-1}AU = \begin{bmatrix} 0 & 1 \\ 10 & 3 \end{bmatrix} \text{ and } \quad \overline{b} = U^{-1}b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Now use this form to compute the state feedback:

$$\overline{A} + \overline{b}\,\overline{K} = \begin{bmatrix} 0 & 1 \\ 10 & 3 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ \overline{k}_0 & \overline{k}_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 10 + \overline{k}_0 & 3 + \overline{k}_1 \end{bmatrix}$$

For which the characteristic equation is:

$$\phi(\lambda) = \lambda^2 + (-\overline{k}_1 - 3)\lambda + (-\overline{k}_0 - 10)$$

$$\phi(\lambda) = \lambda^2 + (-\bar{k}_1 - 3)\lambda + (-\bar{k}_0 - 10)$$

If we desire these poles to be at -5 and -6, then

$$f^{des}(l) = l^2 + 11l + 30 = (l + 5)(l + 6)$$

So by inspection we get:

$$\bar{K} = [\bar{k}_0 \quad \bar{k}_1] = [-40 \quad -14]$$

The state-feedback for the ***controllable canonical form*** is therefore:

$$u = \bar{K}\bar{x}$$

But because of the similarity transformation we performed,

$$\bar{x} = U^{-1}x$$

So $$u = \bar{K}\,\bar{x} = \bar{K}\,U^{-1}x = Kx$$

where $$K = \bar{K}U^{-1}$$

So now because we are implementing state-feedback in our original system, we use feedback gain

$$u = Kx = \overset{\bar{K}}{\begin{bmatrix} -40 & -14 \end{bmatrix}} \overset{U^{-1}}{\begin{bmatrix} -.5 & .5 \\ 1.5 & -.5 \end{bmatrix}} x = \overset{K}{\begin{bmatrix} -1 & -13 \end{bmatrix}} x$$

To check,

$$\sigma\left(A + bK\right) = \sigma\left(A_{CL}\right) = \sigma\left(\begin{bmatrix} 0 & -10 \\ 3 & -11 \end{bmatrix}\right) = \{-5, -6\}$$

So it is very easy to compute the proper state feedback **if** the system happens to be in controllable canonical form. Sometimes it is inconvenient to do this transformation (and then back again), so we have a famous formula: *Ackermann's Formula*

First we point out that there is an easier way to compute the similarity transformation matrix when we know the characteristic equation. We'll use this result in the derivation of Ackermann's Formula.

Note that

$$\overline{A} = U^{-1}AU \quad \text{and} \quad \overline{b} = U^{-1}b \quad \text{because} \quad x = U\overline{x}$$

Then the controllability matrix for the controllable form is:

$$\begin{aligned}
\overline{P} &= \left[\overline{b} \;\vdots\; \overline{A}\,\overline{b} \;\vdots\; \cdots \;\vdots\; \overline{A}^{n-1}\overline{b}\right] \\
&= \left[U^{-1}b \;\vdots\; U^{-1}Ab \;\vdots\; \cdots \;\vdots\; U^{-1}A^{n-1}b\right] \\
&= U^{-1}P
\end{aligned}$$

So

$$U = P\overline{P}^{-1}$$

Transformation is related to the two system's P matrices

Now for the derivation of Ackermann's formula:

When we applied state feedback to the controllable canonical form, we got a closed-loop matrix of the form:

$$A_{CL} = \bar{A} + \bar{b}\bar{K} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \\ -a_0 + \bar{k}_0 & -a_1 + \bar{k}_1 & \cdots & -a_{n-1} + \bar{k}_{n-1} \end{bmatrix}$$

So

$$\left.\phi(\lambda) = \lambda^n + (a_{n-1} - \bar{k}_{n-1})\lambda^{n-1} + \cdots + (a_1 - \bar{k}_1)\lambda + (a_0 - \bar{k}_0)\right\}$$

For A_{CL}

Denote the *desired* closed-loop characteristic polynomial as:

$$\phi^{des}(\lambda) = \lambda^n + a_{n-1}^{des}\lambda^{n-1} + \cdots + a_1^{des}\lambda + a_0^{des}$$

Equate above equation with the last equation on 334

So we have the equalities:

$$a_{n-1}^{des} - a_{n-1} = -\bar{k}_{n-1} \quad \cdots \quad a_1^{des} - a_1 = -\bar{k}_1 \quad a_0^{des} - a_0 = -\bar{k}_0 \Big\}$$ **Hold equations for a minute**

Suppose we plug $\bar{A}$ into this equation:

controllable cannoical form

$$f^{des}(\bar{A}) = \bar{A}^n + a_{n-1}^{des}\bar{A}^{n-1} + \cdots + a_1^{des}\bar{A} + a_0^{des} I \Big\}$$ $\bar{A}$ into f^{des}

$$f(l) = l^n + a_{n-1} l^{n-1} + \cdots + a_1^{des}\bar{A} + a_0^{des} I \Big\}$$ For $\bar{A}$

We know from the Cayley-Hamilton Theorem that:

$$\overline{A}^{n} + a_{n-1}\overline{A}^{n-1} + \cdots + a_1\overline{A} + a_0 I = 0$$

