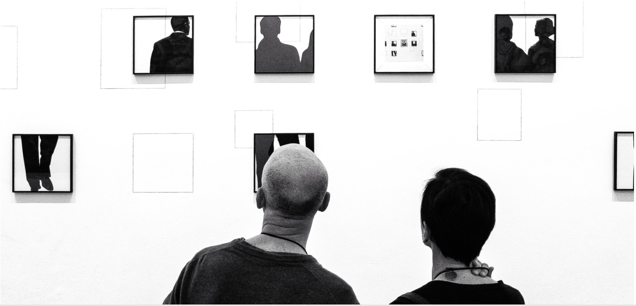
Observer and Observer State Feedback

Xu Chen

University of Washington



Introduction

- full state feedback is usually not available
- the state estimation problem
  - deterministic case: observer design
  - stochastic case: the most frequent option is Kalman filter

Open-loop observer

$$\frac{d}{dt}x(t) = Ax(t) + Bu(t),\ x(k+1) = Ax(k) + Bu(k)$$

- conceptually simplest scheme to estimate x:

$$\frac{d}{dt}\hat{x}(t) = A\hat{x}(t) + Bu(t),\ \hat{x}(k+1) = A\hat{x}(k) + Bu(k)$$

  with a best guess of initial estimate $\hat{x}(0) \overset{e.g.}{=} 0$.

- error dynamics: $e = x - \hat{x}$:

$$\dot{e}(t) = Ae(t),\ e(k+1) = Ae(k),\ e(0) = x_0 - \hat{x}(0)$$

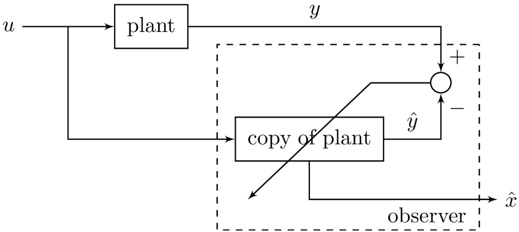
  - sensitive to input disturbances
  - if A is not Hurwitz/Schur stable, the error diverges
- open-loop observers look simple but do not work in practice

Luenberger (closed-loop) observer concept

- given system dynamics

$$\dot{x} = Ax + Bu,\ \ x(0) = x_0,\ \ A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times r}$$
$$y = Cx,\ \ y \in \mathbb{R}^{m\times n}$$

- in contrast to open-loop observers, the Luenberger observer adds correction based on output differences

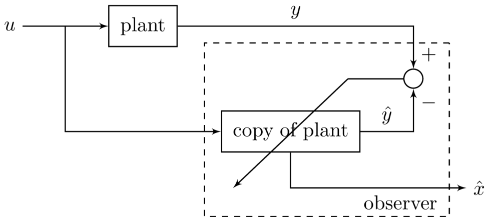


Luenberger (closed-loop) observer algorithm

plant:

$$\dot{x} = Ax + Bu, \;\; x(0) = x_0$$
$$y = Cx$$

observer concept



- observer realization:

$$\dot{\hat{x}} = A\hat{x} + Bu + L\left(y - \hat{y}\right) = A\hat{x} + Bu + L\left(y - C\hat{x}\right), \;\; \hat{x}(0) = 0$$
$$= \left(A - LC\right)\hat{x} + Ly + Bu$$

Luenberger (closed-loop) observer error dynamics

- system dynamics

$$\dot{x} = Ax + Bu,\ x(0) = x_0,\ A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times r}$$
$$y = Cx,\ y \in \mathbb{R}^{m\times n}$$

- Luenberger observer with correction:

$$\dot{\hat{x}} = A\hat{x} + Bu + L\left(y - \hat{y}\right) = A\hat{x} + Bu + L\left(y - C\hat{x}\right),\ \hat{x}(0) = 0$$
$$= \left(A - LC\right)\hat{x} + Ly + Bu$$

- error dynamics: $e = x - \hat{x}$:

$$\dot{e} = Ae - LCe = (A - LC)e,\ e(0) = x(0)$$

- if all eigenvalues of $A - LC$ are on the left half plane, then the error dynamics can be made asymptotically stable

Luenberger (closed-loop) observer

Theorem

If (A, C) is an observable pair, then all the eigenvalues of $A - LC$ can be arbitrarily assigned, provided that they are symmetric with respct to the real axis of the complex plane.

- we show the SISO case when A and C are in observable canonical form (if not, a similarity transform can help out):

$$A = \begin{bmatrix} -\alpha_{n-1} & 1 & 0 & \dots \\ \vdots & 0 & \ddots & \ddots \\ -\alpha_1 & \vdots & \ddots & 1 \\ -\alpha_0 & 0 & \dots & 0 \end{bmatrix}, \; B = \begin{bmatrix} \beta_{n-1} \\ \vdots \\ \beta_1 \\ \beta_0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \end{bmatrix}, \; D = d$$

$$\det(sI - A) = s^n + \alpha_{n-1}s^{n-1} + \cdots + \alpha_1 s + \alpha_0$$

Observer eigenvalue placement: o.c.f.

- Luenberger observer with correction:

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + Bu + L\left(y - \hat{y}\right) = A\hat{x} + Bu + L\left(y - C\hat{x}\right),\ \hat{x}(0) = 0\\ &= \left(A - LC\right)\hat{x} + Ly + Bu\end{aligned}$$

- Goal: place eigenvalues of the observer at locations $\bar{p}_1, \cdots, \bar{p}_n$:

$$\begin{aligned}\det\left(sI - (A - LC)\right) &= (s - \overline{p}_1)(s - \overline{p}_2)\cdots(s - \overline{p}_n)\\ &= s^n + \overline{\gamma}_{n-1}s^{n-1} + \cdots + \overline{\gamma}_1 s + \overline{\gamma}_0\end{aligned}$$

Observer eigenvalue placement: o.c.f.

- Goal: place eigenvalues of the observer at locations $\bar{p}_1, \cdots, \bar{p}_n$:

$$\det\left(sI - (A - LC)\right) = (s - \overline{p}_1)(s - \overline{p}_2)\cdots(s - \overline{p}_n)$$
$$= s^n + \overline{\gamma}_{n-1}s^{n-1} + \cdots + \overline{\gamma}_1 s + \overline{\gamma}_0$$

- Let $L = [l_0, l_1, \ldots, l_{n-1}]^T$. The unique structures of A and C give

$$LC = \begin{bmatrix} l_0 \\ \vdots \\ l_{n-2} \\ l_{n-1} \end{bmatrix} \begin{bmatrix} 1 & 0 & \ldots & 0 \end{bmatrix} = \begin{bmatrix} l_0 & 0 & \ldots & 0 \\ \vdots & 0 & \ddots & \vdots \\ l_{n-2} & \ddots & \ddots & 0 \\ l_{n-1} & 0 & \ldots & 0 \end{bmatrix}$$

$$A - LC = \begin{bmatrix} -\alpha_{n-1} - l_0 & 1 & 0 & \ldots & 0 \\ -\alpha_{n-2} - l_1 & 0 & \ddots & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ -\alpha_1 - l_{n-2} & \vdots & \ddots & 0 & 1 \\ -\alpha_0 - l_{n-1} & 0 & \ldots & 0 & 0 \end{bmatrix}$$

Observer eigenvalue placement: o.c.f.

- A and $A - LC$ have the same structure:

$$A = \begin{bmatrix} -\alpha_{n-1} & 1 & 0 & \dots \\ \vdots & 0 & \ddots & \ddots \\ -\alpha_1 & \vdots & \ddots & 1 \\ -\alpha_0 & 0 & \dots & 0 \end{bmatrix}, \; A - LC = \begin{bmatrix} -\alpha_{n-1} - l_0 & 1 & 0 & \dots \\ \vdots & 0 & \ddots & \ddots \\ -\alpha_1 - l_{n-2} & \vdots & \ddots & 1 \\ -\alpha_0 - l_{n-1} & 0 & \dots & 0 \end{bmatrix}$$

- Recall: $\det(sI - A) = s^n + \alpha_{n-1}s^{n-1} + \dots + \alpha_1 s + \alpha_0$.
- Thus

$$\det(sI - (A - LC)) = s^n + \underbrace{(\alpha_{n-1} + l_0)}_{\text{target: } \overline{\gamma}_{n-1}} s^{n-1} + \dots + \underbrace{(\alpha_0 + l_{n-1})}_{\text{target: } \overline{\gamma}_0}$$

- Hence

$$l_0 = \overline{\gamma}_{n-1} - \alpha_{n-1}$$
$$\vdots$$
$$l_{n-1} = \overline{\gamma}_0 - \alpha_0$$

General observer eigenvalue placement

- What if (A, B, C, D) is not in the observable canonical form?
- We can transform it to o.c.f. via a similarity transform:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \overset{x = R^{-1}x_{ob}}{\Longrightarrow} \begin{cases} \dot{x}_{ob} & = \underbrace{RAR^{-1}}_{A_o} x_{ob} + \underbrace{RB}_{B_o} u \\ y & = C_o x_{ob} = CR^{-1} x_{ob} \end{cases}$$