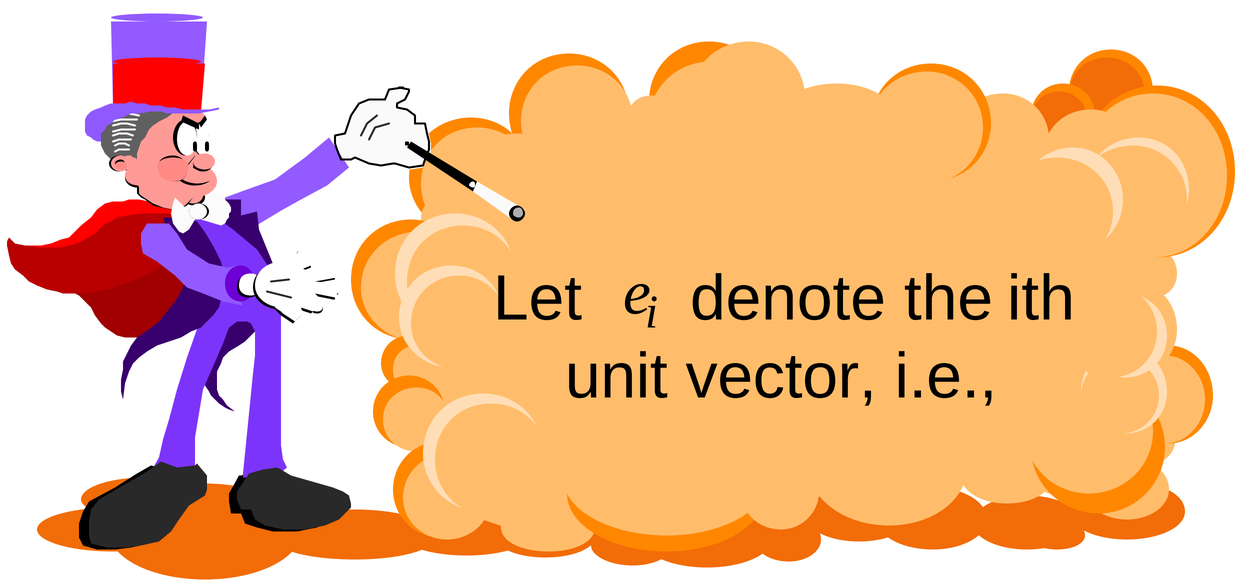
solve for $\overline{A}^{n}$

substitute into

Substituting into $\phi^{des}(\overline{A})$:

$$\phi^{des}(\overline{A}) = (a_{n-1}^{des} - a_{n-1})\overline{A}^{n-1} + \cdots + (a_1^{des} - a_1)\overline{A} + (a_0^{des} - a_0)I$$

Now here's the trick:



$$e_1 = [1 \quad 0 \quad \cdots \quad 0]^T, \quad e_2 = [0 \quad 1 \quad 0 \quad \cdots \quad 0]^T, \quad etc.$$

Now notice that:

$$\left. e_1^T \overline{A} = e_2^T \quad \cdots \quad e_i^T \overline{A} = e_{i+1}^T \quad \cdots \quad e_{n-1}^T \overline{A} = e_n^T \right\}$$

property of the controllable canonical form

$e_n^T \overline{A} = $ [last row of $\overline{A}$] (But this is unimportant)

Multiply these relations *again* by $\overline{A}$ *from the right*:

$$e_1^T \overline{A}\,\overline{A} = e_2^T \overline{A} = e_3^T$$

, etc., up to : ☞

$$\boxed{e_1^T \bar{A}^{n-1} = e_n^T}$$

and use this relationship

So now if we multiply $\phi^{des}(\bar{A})$ by e_1^T we get:

$$
\begin{aligned}
e_1^T \phi^{des}(\bar{A}) &= (a_{n-1}^{des} - a_{n-1}) e_1^T \bar{A}^{n-1} + \cdots + (a_1^{des} - a_1) e_1^T \bar{A} + (a_0^{des} - a_0) e_1^T \\
&= (a_{n-1}^{des} - a_{n-1}) e_n^T + \cdots + (a_1^{des} - a_1) e_2^T + (a_0^{des} - a_0) e_1^T \\
&= -(\bar{k}_{n-1}) e_n^T - \cdots - (\bar{k}_1) e_2^T - (\bar{k}_0) e_1^T \quad \Big\} \text{ use formula for } \bar{K} \\
&= \begin{bmatrix} -\bar{k}_0 & -\bar{k}_1 & \cdots & -\bar{k}_{n-1} \end{bmatrix} \\
&= -\bar{K} \\
&= -KU \quad \Big\} \text{ by definition}
\end{aligned}
$$

This formula is still not useful because it requires knowledge of the similarity transformation matrix U and the controllable form $\bar{A}$.

$$e_1^T \phi^{des}(\overline{A}) = -KU$$ } now manipulate this equation

Recall that $\overline{A} = U^{-1} A U$

Substitute this into the formula at the top of the page:

$$e_1^T U^{-1} \phi^{des}(A) U U^{-1} = -K,$$ } postmultiply by U^{-1}

$$e_1^T U^{-1} \phi^{des}(A) = -K$$

We have already shown that $U = P\overline{P}^{-1}$ so $U^{-1} = \overline{P} P^{-1}$

$$K = -e_1^T \overline{P} P^{-1} \phi^{des}(A)$$

Note $f^{des}\left(U^{-1} A U\right) = U^{-1} f^{des}(A) U$ } see page 215

$$K = -\underbrace{e_1^{\mathrm{T}} \overline{P}}\, P^{-1} \phi^{des}(A)$$

$$\begin{array}{cc} e_1^{T} & \overline{P} \end{array}$$

$$e_1^{T} \overline{P} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ & & & 1 & \\ \vdots & & \iddots & & \\ 0 & 1 & & \ddots & \\ 1 & & & & \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 & 1 \end{bmatrix} = e_n^{T}$$

So finally,

$$\boxed{K = -e_n^{T} P^{-1} \phi^{des}(A)} \Big\} \text{ formula for calculations}$$

Ackermann's Formula

Note that for this formula, we need only the open-loop A-matrix, the controllability matrix P, and the desired characteristic polynomial.

However, it does require the inverse of a matrix, which is not always advisable for numerical accuracy reasons; especially when the system is "weakly" controllable. In fact, MATLAB offers the **ACKER** command, but advises against ever using it!

There is a way to make it work better by using numerically robust methods for solving simultaneous equations rather than computing matrix inverses:

$$\boxed{K = -e_n^T P^{-1} \phi^{des}(A)}$$

Define $$f^T = -e_n^T P^{-1}$$

Then $$f^T P = -e_n^T \text{ or}$$

$$P^T f = -e_n$$

If we solve this as a set of linear simultaneous equations (which is done in MATLAB without the use of inverses), then we can use the solution to compute feedback K:

$$\boxed{K = f^T \phi^{des}(A)}$$

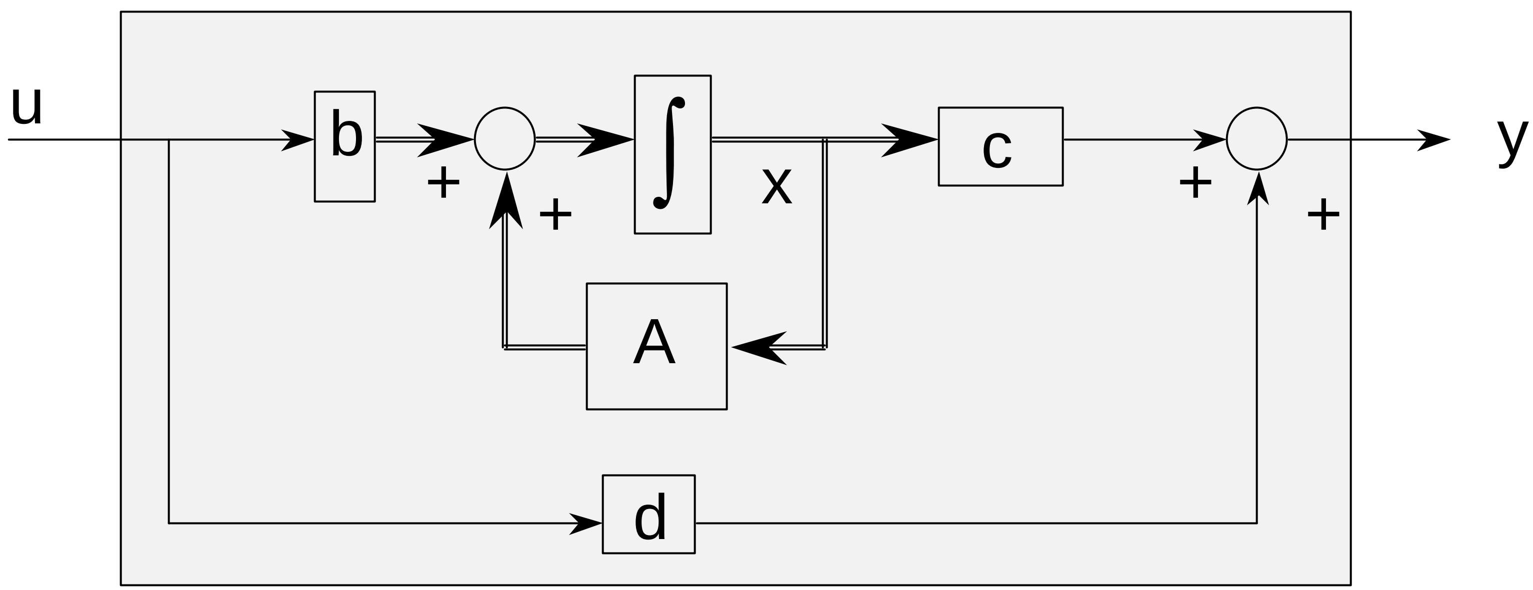
Note: One can show that the numerator of the transfer function is the same before and after state-feedback. This implies that

The zeros of a system are not affected by state feedback.

This also helps explain why state-feedback might affect the ***observability*** of a system: Suppose state feedback were used to place a pole of a system at the *same place* as a zero. Then these modes would not appear in the output, through pole-zero cancellation (and by the definition of a zero).

Full State Observers (Estimators):

State feedback is relatively easy, but we have assumed throughout that we have access to all the signals in the state vector x in order to construct the feedback controller. Actually, we will usually only have physical access to the input and output of a system, with the state-variables being "internal."



To get around this problem, we can show how to build another system, called an **observer** (or **estimator**) that re-constructs the state vector from the system input and output, and allows us to use *its* output for state feedback.

Begin with the system

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

(this procedure holds for multivariable systems)

And assume we know the matrices $\{A, B, C, D\}$.

First let's consider a naive approach:

If we know the initial condition on the state vector and the A and B matrices, we can construct a new system giving an "estimated" state vector $\hat{x}(t)$:

$$\dot{\hat{x}} = A\hat{x} + Bu$$
$$\hat{y} = C\hat{x} + Du, \qquad \hat{x}(0) = x(0)$$

If the world is perfect and we expect no external disturbances or other sources of error, this might work sufficiently well. However, it is an *open-loop* estimator, so that in the presence of uncertainties or imperfections (especially if the original system happens to be unstable), the estimated value $\hat{x}(t)$ will eventually diverge from the true value of $x(t)$.

Instead we seek a **closed-loop** observer:

We construct a new dynamic system, which is similar in construction to the original system, but whose states are the ***estimates*** of $x(t)$ and which has two "inputs," the original system's u and the error between the true output y and the estimated output $\hat{y}$. Inclusion of $y-\hat{y}$ as an input "closes the loop." It is therefore "driven" by the output error:

build this

$$\left\{ \dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \right\} \quad \hat{y} = C\hat{x} + Du$$

$$\dot{\hat{x}} = A\hat{x} + Bu + L\big(y - (C\hat{x} + Du)\big)$$

for analysis

$$\left\{ \dot{\hat{x}} = (A - LC)\hat{x} + (B - LD)u + Ly \right.$$

Now consider the "error" in the observation that results.
Define

$$\tilde{x} \overset{\Delta}{=} x - \hat{x}$$

$$\dot{x} = Ax + Bu \qquad\qquad y = Cx + Du$$

$$\dot{\hat{x}} = (A - LC)\hat{x} + (B - LD)u + Ly$$

Then

$$\begin{aligned}
\dot{\tilde{x}} &= \dot{x} - \dot{\hat{x}} \\
&= Ax + Bu - (A - LC)\hat{x} - (B - LD)u - L(Cx + Du) \\
&= A(x - \hat{x}) - LC(x - \hat{x}) \\
&= (A - LC)(x - \hat{x}) \\
&= (A - LC)\tilde{x}
\end{aligned}$$

closed-loop system

Because $\tilde{x}$ represents the *error* signal, we would like this set of equations to be **asymptotically stable**, so that the eigenvalues of $(A\text{-}LC)$ are in the left half-plane. We can place them there by choosing an appropriate L matrix *as if it were a state-feedback gain*.

Note that the L does not appear in this expression exactly as the K does in $(A+BK)$, so the formulas don't hold exactly.

We can "fix" this by realizing that the eigenvalues of a matrix are always the same as the eigenvalues of its transpose. We therefore use pole-placement methods in order to compute L from the pole-placement problem:

$$A^T - C^T L^T$$

To do this, we must have the pair $\{A^T, C^T\}$ being *controllable*, which is the same as saying $\{A, C\}$ is ***observable***!!