- use previous formulas to design $\tilde{L}$ in:

$$\dot{\hat{x}}_{ob} = \left(A_o - \tilde{L}C_o\right)\hat{x}_{ob} + \tilde{L}y + B_o u \qquad \text{(analysis form)}$$

- correspondingly in the original state space (via $\hat{x}_{ob} = R\hat{x}$):

$$R\dot{\hat{x}} = \left(RAR^{-1} - \tilde{L}CR^{-1}\right)R\hat{x} + \tilde{L}y + RBu$$

$$\Rightarrow \dot{\hat{x}} = (A - \overbrace{R^{-1}\tilde{L}}^{L} C)\hat{x} + Ly + Bu \qquad \text{(implementation form)}$$

General observer eigenvalue placement

- **Powerful fact**: if system $\Sigma = (A, B, C, D)$ is observable, then we can arbitrarily place the observer eigenvalues.

Observer design in MATLAB and Python

%MATLAB	#Python
A = [0 1;-4 -0.2]; B = [0 1]';	import control as ct
C = [1 0];	import numpy as np
sys = ss(A,B,C,0);	A = np.array([[0, 1],[-4, -0.2]])
eig(A)	C = np.array([[1], [0]]).T
L = place(A',C',[-2,-3])'	L = ct.place(A.T,C.T,[-2, -3]).T
eig(A-L*C)	print(L)

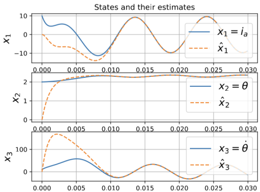
Motor control example

Example 12.2.1 (Motor Control) To see an example performance of the observer design, we apply the algorithm to a motor system and program Equation 12.10. Here, the three states are the current in the motor electronics, the angular position and angular velocity of the motor. Only the angular position is directly measured on the output side. We compute first the plant eigenvalues, then check observability and place the observer eigenvalues.

```
% observer/motorobs.m
% State observer design for motion control in MATLAB
%% Continuous-time system model
% motor parameters
L = 1e-3; R = 1; J = 5e-5; B = 1e-4; K = 0.1;

% state-space model
A = [-R/L, 0, -K/L; 0, 0, 1; K/J, 0, -B/J];
B = [1/L; 0; 0];
C = [0, 1, 0];
D = [0];

% check original eigenvalues
eig(A)
%% Observer design
% check observability
O = obsv(A,C);
rank(O)

% desired poles for the observer
pole_des = [-500+250j, -500-250j, -1000];

% design observer by placing poles of A-LC
Lt = place(A.',C.',pole_des);
L  = Lt.'

% check poles of estimator-error dynamics
est_poles = eig(A - L*C)

%% Simulation
% define augmented system to run the simulation
Aaug = [A, zeros(3,3); L*C, A-L*C];
Baug = [B;B];
Caug = [C, zeros(1,3)];
Daug = 0;
sys = ss(Aaug,Baug,Caug,Daug);

% define intial conditions
x0 = [10, 2, 10]'; xhat0 = [0, 0, 0]'; X0 = [x0; xhat0];

% define simulink parameters
Tend = 0.03; % simulation end time
amplitude = 10; % sin wave input amplitude
initpha = 0; % initial phase
freq = 600; % sin wave freq (rad/s)
t = 0:1e-4:Tend;

u = amplitude*sin(freq*t+initpha);
```

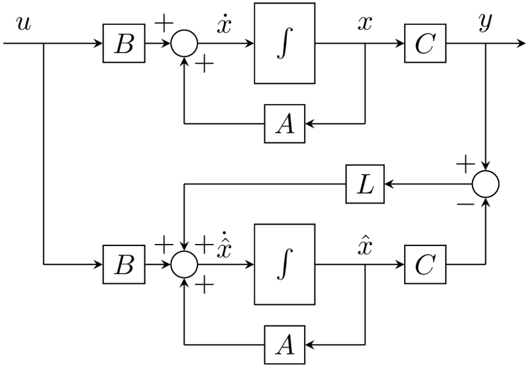
```
x0 = np.array([10, 2, 10]); xhat0 = np.array([0, 0, 0]); X0 =
↪ np.array([x0, xhat0]).reshape((6, 1))

Tend = 0.03; amplitude = 10; initpha = 0; freq = 600
t = np.arange(0, Tend, 1e-4)

u = amplitude * np.sin(freq * t + initpha)

[Y, T, X] = ct.lsim(sys, u, t, X0)

plt.figure()
plt.subplot(3, 1, 1)
plt.plot(t, X[:, 0], t, X[:, 3], '--', linewidth=1.5)
plt.xlabel('time (sec)')
plt.legend(['$x_1 = i_a$', '$\hat x_1$'], fontsize=16)
plt.grid()
plt.ylabel('$x_1$', fontsize=16)
plt.title('States and their estimates')
plt.subplot(3, 1, 2)
plt.plot(t, X[:, 1], t, X[:, 4], '--', linewidth=1.5)
plt.xlabel('time (sec)')
plt.legend(['$x_2 = \\theta$', '$\hat x_2$'], fontsize=16)
plt.grid()
plt.ylabel('$x_2$', fontsize=16)
plt.subplot(3, 1, 3)
plt.plot(t, X[:, 2], t, X[:, 5], '--', linewidth=1.5)
plt.xlabel('time (sec)')
plt.legend(['$x_3 = \dot{\\theta}$', '$\hat x_3$'], fontsize=16)
plt.grid()
plt.ylabel('$x_3$', fontsize=16)
plt.show()
```