Now that the error system is stable,

$$\tilde{x}(t) \to 0 \text{ as } t \to \infty$$

so

$$\hat{x}(t) \to x(t) \text{ as } t \to \infty$$

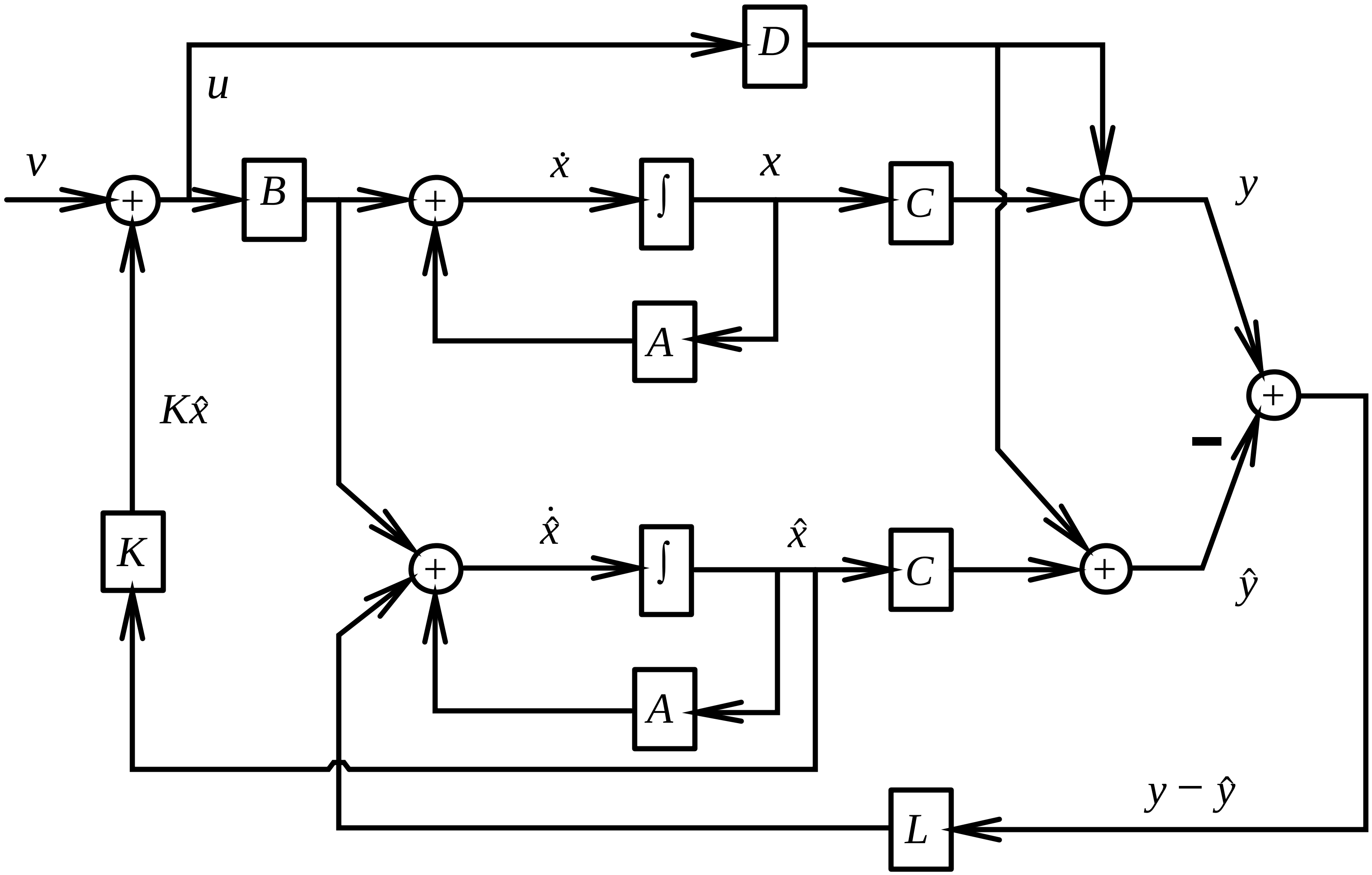
We then use these estimated state-variables to construct the state-feedback control law as before:

$$u = K\hat{x} + v \qquad (E = I) \Big\} \text{ assume } E=I$$

? We usually place the poles of the closed-loop system according to a performance criterion, but where should one place the poles of the *estimator* system?

! They should be placed "farther left" than the poles of the system dynamics. That is, the estimator dynamics should be *faster* than the plant dynamics. That way, the true state variable do no "outrun" the observer variables that are trying to estimate them!

Block Diagram:



How would you simulate the entire system, controller, ($u = K\hat{x} + v$), observer and all? Create an "augmented-state" system:

First notice the "observer dynamics":

$$\begin{aligned}\dot{\hat{x}} &= (A - LC)\hat{x} + (B - LD)u + Ly \\ &= \underbrace{(A - LC)\hat{x} + (B - LD)(K\hat{x} + v) + L(Cx + D(K\hat{x} + v))}_{\text{substitute for } u(t) \text{ and } y(t)} \\ &= \underbrace{[A - LC + BK]\hat{x} + LCx + Bv}_{\text{simplify}}\end{aligned}$$

Together with the plant: $\underbrace{\dot{x} = Ax + B(K\hat{x} + v)}_{\text{substitute } u(t)}$

$$\begin{bmatrix}\dot{x} \\ \dot{\hat{x}}\end{bmatrix} = \begin{bmatrix}A & BK \\ LC & A - LC + BK\end{bmatrix}\begin{bmatrix}x \\ \hat{x}\end{bmatrix} + \boxed{\begin{bmatrix}B \\ B\end{bmatrix}v}$$

necessary for external input

Treat these as the "new" system matrices.

How do we know that attaching the observer onto the plant does not change the eigenvalues of the closed-loop system? That is, how do we know that we can choose the gain matrices K and L *independently* of one-another?

Consider the plant and *error* dynamics together:

$$\begin{aligned} \dot{x} &= Ax + BK\hat{x} && \Big\}\ \text{let } V = 0 \\ &= Ax + BK(x - \tilde{x}) && \Big\}\ \tilde{x} = x - \hat{x} \\ &= (A + BK)x - BK\tilde{x} \end{aligned}$$

$$\dot{\tilde{x}} = (A - LC)\tilde{x} \quad \Big\}\ \text{from before}$$

So together:

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A + BK & -BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix}$$

$$\begin{bmatrix} \dot{x} \\ \dot{\tilde{x}} \end{bmatrix} = \begin{bmatrix} A + BK & -BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix}$$

When we try to find the eigenvalues of this system:

$$\det\left(\begin{bmatrix} A + BK & -BK \\ 0 & A - LC \end{bmatrix} - \lambda I \right) = \begin{vmatrix} A + BK - \lambda I & -BK \\ 0 & A - LC - \lambda I \end{vmatrix}$$

$$= |A + BK - \lambda I| |A - LC - \lambda I|$$

So we see that part of the eigenvalues are determined by choice of K alone (the closed-loop plant eigenvalues), and the others are determined by L alone (observer eigenvalues).

This important result is called the ***separation principal***.

Reduced-Order Observers:

We often have available **part** of the state vector for feedback; for example if the output equation is

$$y = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

either naturally or through a similarity transformation to make it that way.

If this is the case, then we can save time and money by using a reduced-order observer, that just estimates the missing part of the state vector, x_2 .

Unfortunately, it will be more complex to derive.

Step one: How to transform the system such that the C-matrix has the form $\begin{bmatrix} I & 0 \end{bmatrix}$

First augment the C-matrix to create a nonsingular matrix:

$$W = \begin{bmatrix} C \\ R \end{bmatrix} \quad \begin{matrix} \} \text{ all q rows from the original C-matrix} \\ \} \; n - q \text{ more rows lin. indep. of the first q rows} \end{matrix}$$

Now compute $V = W^{-1} = \begin{bmatrix} V_1 & \vdots & V_2 \end{bmatrix}$ so that

$$WV = I = \begin{bmatrix} I_{q\times q} & 0 \\ 0 & I_{(n-q)\times(n-q)} \end{bmatrix} = \begin{bmatrix} C \\ R \end{bmatrix} \begin{bmatrix} V_1 & V_2 \end{bmatrix}$$

Now if we use V as a similarity transformation:

$$\dot{\bar{x}} = V^{-1}AV\bar{x} + V^{-1}Bu$$
$$y = CV\bar{x} = \begin{bmatrix} CV_1 & CV_2 \end{bmatrix}\bar{x} = \begin{bmatrix} I_{q\times q} & 0 \end{bmatrix}\bar{x} = \bar{x}_1$$

where $\bar{x}_1$ is clearly a vector of the first q elements of the transformed state vector $\bar{x}$. Partition this transformed system to the form:

$$\begin{bmatrix} \dot{\bar{x}}_1 \\ \dot{\bar{x}}_2 \end{bmatrix} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} + \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \end{bmatrix} u$$
$$y = \begin{bmatrix} I_{q\times q} & 0 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} + Du$$

So we *have* $\bar{x}_1$ and we would like to *observe* $\bar{x}_2$.

Step two: Strategy: Find equations for $\bar{x}_2$ alone; i.e., considering $\bar{x}_1$ to be a *known* signal:

$$\bar{A}_{12}\bar{x}_2 = \boxed{\dot{\bar{x}}_1 - \bar{A}_{11}y - \bar{B}_1 u} \quad \text{new output}$$

$$\dot{\bar{x}}_2 = \bar{A}_{22}\bar{x}_2 + \underbrace{\bar{A}_{21}\bar{x}_1 + \bar{B}_2 u}_{\bar{u}}$$

Define some new variables:

$$\bar{u} \stackrel{\Delta}{=} \bar{A}_{21}\bar{x}_1 + \bar{B}_2 u$$

$$\bar{y} \stackrel{\Delta}{=} \boxed{\dot{\bar{x}}_1 - \bar{A}_{11}\bar{x}_1 - \bar{B}_1 u}$$

functions of known (available) signals.

more about $\dot{\bar{x}}_1$ later

Then

$$\dot{\bar{x}}_2 = \bar{A}_{22}\bar{x}_2 + \bar{u}$$

$$\bar{y} = \bar{A}_{12}\bar{x}_2$$

Treat these as a new set of state- and output equations.

$$\boxed{\begin{aligned} \dot{\bar{x}}_2 &= \bar{A}_{22}\bar{x}_2 + \bar{u} \\ \bar{y} &= \bar{A}_{12}\bar{x}_2 \end{aligned}}$$

One can show that if the original system $\{A, C\}$ is observable, then this new system $\{\bar{A}_{22}, \bar{A}_{12}\}$ is also observable.

Step three: Simply design a full-order observer for this reduced set of equations:

$$\begin{aligned} \dot{\hat{\bar{x}}}_2 &= \bar{A}_{22}\hat{\bar{x}}_2 + \bar{u} + \bar{L}(\bar{y} - \hat{\bar{y}}) \quad \Big\} \ \bar{y} = \bar{A}_{12}\bar{x}_2 \\ &= \left(\bar{A}_{22} - \bar{L}\bar{A}_{12}\right)\hat{\bar{x}}_2 + \bar{u} + \bar{L}\bar{y} \end{aligned}$$

where the eigenvalues of $(\bar{A}_{22} - \bar{L}\bar{A}_{12})$ are placed *to the left of* the eigenvalues of $\bar{A}_{22}$ through proper choice of matrix $\bar{L}$.

However, recall that we are considering $\bar{x}_1$ to be known, and that $\bar{y}$ depends on $\dot{\bar{x}}_1$, a pure time derivative, which is difficult to compute.

To get around this problem, perform the following change of variable:

$$z = \hat{\bar{x}}_2 - \bar{L}\bar{x}_1 \qquad \left(\hat{\bar{x}}_2 = z + \bar{L}\bar{x}_1\right)$$

$$\dot{z} = \dot{\hat{\bar{x}}}_2 - \bar{L}\dot{\bar{x}}_1$$

subst. $\bar{y}$

Then:

$$\dot{z} = \left(\bar{A}_{22} - \bar{L}\bar{A}_{12}\right)\left(z + \bar{L}\bar{x}_1\right) + \bar{u} + \bar{L}\bar{y} - \bar{L}\dot{\bar{x}}_1 \Big\}$$

subst. for $\dot{\bar{x}}_2$

$$= \left(\bar{A}_{22} - \bar{L}\bar{A}_{12}\right)\left(z + \bar{L}\bar{x}_1\right) + (\bar{A}_{21}\bar{x}_1 + \bar{B}_2 u) + \bar{L}(\dot{\bar{x}}_1 - \bar{A}_{11}\bar{x}_1 - \bar{B}_1 u) - \bar{L}\dot{\bar{x}}_1 \Big\}$$

$$= \left(\bar{A}_{22} - \bar{L}\bar{A}_{12}\right)z + \left(\bar{A}_{22} - \bar{L}\bar{A}_{12}\right)\bar{L}\bar{x}_1 + (\bar{A}_{21} - \bar{L}\bar{A}_{11})\bar{x}_1 + (\bar{B}_2 - \bar{L}\bar{B}_1)u$$

$$= \left(\bar{A}_{22} - \bar{L}\bar{A}_{12}\right)z + \big[\underbrace{\left(\bar{A}_{22} - \bar{L}\bar{A}_{12}\right)\bar{L} + (\bar{A}_{21} - \bar{L}\bar{A}_{11})}_{\text{factor out } \bar{x}_1}\big]\bar{x}_1 + (\bar{B}_2 - \bar{L}\bar{B}_1)u$$

This is the second equation to simulate (with the $\bar{x}_1$ equation) in order to find an estimate of $\bar{x}_2$.