From the generated result below, we see that despite the initial error between the true states and the estimated states, the estimation errors quickly converge to zero for all the three states after about 0.01 second. Try modify the observer eigenvalues and see how they affect the convergence.



Luenberger observer summary

- observer dynamics: $\dot{\hat{x}} = A\hat{x} + Bu + L\left(y - C\hat{x}\right), \ \hat{x}(0) = 0$
- block diagram



Luenberger observer summary

- system dynamics

$$\dot{x} = Ax + Bu, \; x(0) = x_0, \; A \in \mathbb{R}^{n\times n}, B \in \mathbb{R}^{n\times r}$$
$$y = Cx, \; y \in \mathbb{R}^{m\times 1}$$

- observer dynamics

$$\begin{aligned}\dot{\hat{x}} &= A\hat{x} + Bu + L\left(y - C\hat{x}\right), \; \hat{x}(0) = 0 \\ &= \left(A - LC\right)\hat{x} + LCx + Bu\end{aligned}$$

- augmented system

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & 0 \\ LC & A - LC \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} u$$

Luenberger observer summary

- augmented system

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & 0 \\ LC & A-LC \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} u$$
$$y = Cx$$

- to see the distribution of eigenvalues, note the error dynamics $\dot{e} = (A - LC)e \Rightarrow$

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A-LC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u$$

$\Rightarrow$eigenvalues are separated into: $\lambda(A)$ and observer eigenvalues

- underlying similarity transform: $\begin{bmatrix} x \\ e \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ I_n & -I_n \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix}$

Discrete-time observers: Introduction

- full state feedback is usually not available
- often observers are implemented in the discrete-time domain
- the discrete-time observer design
  - basic form: analogous to the continuous-time Luenberger observer
  - predict and correct form:
    - direct DT design
    - leverages discrete-time signal properties

Discrete-time full state observer

- standard discrete-time observer:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ \hat{x}(k+1) &= A\hat{x}(k) + Bu(k) + L(y(k) - C\hat{x}(k)) \\ y(k) &= Cx(k) \end{aligned}$$

- error dynamics:$e(k) = x(k) - \hat{x}(k)$, $e(k+1) = Ae(k) - LCe(k)$
- overall dynamics

$$\begin{bmatrix} x(k+1) \\ e(k+1) \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x(k) \\ e(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k)$$

$$y(k+1) = [C,\ 0] \begin{bmatrix} x(k+1) \\ e(k+1) \end{bmatrix}$$

- **Powerful fact**: the error dynamics can be arbitrarily assigned if the system is observable.

DT full state observer with predictor

- motivation: $\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L(y(k) - C\hat{x}(k))$ doesn't use most recent measurement $y(k+1) = Cx(k+1)$
- discrete-time observer **with predictor**:

predictor: $\hat{x}(k+1|k) = A\hat{x}(k|k) + Bu(k)$

corrector: $\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + L(y(k+1) - C\hat{x}(k+1|k))$

  - $\hat{x}(k|k)$: estimate of $x(k)$ based on measurements up to time k
  - $\hat{x}(k|k-1)$: estimate based on measurements up to time $k-1$
  - $e(k) \triangleq x(k) - \hat{x}(k|k)$: estimation error
- error dynamics

$$
\begin{aligned}
\hat{x}(k+1|k+1) &= (I - LC)\hat{x}(k+1|k) + Ly(k+1) \\
&= (I - LC)A\hat{x}(k|k) + (I - LC)Bu(k) + Ly(k+1) \\
\Rightarrow e(k+1) &= x(k+1) - Ly(k+1) - (I - LC)A\hat{x}(k|k) - (I - LC)Bu(k) \\
&= (A - LCA)e(k)
\end{aligned}
$$