Step four: Now because $\hat{\bar{x}}_2 = z + \bar{L}y$ is an *estimate* of $\bar{x}_2$, we can define an error signal $e = \bar{x}_2 - (z + \bar{L}\bar{x}_1)$. So

$$\dot{e} = \dot{\bar{x}}_2 - \dot{z} - \bar{L}\dot{\bar{x}}_1 \quad \Big\} \text{ subst. for } \dot{\bar{x}}_2, \dot{z}, \dot{\bar{x}}_1$$

$$= \cancel{\bar{A}_{21}\bar{x}_1} + \bar{A}_{22}\bar{x}_2 + \cancel{\bar{B}_2 u} - \left(\bar{A}_{22} - \bar{L}\bar{A}_{12}\right)\left(z + \bar{L}\bar{x}_1\right)$$

$$- \left(\cancel{\bar{A}_{21}} - \bar{L}\bar{A}_{11}\right)\bar{x}_1 - \left(\cancel{\bar{B}_2} - \bar{L}\bar{B}_1\right)u - \bar{L}\bar{A}_{11}\bar{x}_1 - \bar{L}\bar{A}_{12}\bar{x}_2 - \bar{L}\bar{B}_1 u$$

$$= \left(\bar{A}_{22} - \bar{L}\bar{A}_{12}\right)\underbrace{\left(\bar{x}_2 - z - \bar{L}\bar{x}_1\right)}_{e}$$

$$\dot{e} = \left(\bar{A}_{22} - \bar{L}\bar{A}_{12}\right)e$$

The eigenvalues of $\overline{A}_{22} - \overline{L}\,\overline{A}_{12}$ are placed to adjust the rate at which $z + \overline{L}y$ approaches $\overline{x}_2$.

Step five: Now the estimate for the whole state is:

$$\hat{\overline{x}} = \begin{bmatrix} \hat{\overline{x}}_1 \\ \hat{\overline{x}}_2 \end{bmatrix} = \begin{bmatrix} y - Du \\ z + \overline{L}\overline{x}_1 \end{bmatrix}$$

To un-do the original similarity transformation, recall that $\overline{x} = Wx \quad \left(x = V\overline{x}\right)$ so

$$\hat{x} = V\hat{\overline{x}} = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \begin{bmatrix} y - Du \\ z + \overline{L}\overline{x}_1 \end{bmatrix}$$

REMARKS:

Except for the similarity transformation, the reduced order observer requires fewer computations and integrations.

In the reduced order observer, the output variable y appears directly as an estimate for part of the state. This means that sensor noise appears in the state variable that is fed back. In the full-order observer, the state variable estimates are all the result of at least one integration which tends to smooth out any noise.

On the other hand, the reduced order observer can often have smaller transients, dur to part of the estimate being “perfect”.

How would one simulate the whole system??

First define an "augmented" state vector as

$$\xi = \begin{bmatrix} x \\ z \end{bmatrix} \quad \left(\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \cdots \right)$$

which will be a complicated expression! Then simulate the system using output equation

$$y = \begin{bmatrix} C & 0 \end{bmatrix} \xi$$

To compute *estimates* of the state variables,

$$\hat{x} = V \begin{bmatrix} y - Du \\ z + \overline{L}\overline{x}_1 \end{bmatrix}$$

And then when you want to apply state-feedback, use

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

or

$$u = K\hat{x} + v$$

when a non-zero *reference* input signal v is present.

Above all, do this in a well-documented script-file, so that the project will be re-usable and de-buggable!

Some further details: We have the original system

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

and

$$\dot{z} = (A_{22} - LA_{12})z + \left[(\bar{A}_{22} - \bar{L}\bar{A}_{12})\bar{L} + (\bar{A}_{21} - \bar{L}\bar{A}_{11})\right]\bar{x}_1 + (\bar{B}_2 - \bar{L}\bar{B}_1)u$$

but

$$\bar{x}_1 = y = \bar{C}\bar{x} = (CV)(V^{-1}x) = Cx$$

so

$$\dot{z} = (A_{22} - LA_{12})z + \left[(\bar{A}_{22} - \bar{L}\bar{A}_{12})\bar{L} + (\bar{A}_{21} - \bar{L}\bar{A}_{11})\right]Cx + (\bar{B}_2 - \bar{L}\bar{B}_1)u$$

These are in terms of x and z

Now apply the feedback: $u = K\hat{x} + v$

$$
\begin{aligned}
\dot{x} &= Ax + Bu = Ax + B\left(K\hat{x} + v\right) \\
&= Ax + BK\hat{x} + Bv \\
&= Ax + BK\left\{\begin{bmatrix} V_1 & V_2 \end{bmatrix}\begin{bmatrix} Cx \\ z + \overline{L}Cx \end{bmatrix}\right\} + Bv \\
&= Ax + BK\left[V_1 Cx + V_2 \overline{L} Cx + V_2 z\right] + Bv \\
&= \left[A + BK(V_1 + V_2 \overline{L})C\right]x + BKV_2 z + Bv
\end{aligned}
$$

This is one state equation, the other is for $\dot{z}$:

$$\dot{z} = (A_{22} - LA_{12})z + \left[(\overline{A}_{22} - \overline{L}\overline{A}_{12})\overline{L} + (\overline{A}_{21} - \overline{L}\overline{A}_{11})\right]Cx$$
$$+ (\overline{B}_2 - \overline{L}\overline{B}_1)u$$

$$u = K\hat{x} + v$$

$$= (A_{22} - LA_{12})z + \left[(\overline{A}_{22} - \overline{L}\overline{A}_{12})\overline{L} + (\overline{A}_{21} - \overline{L}\overline{A}_{11})\right]Cx$$
$$+ (\overline{B}_2 - \overline{L}\overline{B}_1)(K\left[V_1 Cx + V_2 \overline{L}Cx + V_2 z\right] + v)$$

$$= \left[(A_{22} - LA_{12}) + (\overline{B}_2 - \overline{L}\overline{B}_1)KV_2\right]z$$
$$+ \left[(\overline{A}_{22} - \overline{L}\overline{A}_{12})\overline{L} + (\overline{A}_{21} - \overline{L}\overline{A}_{11}) + (\overline{B}_2 - \overline{L}\overline{B}_1)K(V_1 + V_2\overline{L})\right]Cx$$
$$+ (\overline{B}_2 - \overline{L}\overline{B}_1)v$$

This is the second state equation. It is (n+(n-q)) dimensional. Just simulate the two *augmented* equations in MATLAB the result will be an ntime x (n+(n-q)) matrix (ntime = # of time points).