DT full state observer with predictor

$$e(k+1) = \left(A - L\underbrace{CA}_{\tilde{C}}\right) e(k), \quad e(0) = (I - LC)\, x_0$$

- the error dynamics can be arbitrarily assigned if the pair $\left(A,\ \tilde{C}\right) = (A,\ CA)$ is observable
- observability matrix

$$\tilde{Q}_d = \begin{bmatrix} \tilde{C} \\ \tilde{C}A \\ \vdots \\ \tilde{C}A^{n-1} \end{bmatrix} = \overbrace{\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}}^{Q_d} A$$

- if A is invertible, then $\tilde{Q}_d$ has the same rank as Q_d
- $\left(A,\ \tilde{C}\right)$ is observable if $(A,\ C)$ is observable and A is nonsingular (guaranteed if discretized from a CT system)

Example

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} -a_2 & 1 & 0 \\ -a_1 & 0 & 1 \\ -a_0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix} u(k),$$

$y(k) = x_1(k)$. Place all eigenvalues of an **observer with predictor** at the origin.

$$A - LCA = \begin{bmatrix} -a_2 & 1 & 0 \\ -a_1 & 0 & 1 \\ -a_0 & 0 & 0 \end{bmatrix} - \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} \begin{bmatrix} -a_2 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} (l_1 - 1)a_2 & 1 - l_1 & 0 \\ l_2 a_2 - a_1 & -l_2 & 1 \\ l_3 a_2 - a_0 & -l_3 & 0 \end{bmatrix}$$

$\det(A - LCA - \lambda I) = ((l_1 - 1)a_2 - \lambda)(l_2 + \lambda)\lambda +$
$(1 - l_1)(l_3 a_a - a_0) + l_3((l_1 - 1)a_2 - \lambda) + \lambda(1 - l_1)(l_2 a_2 - a_1)$
roots must be all 0 $\Rightarrow l_1 = 1,\ l_2 = l_3 = 0.$

Observer state feedback

given system dynamics:

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

- state feedback control: arbitrary eigenvalue assignment if system controllable
- observer design: arbitrary observer eigenvalue assignment for state estimation if system observable
- when full states are not available, what's the performance if we combine both?

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

Closed-loop dynamics

- full closed-loop system

$$\dot{x} = Ax + Bu$$
$$y = Cx$$
$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$
$$u = -K\hat{x} + v$$

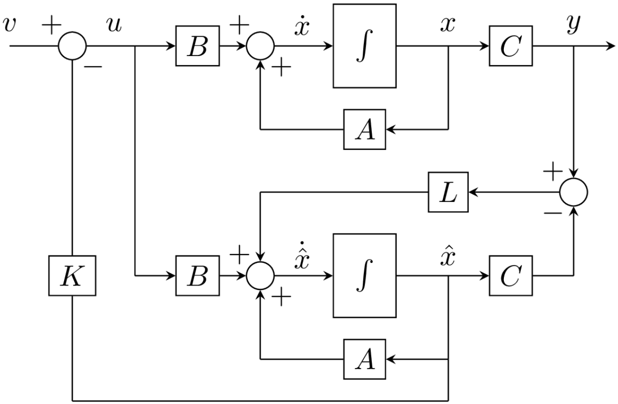
$$\Rightarrow \frac{d}{dt}\begin{bmatrix} x \\ \hat{x} \end{bmatrix} = \begin{bmatrix} A & -BK \\ LC & A - LC - BK \end{bmatrix}\begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B \\ B \end{bmatrix} v$$

- using again similarity transform $\begin{bmatrix} x \\ e \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ I_n & -I_n \end{bmatrix}\begin{bmatrix} x \\ \hat{x} \end{bmatrix}$ gives

$$\frac{d}{dt}\begin{bmatrix} x \\ e \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix}\begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} v$$

Block diagram

- $\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$, $u = -K\hat{x} + v$



The separation theorem

- closed-loop dynamics

$$\frac{d}{dt}\begin{bmatrix} x \\ e \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix}\begin{bmatrix} x \\ e \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} v$$

- **powerful result: separation theorem:** closed-loop eigenvalues consist of
  - eigenvalues of $A - BK$ from the state feedback control design
  - eigenvalues of $A - LC$ from the observer design
- can design K and L separately based on discussed tools
- if system is controllable and observable, we can arbitrarily assign the closed-loop eigenvalues
- rule of thumb: assign observer dynamics to be faster than state-feedback dynamics