The first n columns will be the state variables of the controlled plant. The last n-q columns will be the *observed auxiliary* state variables *z*; not the actual observed plant state variables themselves.

Now if you want to get MATLAB to show you the actual observed state variables, we must compute:

$$\hat{x} = V\begin{bmatrix} Cx \quad (+Du) \\ z + \bar{L}Cx \end{bmatrix} = \begin{bmatrix} V_1 & V_2 \end{bmatrix}\begin{bmatrix} Cx \\ z + \bar{L}Cx \end{bmatrix}$$

$$= V_1 Cx + V_2 \bar{L} Cx + V_2 z$$

$$= \left(V_1 + V_2 \bar{L}\right) Cx + V_2 z$$

$$= \begin{bmatrix} \left(V_1 + V_2 \bar{L}\right) C & V_2 \end{bmatrix}\begin{bmatrix} x \\ z \end{bmatrix}$$

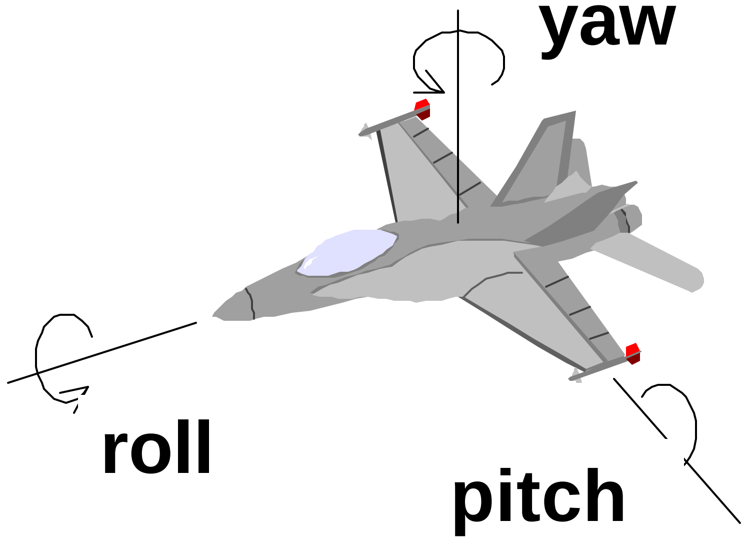
or, because MATLAB gives the state variables at *each* time in *rows*, instead of *columns*, use the command

```
xhat=([ (V1+V2*lbar)*C  V2]) * xaug' )' ;
```

where **`xaug`** is the *augmented* state vector

$$x_{aug} = \begin{bmatrix} x \\ z \end{bmatrix}$$

Try this in an example:



v : sideslip velocity

p : roll rate

r : yaw rate

ϕ : roll angle

ψ : yaw angle

ζ : rudder angle

ξ : aileron angle

$$x = [v \quad p \quad r \quad \phi \quad \psi \quad \zeta \quad \xi]^T$$

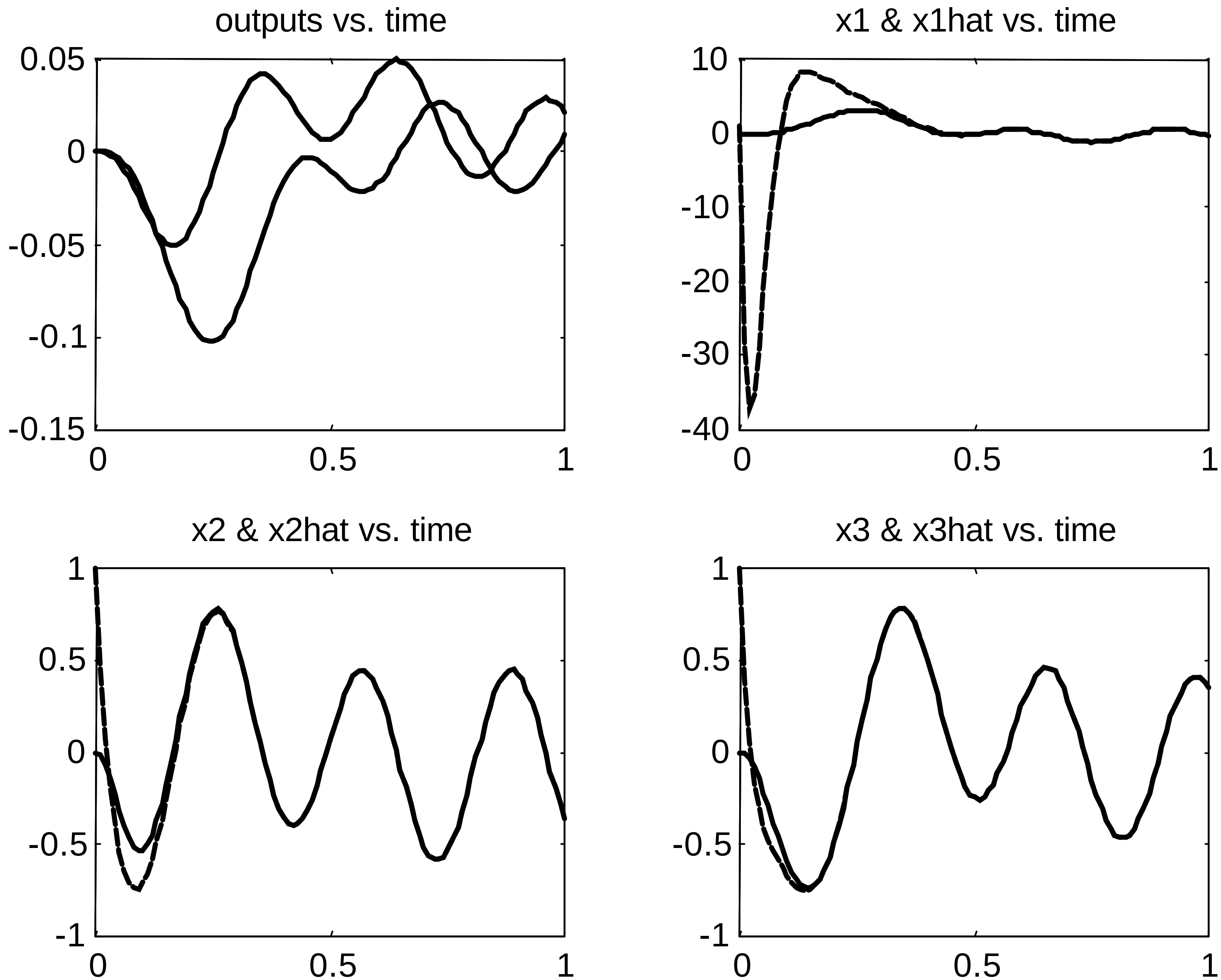
α : aileron command

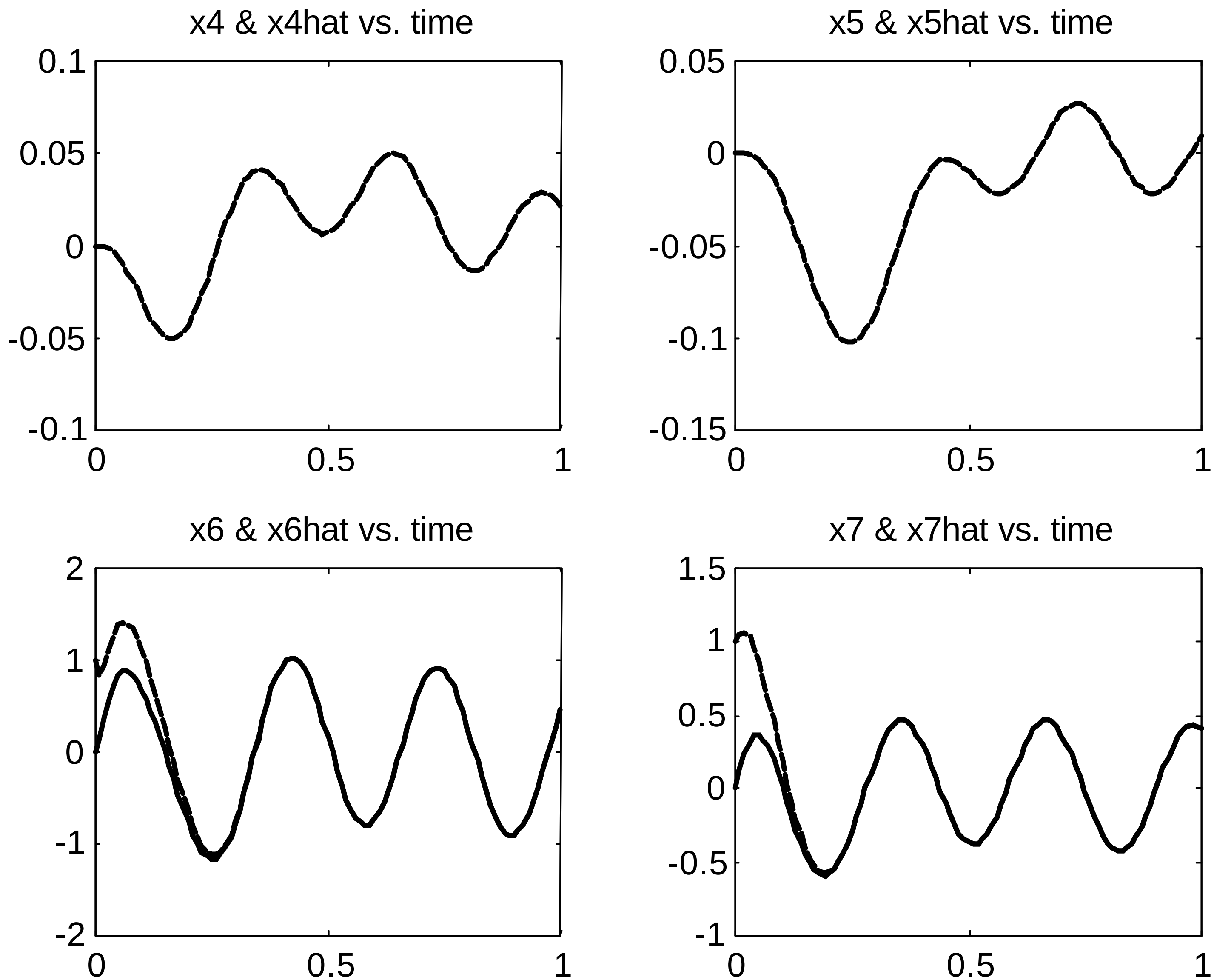
ρ : rudder command

$$u = [\alpha \quad \rho]^T$$

$$A=\begin{bmatrix} 0.277 & 0 & -32.9 & 9.81 & 0 & -5.543 & 0 \\ -0.103 & -8.325 & 3.75 & 0 & 0 & 0 & -28.640 \\ 0.365 & 0 & -.639 & 0 & 0 & -9.49 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -15 \end{bmatrix} \qquad B=\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20 & 0 \\ 0 & 10 \end{bmatrix}$$

$$C=\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \qquad D=\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$


outputs vs. time
x1 & x1hat vs. time
x2 & x2hat vs. time
x3 & x3hat vs. time
0.05
0
-0.05
-0.1
-0.15
10
-10
-20
-30
-40
1
0.5
-0.5
-1
0
0.5
1



x4 & x4hat vs. time
0.1
0.05
0
-0.05
-0.1
0
0.5
1
x5 & x5hat vs. time
0.05
0
-0.05
-0.1
-0.15
0
0.5
1
x6 & x6hat vs. time
2
1
0
-1
-2
0
0.5
1
x7 & x7hat vs. time
1.5
1
0.5
0
-0.5
-1
0
0.5
1

MORE REMARKS:

We have less flexibility if we try to feedback not the entire state, but just the *output* (as in classical controller design); i.e., *not all* eigenvalues can be placed if we use:

$$u = Ky = KCx$$

There are estimators, called ***functional*** estimators, that estimate not the entire state vector, but some scalar *function* of it, for example $K\hat{x}$. These can be quite efficient, and suffice to compute the estimate and the feedback signal simultaneously.

When placing observer poles, it is not advisable to place them too far left of the plant poles, because

1) they then have a large bandwidth and are susceptible to noise.

2) they may then give a large transient response, saturating the amplifiers.

A good rule of thumb is to place observer poles about *two to three times* farther left than the plant's closed-loop poles.

It is also possible to "place" eigenvalues *and eigenvectors*, although not *all* at the same time.

And finally, . . .

The state feedback we have computed is *unique* for a given set of desired closed-loop poles. This is only true in the *single input/single output case*. In multivariable systems, there will be **many** different gain matrices K that will place the closed-loop poles at any given location.

Each choice of gain K will have its own merits, and is generally a more difficult problem. This is largely the topic of the next course in multivariable control.